# MATH3014-6027 Design (and Analysis) of Experiments

Dave Woods

2023-02-22

# Contents

# Preface

These are lecture notes for the modules MATH3014 and MATH6027 Design (and Analysis) of Experiments at the University of Southampton for academic year 2022-23.

Southampton prerequisites for this module are MATH2010 or MATH6174 and STAT6123 (or equivalent modules on linear modelling).

# Chapter 1

# Motivation, introduction and revision

**Definition 1.1.** An **experiment** is the process through which data are collected to answer a scientific question (physical science, social science, actuarial science ...) by **deliberately** varying some features of the process under study in order to understand the impact of these changes on measureable responses.

In this course we consider only *intervention* experiments, in which some aspects of the process are under the experimenters' control. We do not consider *surveys* or *observational* studies.

**Definition 1.2. Design of experiments** is the topic in Statistics concerned with the selection of settings of controllable variables or factors in an experiment and their allocation to experimental units in order to maximise the effectiveness of the experiment at achieving its aim.

People have been designing experiments for as long as they have been exploring the natural world. Collecting empirical evidence is key for scientific development, as described in terms of clinical trials by xkcd:

Some notable milestones in the history of the design of experiments include:

- prior to the 20th century:
  - Francis Bacon (17th century; pioneer of the experimental methods)
  - James Lind (18th century; experiments to eliminate scurvy)
  - Charles Peirce (19th century; advocated randomised experiments and randomisation-based inference)
- 1920s: agriculture (particularly at the Rothamsted Agricultural Research Station)
- 1940s: clinical trials (Austin Bradford-Hill)
- 1950s: (manufacturing) industry (W. Edwards Deming; Genichi Taguchi)

- 1960s: psychology and economics (Vernon Smith)
- 1980s: in-silico (computer experiments)
- 2000s: online (A/B testing)

See Luca and Bazerman (2020) for further history, anecdotes and examples, especially from psychology and technology.

Figure 1.1 shows the Broadbalk agricultural field experiment at Rothamsted, one of the longest continuous running experiments in the world, which is testing the impact of different manures and fertilizers on the growth of winter wheat.

Figure 1.1: The Broadbalk experiment, Rothamsted (photograph taken 2016)

## 1.1 Motivation

**Example 1.1.** Consider an experiment to compare two treatments (e.g. drugs, diets, fertilisers, ...). We have $n$ subjects (people, mice, plots of land, ...), each of which can be assigned one of the two treatments. A response (protein measurement, weight, yield, ...) is then measured.

**Question:** How many subjects should be assigned to each treatment to gain the most precise[1] inference about the difference in response from the two treatments?

Consider a linear statistical model[2] for the response (see MATH2010 or MATH6174/STAT6123):

[1] Smallest variance.

[2] In this course, we will almost always start with a statistical model which we wish to use to answer our scientific question.

$$Y_j = \beta_0 + \beta_1 x_j + \varepsilon_j\,, \qquad j = 1, \ldots, n\,, \tag{1.1}$$

where $\varepsilon_j \sim N(0, \sigma^2)$ are independent and identically distributed errors and $\beta_0, \beta_1$ are unknown constants (parameters).

Let[3]

$$x_j = \begin{cases} 0 & \text{if treatment 1 is applied to the } j\text{th subject} \\ 1 & \text{if treatment 2 is applied to the } j\text{th subject,} \end{cases}$$

for $j = 1, \ldots, n$.

The difference in expected response from treatments 1 and 2 is

$$\begin{aligned} \mathrm{E}[Y_j \,|\, x_j = 1] - \mathrm{E}[Y_j \,|\, x_j = 0] &= \beta_0 + \beta_1 - \beta_0 \\ &= \beta_1\,. \end{aligned} \tag{1.2}$$

Therefore, we require the the most precise estimator of $\beta_1$ possible. That is, we wish to make the variance of our estimator of $\beta_1$ as small as possible.

Parameters $\beta_0$ and $\beta_1$ can be estimated using least squares (see MATH2010 or MATH6174/STAT6123). For $Y_1, \ldots, Y_n$, we can write the model down in matrix form:

$$\begin{bmatrix} Y_1 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{bmatrix}.$$

Or, by defining some notation:

$$\boldsymbol{Y} = X\boldsymbol{\beta} + \boldsymbol{\varepsilon} \tag{1.3}$$

where

- $\boldsymbol{Y}$ - $n \times 1$ vector of responses;
- $X$ - $n \times p$ model matrix;
- $\boldsymbol{\beta}$ - $p \times 1$ vector of parameters;
- $\boldsymbol{\varepsilon}$ - $n \times 1$ vector of errors.

The **least squares estimators**, $\hat{\boldsymbol{\beta}}$, are chosen such that the quadratic form

$$(\boldsymbol{Y} - X\boldsymbol{\beta})^{\mathrm{T}}(\boldsymbol{Y} - X\boldsymbol{\beta})$$

is minimised (recall that $\mathrm{E}(\mathbf{Y}) = X\boldsymbol{\beta}$). Therefore

[3] Other choices of *coding* can be used: e.g. -1,1; it makes no difference for our current purpose.

$$\hat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}}(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y} + \boldsymbol{\beta}^{\mathrm{T}}X^{\mathrm{T}}X\boldsymbol{\beta} - 2\boldsymbol{\beta}^{\mathrm{T}}X^{\mathrm{T}}\boldsymbol{Y})\,.$$

If we differentiate with respect to $\boldsymbol{\beta}$[4],

$$\frac{\partial}{\partial\boldsymbol{\beta}} = 2X^{\mathrm{T}}X\boldsymbol{\beta} - 2X^{\mathrm{T}}\boldsymbol{Y}\,,$$

and equate to 0, we get the estimators

$$\hat{\boldsymbol{\beta}} = (X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}\boldsymbol{Y}\,. \tag{1.4}$$

These are the least squares estimators.

For Example 1.1,

$$X = \begin{bmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}, \qquad X^{\mathrm{T}}X = \begin{bmatrix} n & \sum x_j \\ \sum x_j & \sum x_j^2 \end{bmatrix},$$

$$(X^{\mathrm{T}}X)^{-1} = \frac{1}{n\sum x_j^2 - (\sum x_j)^2}\begin{bmatrix} \sum x_j^2 & -\sum x_j \\ -\sum x_j & n \end{bmatrix}, \qquad X^{\mathrm{T}}\boldsymbol{Y} = \begin{bmatrix} \sum Y_j \\ \sum x_jY_j \end{bmatrix}.$$

Then,

$$\begin{aligned}\hat{\boldsymbol{\beta}} = \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \end{bmatrix} &= \frac{1}{n\sum x_j^2 - (\sum x_j)^2}\begin{bmatrix} \sum x_j^2 & -\sum x_j \\ -\sum x_j & n \end{bmatrix}\begin{bmatrix} \sum Y_j \\ \sum x_jY_j \end{bmatrix} \\ &= \frac{1}{n\sum x_j^2 - (\sum x_j)^2}\begin{bmatrix} \sum Y_j\sum x_j^2 - \sum x_j\sum x_jY_j \\ n\sum x_jY_j - \sum x_j\sum Y_j \end{bmatrix}. \end{aligned} \tag{1.5}$$

We don't usually work through the algebra in such detail; the matrix form is often sufficient for theoretical and numerical calculations and software, e.g. `R`, can be used.

The precision of $\hat{\boldsymbol{\beta}}$ is measured via the variance-covariance matrix, given by

$$\begin{aligned}\text{Var}(\hat{\boldsymbol{\beta}}) &= \text{Var}\{(X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}\boldsymbol{Y}\} && (1.6)\\ &= (X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}\text{Var}(\boldsymbol{Y})X(X^{\mathrm{T}}X)^{-1} && (1.7)\\ &= (X^{\mathrm{T}}X)^{-1}\sigma^2\,, && (1.8)\end{aligned}$$

where $\boldsymbol{Y} \sim N(X\boldsymbol{\beta}, I_n\sigma^2)$, where $I_n$ is an $n \times n$ identity matrix.

[4]Check the Matrix Cookbook for matrix calculus, amongst much else.

Hence, in our example,

$$\begin{aligned}\mathrm{Var}(\hat{\boldsymbol{\beta}}) &= \frac{1}{n\sum x_j^2 - (\sum x_j)^2}\begin{bmatrix} \sum x_j^2 & -\sum x_j \\ -\sum x_j & n \end{bmatrix}\sigma^2 \\ &= \begin{bmatrix} \mathrm{Var}(\hat{\beta}_0) & \mathrm{Cov}(\hat{\beta}_0, \hat{\beta}_1) \\ \mathrm{Cov}(\hat{\beta}_0, \hat{\beta}_1) & \mathrm{Var}(\hat{\beta}_1) \end{bmatrix}.\end{aligned}$$

For estimating the difference between treatments, we are interested in

$$\begin{aligned}\mathrm{Var}(\hat{\beta}_1) &= \frac{n}{n\sum x_j^2 - (\sum x_j)^2}\sigma^2 \\ &= \frac{1}{\sum x_j^2 - n\bar{x}^2}\sigma^2\,,\end{aligned}$$

where $\bar{x} = \sum x_j/n$.

Assuming constant $\sigma^2$, to achieve the most precise estimator we need to minimise $\mathrm{Var}(\hat{\beta}_1)$ or equivalently maxmise $\sum x_j^2 - n\bar{x}^2$. This goal can be achieved through the choice of $x_1, \ldots, x_n$:

- as each $x_j$ can only take one of two values, 0 or 1, this is equivalent to choosing the numbers of subjects assigned to treatment 1 and treatment 2;
- call these $n_1$ and $n_2$ respectively, with $n_1 + n_2 = n$.

We can find an upper bound for the quantity $\sum x_j^2 - n\bar{x}^2$. As each $x_i \in \{0, 1\}$ we have

$$\begin{aligned}\sum x_j^2 &= \sum x_j && (1.9)\\ &= n\bar{x}\,. && (1.10)\end{aligned}$$

Hence,

$$\begin{aligned}\sum x_j^2 - n\bar{x}^2 &= n\bar{x} - n\bar{x}^2 \\ &= n\bar{x}(1-\bar{x}) \\ &\leq n/4\,,\end{aligned}$$

as we have a quadratic equation in $\bar{x}$ that is maximised at $\bar{x} = 1/2$.

If we can find a set of design points that satisfy $\bar{x} = 1/2$, we will have an **optimal design**. Assuming $n$ is even, one possibility is

- $n_1 = \frac{n}{2}$ subjects assigned to treatment 1 ($x_j = 0$) and
- $n_2 = \frac{n}{2}$ subjects assigned to treatment 2 ($x_j = 1$).

For $n$ odd, we choose $n_1 = \frac{n+1}{2}$, $n_2 = \frac{n-1}{2}$, or vice versa, to get as close as possible to the optimal design.

**Definition 1.3.** We can assess different designs using their **efficiency**:

$$\text{Eff} = \frac{\text{Var}(\hat{\beta}_1 \,|\, d^*)}{\text{Var}(\hat{\beta}_1 \,|\, d_1)} \tag{1.11}$$

where $d_1$ is a design we want to assess and $d^*$ is the optimal design with smallest variance. Note that $0 \leq \text{Eff} \leq 1$.

In Figure 1.2 below, we plot this efficiency for Example 1.1, using different choices of $n_1$. The total number of runs is fixed at $n = 100$, and the function `eff` calculates the efficiency (assuming $n$ is even) from Definition 1.3 for a design with $n_1$ subjects assigned to treatment 1. Clearly, efficiency of 1 is achieved when $n_1 = n_2$ (equal allocation of treatments 1 and 2). If $n_1 = 0$ or $n_1 = 1$, the efficiency is zero; we cannot estimate the difference between two treatments if we only allocate subjects to one of them.

```
n <- 100
eff <- function(n2) 4 * n2 * (n - n2) / n^2
curve(eff, from = 0, to = n, ylab = "Eff", xlab = expression(n[1]))
```

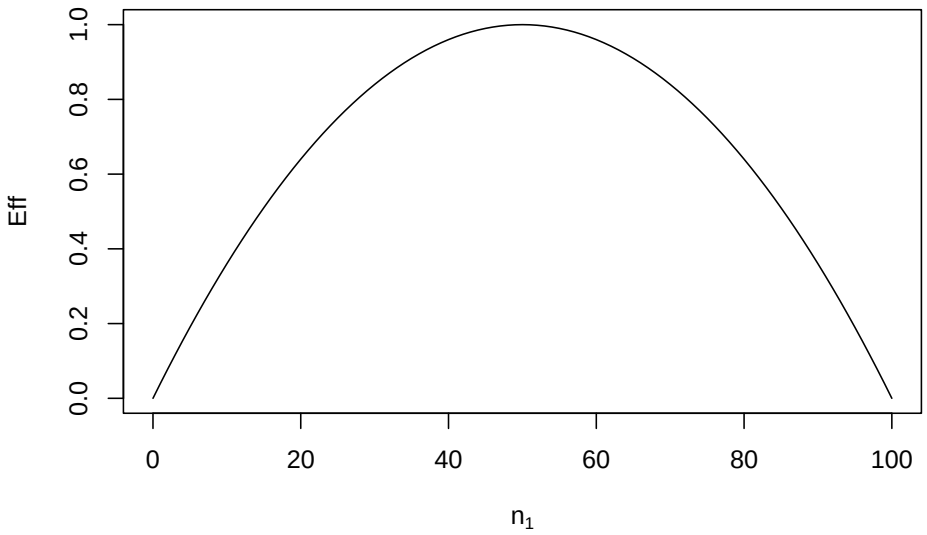



Figure 1.2: Efficiencies for designs for Example 1.1 with different numbers, $n_1$, of subjects assigned to treatment 1 when the total number of subjects is $n = 100$.

## 1.2 Aims of experimentation and some examples

Some reasons experiments are performed:

1. Treatment comparison

- compare several treatments (and choose the best)
- e.g. clinical trial, agricultural field trial

2. Factor screening

- many complex systems may involve a large number of (discrete) factors (controllable features)
- which of these factors have a substantive impact?
- (relatively) small experiments
- e.g. industrial experiments on manufacturing processes

3. Response surface exploration

- detailed description of relationship between important (continuous) variables and response
- typically second order polynomial regression models
- larger experiments, often built up sequentially
- e.g. alcohol yields in a pharmaceutical experiments

4. Optimisation

- finding settings of variables that lead to maximum or minimum response
- typically use response surface methods and sequential "hill climbing" strategy

In this module, we will focus on **treatment comparison** (Chapters 2 and 3) and **factor screening** (Chapters 4, 5 and 6).

## 1.3 Some definitions

**Definition 1.4.** The **response** $Y$ is the outcome measured in an experiment; e.g. yield from a chemical process. The response from the $n$ observations are denoted $Y_1, \ldots, Y_n$.

**Definition 1.5. Factors** (discrete) or **variables** (continuous) are features which can be set or controlled in an experiment; $m$ denotes the number of factors or variables under investigation. For discrete factors, we call the possible settings of the factor its **levels**. We denote by $x_{ij}$ the value taken by factor or variable $i$ in the $j$th run of the experiment ($i = 1, \ldots, m$; $j = 1, \ldots, n$).

**Definition 1.6.** The **treatments** or **support points** are the *distinct* combinations of factor or variable values in the experiment.

**Definition 1.7.** An experimental **unit** is the basic element (material, animal, person, time unit, ...) to which a treatment can be applied to produce a response.

In Example 1.1 (comparing two treatments):

- Response $Y$: Measured outcome, e.g. protein level or pain score in clinical trial, yield in an agricultural field trial.
- Factor $x$: "treatment" applied
- Levels

$$\begin{array}{ll} \text{treatment 1} & x = 0 \\ \text{treatment 2} & x = 1 \end{array}$$

- Treatment or support point: Two treatments or support points
- Experimental unit: Subject (person, animal, plot of land, ...).

## 1.4 Principles of experimentation

Three fundamental principles that need to be considered when designing an experiment are:

- replication
- randomisation
- stratification (blocking)

### 1.4.1 Replication

Each treatment is applied to a number of experimental units, with the $j$th treatment replicated $r_j$ times. This enables the estimation of the variances of treatment effect estimators; increasing the number of replications, or replicates, decreases the variance of estimators of treatment effects. (Note: proper replication involves independent application of the treatment to different experimental units, not just taking several measurements from the same unit).

### 1.4.2 Randomisation

Randomisation should be applied to the allocation of treatments to units. Randomisation protects against **bias**; the effect of variables that are unknown and potentially uncontrolled or subjectivity in applying treatments. It also provides a formal basis for inference and statistical testing.

For example, in a clinical trial to compare a new drug and a control random allocation protects against

- "unmeasured and uncontrollable" features (e.g. age, sex, health)
- bias resulting from the clinician giving new drug to patients who are sicker.

Clinical trials are usually also *double-blinded*, i.e. neither the healthcare professional nor the patient knows which treatment the patient is receiving.

### 1.4.3 Stratification (or blocking)

We would like to use a wide variety of experimental units (e.g. people or plots of land) to ensure **coverage** of our results, i.e. validity of our conclusions across the

population of interest. However, if the sample of units from the population is too heterogenous, then this will induce too much random variability, i.e. increase $\sigma^2$ in $\varepsilon_j \sim N(0, \sigma^2)$, and hence increase the variance of our parameter estimators.

We can reduce this extraneous variation by splitting our units into homogenous sets, or **blocks**, and including a blocking term in the model. The simplest blocked experiment is a **randomised complete block design**, where each block contains enough units for all treatments to be applied. Comparisons can then be made *within* each block.

Basic principle: block what you can, randomise what you cannot.

Later we will look at blocking in more detail, and the principle of **incomplete blocks**.

## 1.5 Revision on the linear model

Recall: $\boldsymbol{Y} = X\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, with $\boldsymbol{\varepsilon} \sim N(\mathbf{0}, I_n\sigma^2)$. Let the $j$th row of $X$ be denoted $\boldsymbol{x}_j^{\mathrm{T}}$, which holds the values of the predictors, or explanatory variables, for the $j$th observation. Then

$$Y_j = \boldsymbol{x}_j^{\mathrm{T}}\boldsymbol{\beta} + \varepsilon_j\,, \quad j = 1, \ldots, n\,.$$

For example, quite commonly, for continuous variables

$$\boldsymbol{x}_j = (1, x_{1j}, x_{2j}, \ldots, x_{mj})^{\mathrm{T}}\,,$$

and so

$$\boldsymbol{x}_j^{\mathrm{T}}\boldsymbol{\beta} = \beta_0 + \beta_1 x_{1j} + \cdots + \beta_m x_{mj}\,.$$

The least squares estimators are given by

$$\hat{\boldsymbol{\beta}} = (X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}\boldsymbol{Y}\,,$$

with

$$\mathrm{Var}(\hat{\boldsymbol{\beta}}) = (X^{\mathrm{T}}X)^{-1}\sigma^2\,.$$

### 1.5.1 Analysis of Variance as Model Comparison

To assess the goodness-of-fit of a model, we can use the residual sum of squares

$$\begin{aligned}\text{RSS} &= (\boldsymbol{Y} - X\hat{\boldsymbol{\beta}})^{\mathrm{T}}(\boldsymbol{Y} - X\hat{\boldsymbol{\beta}}) \\ &= \sum_{j=1}^{n} \left\{ Y_j - \boldsymbol{x}_j^{\mathrm{T}}\hat{\boldsymbol{\beta}} \right\}^2 \\ &= \sum_{j=1}^{n} r_j^2 \,,\end{aligned}$$

where

$$r_j = Y_j - \boldsymbol{x}_j^{\mathrm{T}}\hat{\boldsymbol{\beta}} \,.$$

Often, a comparison is made to the null model

$$Y_j = \beta_0 + \varepsilon_j \,,$$

i.e. $Y_i \sim N(\beta_0, \sigma^2)$. The residual sum of squares for the null model is given by

$$\text{RSS(null)} = \boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y} - n\bar{Y}^2 \,,$$

as

$$\hat{\beta}_0 = \bar{Y} = \frac{1}{n}\sum_{j=1}^{n} Y_j \,.$$

An Analysis of variance (ANOVA) table is compact way of presenting the results of (sequential) comparisons of nested models. You should be familiar with an ANOVA table of the following form.

Table 1.1: A standard ANOVA table.

| Source | Degress of Freedom | (Sequential) Sum of Squares | Mean Square |
|---|---|---|---|
| Regression | $p-1$ | By subtraction; see (1.14) | Reg SS$/(p-1)$ |
| Residual | $n-p$ | $(\boldsymbol{Y} - X\hat{\boldsymbol{\beta}})^{\mathrm{T}}(\boldsymbol{Y} - X\hat{\boldsymbol{\beta}})$[5] | RSS$/(n-p)$ |
| Total | $n-1$ | $\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y} - n\bar{Y}^2$[6] | |

[5] Residual sum of squares for the full regression model.
[6] Residual sum of squares for the null model.

In row 1 of Table 1.1 above,

$$\text{Regression SS} = \text{Total SS} - \text{RSS} = \boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y} - n\bar{Y}^2 - (\boldsymbol{Y} - X\hat{\boldsymbol{\beta}})^{\mathrm{T}}(\boldsymbol{Y} - X\hat{\boldsymbol{\beta}}) \tag{1.12}$$

$$= -n\bar{Y}^2 - \hat{\boldsymbol{\beta}}^{\mathrm{T}}(X^{\mathrm{T}}X)\hat{\boldsymbol{\beta}} + 2\hat{\boldsymbol{\beta}}^{\mathrm{T}}X^{\mathrm{T}}\boldsymbol{Y} \tag{1.13}$$

$$= \hat{\boldsymbol{\beta}}^{\mathrm{T}}(X^{\mathrm{T}}X)\hat{\boldsymbol{\beta}} - n\bar{Y}^2\,, \tag{1.14}$$

with the last line following from

$$\begin{aligned}\hat{\boldsymbol{\beta}}^{\mathrm{T}}X^{\mathrm{T}}\boldsymbol{Y} &= \hat{\boldsymbol{\beta}}^{\mathrm{T}}(X^{\mathrm{T}}X)(X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}\boldsymbol{Y} \\ &= \hat{\boldsymbol{\beta}}^{\mathrm{T}}(X^{\mathrm{T}}X)\hat{\boldsymbol{\beta}}\end{aligned}$$

This idea can be generalised to the comparison of a *sequence* of nested models - see Problem Sheet 1.

Hypothesis testing is performed using the mean square:

$$\frac{\text{Regression SS}}{p-1} = \frac{\hat{\boldsymbol{\beta}}^{\mathrm{T}}(X^{\mathrm{T}}X)\hat{\boldsymbol{\beta}} - n\bar{Y}^2}{p-1}\,.$$

Under $\mathrm{H}_0 : \beta_1 = \cdots = \beta_{p-1} = 0$

$$\begin{aligned}\frac{\text{Regression SS}/(p-1)}{\text{RSS}/(n-p)} &= \frac{(\hat{\boldsymbol{\beta}}^{\mathrm{T}}(X^{\mathrm{T}}X)\hat{\boldsymbol{\beta}} - n\bar{Y}^2)/(p-1)}{(\boldsymbol{Y} - X\hat{\boldsymbol{\beta}})^{\mathrm{T}}(\boldsymbol{Y} - X\hat{\boldsymbol{\beta}})/(n-p)} \\ &\sim F_{p-1,n-p}\,,\end{aligned}$$

an $F$ distribution with $p-1$ and $n-p$ degrees of freedom; defined via the ratio of two independent $\chi^2$ distributions.

Also,

$$\frac{\text{RSS}}{n-p} = \frac{(\boldsymbol{Y} - X\hat{\boldsymbol{\beta}})^{\mathrm{T}}(\boldsymbol{Y} - X\hat{\boldsymbol{\beta}})}{n-p} = \hat{\sigma}^2$$

is an unbiased estimator for $\sigma^2$, and

$$\frac{(n-p)}{\sigma^2}\hat{\sigma}^2 \sim \chi^2_{n-p}\,.$$

This is a Chi-squared distribution with $n-p$ degrees of freedom (see MATH2010 or MATH6174 notes).

## 1.6 Exercises

1. (Adapted from Morris, 2011) A classic and famous example of a simple hypothetical experiment was described by Fisher (1935):

   > A lady declares that by tasting a cup of tea made with milk she can discriminate whether the milk or the tea infusion was added first to the cup. We will consider the problem of designing an experiment by means of which this assertion can be tested. For this purpose let us first lay down a simple form of experiment with a view to studying its limitations and its characteristics, both those that same essential to the experimental method, when well developed, and those that are not essential but auxiliary.
   >
   > Our experiment consists in mixing eight cups of tea, four in one way and four in the other, and presenting them to the subject for judgement in a random order. The subject has been told in advance of what the test will consist, namely that she will be asked to taste eight cups, that these shall be four of each kind, and that they shall be presented to her in a random order, that is an order not determined arbitrarily by human choice, but by the actual manipulation of the physical appartatus used in games of chance, cards, dice, roulettes, etc., or, more expeditiously, from a published collection of random sampling numbers purporting to give the actual results of such manipulation[7]. Her task is to divide the 8 cups into two sets of 4, agreeing, if possible, with the treatments received.

   a. Define the treatments in this experiment.
   b. Identify the units in this experiment.
   c. How might a "physical appartatus" from a "game of chance" be used to perform the randomisation? Explain one example.
   d. Suppose eight tea cups are available for this experiment but they are not identical. Instead they come from two sets. Four are made from heavy, thick porcelain; four from much lighter china. If each cup can only be used once, how might this fact be incorporated into the design of the experiment?

Solution

a. There are two treatments in the experiment - the two ingredients "milk first" and "tea first".

b. The experimental units are the "cups of tea", made up from the tea and milk used and also the cup itself.

c. The simplest method here might be to select four black playing cards and four red playing cards, assign one treatment to each colour, shuffle

[7] Now, we would use routines such as `sample` in `R`.

the cards, and then draw them in order. The colour drawn indicates the treatment that should be used to make the next cup of tea. This operation would give one possible randomisation.

We could of course also use R.

```
sample(rep(c("Milk first", "Tea first"), c(4, 4)), size = 8, replace = F)
```

```
## [1] "Milk first" "Tea first"  "Milk first" "Milk first" "Tea first"
## [6] "Tea first"  "Milk first" "Tea first"
```

d. Type of cup could be considered as a blocking factor. One way of incorporating it would be to split the experiment into two (blocks), each with four cups (two milk first, two tea first). We would still wish to randomise allocation of treatments to units within blocks.

```
# block 1
sample(rep(c("Milk first", "Tea first"), c(2, 2)), size = 4, replace = F)
```

```
## [1] "Milk first" "Tea first"  "Milk first" "Tea first"
```

```
# block 2
sample(rep(c("Milk first", "Tea first"), c(2, 2)), size = 4, replace = F)
```

```
## [1] "Tea first"  "Milk first" "Tea first"  "Milk first"
```

2. Consider the linear model

$$\boldsymbol{y} = X\boldsymbol{\beta} + \boldsymbol{\varepsilon}\,,$$

with $\boldsymbol{y}$ an $n \times 1$ vector of responses, $X$ a $n \times p$ model matrix and $\boldsymbol{\varepsilon}$ a $n \times 1$ vector of independent and identically distributed random variables with constant variance $\sigma^2$.

a. Derive the least squares estimator $\hat{\boldsymbol{\beta}}$ for this multiple linear regression model, and show that this estimator is unbiased. Using the definition of (co)variance, show that

$$\mathrm{Var}(\hat{\boldsymbol{\beta}}) = \left(X^{\mathrm{T}}X\right)^{-1}\sigma^2\,.$$

b. If $\boldsymbol{\varepsilon} \sim N(\mathbf{0}, I_n\sigma^2)$, with $I_n$ being the $n \times n$ identity matrix, show that the maximum likelihood estimators for $\boldsymbol{\beta}$ coincide with the least squares estimators.

Solution

a. The method of least squares minimises the sum of squared differences between the responses and the expected values, that is, minimises the expression

$$(\boldsymbol{y} - X\boldsymbol{\beta})^{\mathrm{T}}(\boldsymbol{y} - X\boldsymbol{\beta}) = \boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} - 2\boldsymbol{\beta}^{\mathrm{T}}X^{\mathrm{T}}\boldsymbol{y} + \boldsymbol{\beta}^{\mathrm{T}}X^{\mathrm{T}}X\boldsymbol{\beta}\,.$$

Differentiating with respect to the vector $\boldsymbol{\beta}$, we obtain

$$\frac{\partial}{\partial \boldsymbol{\beta}} = -2X^{\mathrm{T}}\boldsymbol{y} + 2X^{\mathrm{T}}X\boldsymbol{\beta}\,.$$

Set equal to $\mathbf{0}$ and solve:

$$X^{\mathrm{T}}X\hat{\boldsymbol{\beta}} = X^{\mathrm{T}}\boldsymbol{y} \Rightarrow \hat{\boldsymbol{\beta}} = \left(X^{\mathrm{T}}X\right)^{-1}X^{\mathrm{T}}\boldsymbol{y}\,.$$

The estimator $\hat{\boldsymbol{\beta}}$ is unbiased:

$$E(\hat{\boldsymbol{\beta}}) = \left(X^{\mathrm{T}}X\right)^{-1}X^{\mathrm{T}}E(\boldsymbol{y}) = \left(X^{\mathrm{T}}X\right)^{-1}X^{\mathrm{T}}X\boldsymbol{\beta} = \boldsymbol{\beta}\,,$$

and has variance:

$$\begin{aligned}
\mathrm{Var}(\hat{\boldsymbol{\beta}}) &= E\left\{\left[\hat{\boldsymbol{\beta}} - E(\hat{\boldsymbol{\beta}})\right]\left[\hat{\boldsymbol{\beta}} - E(\hat{\boldsymbol{\beta}})\right]^{\mathrm{T}}\right\} \\
&= E\left\{\left[\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\right]\left[\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\right]^{\mathrm{T}}\right\} \\
&= E\left\{\hat{\boldsymbol{\beta}}\hat{\boldsymbol{\beta}}^{\mathrm{T}} - 2\boldsymbol{\beta}\hat{\boldsymbol{\beta}}^{\mathrm{T}} + \boldsymbol{\beta}\boldsymbol{\beta}^{\mathrm{T}}\right\} \\
&= E\left\{\left(X^{\mathrm{T}}X\right)^{-1}X^{\mathrm{T}}\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}X\left(X^{\mathrm{T}}X\right)^{-1} - 2\boldsymbol{\beta}\boldsymbol{y}^{\mathrm{T}}X\left(X^{\mathrm{T}}X\right)^{-1} + \boldsymbol{\beta}\boldsymbol{\beta}^{\mathrm{T}}\right\} \\
&= \left(X^{\mathrm{T}}X\right)^{-1}X^{\mathrm{T}}E(\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}})X\left(X^{\mathrm{T}}X\right)^{-1} - 2\boldsymbol{\beta}E(\boldsymbol{y}^{\mathrm{T}})X\left(X^{\mathrm{T}}X\right)^{-1} + \boldsymbol{\beta}\boldsymbol{\beta}^{\mathrm{T}} \\
&= \left(X^{\mathrm{T}}X\right)^{-1}X^{\mathrm{T}}\left[\mathrm{Var}(\boldsymbol{y}) + E(\boldsymbol{y})E(\boldsymbol{y}^{\mathrm{T}})\right]X\left(X^{\mathrm{T}}X\right)^{-1} - 2\boldsymbol{\beta}\boldsymbol{\beta}^{\mathrm{T}}X^{\mathrm{T}}X\left(X^{\mathrm{T}}X\right)^{-1} + \boldsymbol{\beta}\boldsymbol{\beta}^{\mathrm{T}} \\
&= \left(X^{\mathrm{T}}X\right)^{-1}X^{\mathrm{T}}\left[I_N\sigma^2 + X\boldsymbol{\beta}\boldsymbol{\beta}^{\mathrm{T}}X^{\mathrm{T}}\right]X\left(X^{\mathrm{T}}X\right)^{-1} - \boldsymbol{\beta}\boldsymbol{\beta}^{\mathrm{T}} \\
&= \left(X^{\mathrm{T}}X\right)^{-1}\sigma^2\,.
\end{aligned}$$

b. As $\boldsymbol{y} \sim N\left(X\boldsymbol{\beta}, I_N\sigma^2\right)$, the likelihood is given by

$$L(\boldsymbol{\beta}\,;\boldsymbol{y}) = \left(2\pi\sigma^2\right)^{-N/2}\exp\left\{-\frac{1}{2\sigma^2}(\boldsymbol{y} - X\boldsymbol{\beta})^{\mathrm{T}}(\boldsymbol{y} - X\boldsymbol{\beta})\right\}\,.$$

The log-likelihood is given by

$$l(\boldsymbol{\beta}\,;\boldsymbol{y}) = -\frac{1}{2\sigma^2}(\boldsymbol{y} - X\boldsymbol{\beta})^{\mathrm{T}}(\boldsymbol{y} - X\boldsymbol{\beta}) + \text{constant}\,.$$

Up to a constant, this expression is $-1\times$ the least squares equations; hence maximising the log-likelihood is equivalent to minimising the least squares equation.

3. Consider the two nested linear models

   (i) $Y_j = \beta_0 + \beta_1 x_{1j} + \beta_2 x_{2j} + \ldots + \beta_{p_1} x_{p_1 j} + \varepsilon_j$, or $\boldsymbol{y} = X_1 \boldsymbol{\beta}_1 + \boldsymbol{\varepsilon}$,

   (ii) $Y_j = \beta_0 + \beta_1 x_{1j} + \beta_2 x_{2j} + \ldots + \beta_{p_1} x_{p_1 j} + \beta_{p_1+1} x_{(p_1+1)j} + \ldots + \beta_{p_1+p_2} x_{p_1+p_2 j} + \varepsilon_j$, or $\boldsymbol{y} = X_1 \boldsymbol{\beta}_1 + X_2 \boldsymbol{\beta}_2 + \boldsymbol{\varepsilon}$

   with $\varepsilon_j \sim N(0, \sigma^2)$, and $\varepsilon_j$, $\varepsilon_k$ independent ($\boldsymbol{\varepsilon} \sim N(\mathbf{0}, I_n \sigma^2)$).

   a. Construct an ANOVA table to compare model (ii) with the null model $Y_j = \beta_0 + \varepsilon_j$.

   b. Extend this ANOVA table to compare models (i) and (ii) by further decomposing the regression sum of squares for model (ii).

   **Hint:** which residual sum of squares are you interested in to compare models (i) and (ii)?

   You should end up with an ANOVA table of the form

| Source | Degrees of freedom | Sums of squares | Mean square |
|---|---|---|---|
| Model (i) | $p_1$ | ? | ? |
| Model (ii) | $p_2$ | ? | ? |
| Residual | $n - p_1 - p_2 - 1$ | ? | ? |
| Total | $n - 1$ | $\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} - n\bar{Y}^2$ | |

   The second row of the table gives the **extra sums of squares** for the additional terms in fitting model (ii), over and above those in model (i).

   c. Calculate the extra sum of squares for fitting the terms in model (i), over and above those terms only in model (ii), i.e. those held in $X_2\boldsymbol{\beta}_2$. Construct an ANOVA table containing both the extra sum of squares for the terms only in model (i) and the extra sum of squares for the terms only in model (ii). Comment on the table.

Solution

a. From lectures

| Source | Degrees of freedom | Sums of squares | Mean square |
|---|---|---|---|
| Regression | $p_1 + p_2$ | $\hat{\boldsymbol{\beta}}^{\mathrm{T}} X^{\mathrm{T}} X \hat{\boldsymbol{\beta}} - n\bar{Y}^2$ | $\left(\hat{\boldsymbol{\beta}}^{\mathrm{T}} X^{\mathrm{T}} X \hat{\boldsymbol{\beta}} - n\bar{Y}^2\right)/(p_1+p_2)$ |
| Residual | $n - p_1 - p_2 - 1$ | $(\boldsymbol{y} - X\hat{\boldsymbol{\beta}})^{\mathrm{T}}(\boldsymbol{y} - X\hat{\boldsymbol{\beta}})$ | $\mathrm{RSS}/(n - p_1 - p_2 - 1)$ |
| Total | $n - 1$ | $\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} - n\bar{Y}^2$ | |

where

$$\begin{aligned}\text{RSS(null) - RSS(ii)} &= \boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} - n\bar{Y}^2 - (\boldsymbol{y} - X\hat{\boldsymbol{\beta}})^{\mathrm{T}}(\boldsymbol{y} - X\hat{\boldsymbol{\beta}}) \\ &= \boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} - n\bar{Y}^2 - \boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} + 2\boldsymbol{y}^{\mathrm{T}}X\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}^{\mathrm{T}}X^{\mathrm{T}}X\hat{\boldsymbol{\beta}} \\ &= 2\hat{\boldsymbol{\beta}}^{\mathrm{T}}X^{\mathrm{T}}X\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}^{\mathrm{T}}X^{\mathrm{T}}X\hat{\boldsymbol{\beta}} - n\bar{Y}^2 \\ &= \hat{\boldsymbol{\beta}}^{\mathrm{T}}X^{\mathrm{T}}X\hat{\boldsymbol{\beta}} - n\bar{Y}^2 \,.\end{aligned}$$

b. To compare model (i) with the null model, we compute

$$\begin{aligned}\text{RSS(null) - RSS(i)} &= \boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} - N\bar{Y}^2 - (\boldsymbol{y} - X_1\hat{\boldsymbol{\beta}}_1)^{\mathrm{T}}(\boldsymbol{y} - X_1\hat{\boldsymbol{\beta}}_1) \\ &= \hat{\boldsymbol{\beta}}_1^{\mathrm{T}}X_1^{\mathrm{T}}X_1\hat{\boldsymbol{\beta}}_1 - n\bar{Y}^2 \,.\end{aligned}$$

To compare models (i) and (ii), we compare them both to the null model, and look at the difference between these comparisons:

$$\begin{aligned}\text{[RSS(null) - RSS(ii)] - [RSS(null) - RSS(i)]} &= \text{RSS(i) - RSS(ii)} \\ &= (\boldsymbol{y} - X_1\hat{\boldsymbol{\beta}}_1)^{\mathrm{T}}(\boldsymbol{y} - X_1\hat{\boldsymbol{\beta}}_1) - (\boldsymbol{y} - X\hat{\boldsymbol{\beta}})^{\mathrm{T}}(\boldsymbol{y} - X\hat{\boldsymbol{\beta}}) \\ &= \hat{\boldsymbol{\beta}}^{\mathrm{T}}X^{\mathrm{T}}X\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_1^{\mathrm{T}}X_1^{\mathrm{T}}X_1\hat{\boldsymbol{\beta}}_1 \,.\end{aligned}$$

| Source | Degrees of freedom | Sums of squares | Mean square |
|---|---|---|---|
| Regression | $p_1 + p_2$ | $\hat{\boldsymbol{\beta}}X^{\mathrm{T}}X\hat{\boldsymbol{\beta}} - n\bar{Y}^2$ | $\left(\hat{\boldsymbol{\beta}}X^{\mathrm{T}}X\hat{\boldsymbol{\beta}} - n\bar{Y}^2\right)/(p_1+p_2)$ |
| Model (i) | $p_1$ | $\hat{\boldsymbol{\beta}}_1^{\mathrm{T}}X_1^{\mathrm{T}}X_1\hat{\boldsymbol{\beta}}_1 - n\bar{Y}^2$ | $\left(\hat{\boldsymbol{\beta}}_1^{\mathrm{T}}X_1^{\mathrm{T}}X_1\hat{\boldsymbol{\beta}}_1 - n\bar{Y}^2\right)/p_1$ |
| Extra due to Model (ii) | $p_2$ | $\hat{\boldsymbol{\beta}}^{\mathrm{T}}X^{\mathrm{T}}X\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_1^{\mathrm{T}}X_1^{\mathrm{T}}X_1\hat{\boldsymbol{\beta}}_1$ | $\left(\hat{\boldsymbol{\beta}}^{\mathrm{T}}X^{\mathrm{T}}X\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_1^{\mathrm{T}}X_1^{\mathrm{T}}X_1\hat{\boldsymbol{\beta}}_1\right)/p_2$ |
| Residual | $n - p_1 - p_2 - 1$ | $(\boldsymbol{y} - X\hat{\boldsymbol{\beta}})^{\mathrm{T}}(\boldsymbol{y} - X\hat{\boldsymbol{\beta}})$ | RSS$/(n - p_1 - p_2 - 1)$ |
| Total | $n - 1$ | $\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} - n\bar{Y}^2$ | |

By definition, the Model (i) SS and the Extra SS for Model (ii) sum to the Regression SS.

c. The extra sum of squares for the terms in model (i) over and above those in model (ii) are obtained through comparison of the models

ia. $\boldsymbol{y} = X_2\boldsymbol{\beta}_2 + \boldsymbol{\varepsilon}$,

iia. $\boldsymbol{y} = X_1\boldsymbol{\beta}_1 + X_2\boldsymbol{\beta}_2 + \boldsymbol{\varepsilon} = X\boldsymbol{\beta} + \varepsilon$

Extra sum of squares for model (iia):

$$\begin{aligned}\text{[RSS(null) - RSS(iia)] - [RSS(null) - RSS(ia)]} &= \text{RSS(ia) - RSS(iia)} \\ &= (\boldsymbol{y} - X_2\hat{\boldsymbol{\beta}}_2)^{\mathrm{T}}(\boldsymbol{y} - X_2\hat{\boldsymbol{\beta}}_2) - (\boldsymbol{y} - X\hat{\boldsymbol{\beta}})^{\mathrm{T}}(\boldsymbol{y} - X\hat{\boldsymbol{\beta}}) \\ &= \hat{\boldsymbol{\beta}}^{\mathrm{T}}X^{\mathrm{T}}X\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_2^{\mathrm{T}}X_2^{\mathrm{T}}X_2\hat{\boldsymbol{\beta}}_2\,.\end{aligned}$$

Hence, an ANOVA table for the extra sums of squares is given by

| Source | Degrees of freedom | Sums of squares | Mean square |
|---|---|---|---|
| Regression | $p_1 + p_2$ | $\hat{\boldsymbol{\beta}}X^{\mathrm{T}}X\hat{\boldsymbol{\beta}} - n\bar{Y}^2$ | $\left(\hat{\boldsymbol{\beta}}X^{\mathrm{T}}X\hat{\boldsymbol{\beta}} - n\bar{Y}^2\right)/(p_1+p_2)$ |
| Extra Model (i) | $p_1$ | $\hat{\boldsymbol{\beta}}^{\mathrm{T}}X^{\mathrm{T}}X\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_2^{\mathrm{T}}X_2^{\mathrm{T}}X_2\hat{\boldsymbol{\beta}}_2$ | $\left(\hat{\boldsymbol{\beta}}^{\mathrm{T}}X^{\mathrm{T}}X\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_2^{\mathrm{T}}X_2^{\mathrm{T}}X_2\hat{\boldsymbol{\beta}}_2\right)/p_1$ |
| Extra Model (ii) | $p_2$ | $\hat{\boldsymbol{\beta}}^{\mathrm{T}}X^{\mathrm{T}}X\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_1^{\mathrm{T}}X_1^{\mathrm{T}}X_1\hat{\boldsymbol{\beta}}_1$ | $\left(\hat{\boldsymbol{\beta}}^{\mathrm{T}}X^{\mathrm{T}}X\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_1^{\mathrm{T}}X_1^{\mathrm{T}}X_1\hat{\boldsymbol{\beta}}_1\right)/p_2$ |
| Residual | $n - p_1 - p_2 - 1$ | $(\boldsymbol{y} - X\hat{\boldsymbol{\beta}})^{\mathrm{T}}(\boldsymbol{y} - X\hat{\boldsymbol{\beta}})$ | $\text{RSS}/(n - p_1 - p_2 - 1)$ |
| Total | $n - 1$ | $\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} - n\bar{Y}^2$ | |

Note that for these *adjusted* sums of squares, in general the extra sum of squares for model (i) and (ii) do not sum to the regression sum of squares. This will only be the case if the columns of $X_1$ and $X_2$ are mutually orthogonal, i.e. $X_1^{\mathrm{T}}X_2 = \mathbf{0}$.

# Chapter 2

# Completely randomised designs

The simplest form of experiment we will consider compares $t$ different **unstructured** treatments. By unstructured, we mean the treatments form a discrete collection, not related through the settings of other experimental features (compare with factorial experiments in Chapter 4). We also make the assumption that there are no restrictions in the randomisation of treatments to experimental units (compare with Chapter 3 on blocking). A designs for such an experiment is therefore called a **completely randomised design** (CRD).

**Example 2.1.** Pulp experiment (Wu and Hamada, 2009, ch. 2)

In a paper pulping mill, an experiment was run to examine differences between the reflectance (brightness; ratio of amount of light leaving a target to the amount of light striking the target) of sheets of pulp made by $t = 4$ operators. The data are given in Table 2.1 below.

```
pulp <- data.frame(operator = rep(factor(1:4), 5),
                   repetition = rep(1:5, rep(4, 5)),
                   reflectance = c(59.8, 59.8, 60.7, 61.0, 60.0, 60.2, 60.7, 60.8,
                                   60.8, 60.4, 60.5, 60.6, 60.8, 59.9, 60.9, 60.5, 59.8, 60.0, 6
                     )
knitr::kable(
 tidyr::pivot_wider(pulp, names_from = operator, values_from = reflectance)[, -1],
 col.names = paste("Operator", 1:4),
 caption = "Pulp experiment: reflectance values (unitless) from four different operators."
)
```

The experiment has one factor (operator) with four levels (sometimes called a one-way layout). The CRD employed has equal replication of each treatment (operator).

Table 2.1: Pulp experiment: reflectance values (unitless) from four different operators.

| Operator 1 | Operator 2 | Operator 3 | Operator 4 |
|---|---|---|---|
| 59.8 | 59.8 | 60.7 | 61.0 |
| 60.0 | 60.2 | 60.7 | 60.8 |
| 60.8 | 60.4 | 60.5 | 60.6 |
| 60.8 | 59.9 | 60.9 | 60.5 |
| 59.8 | 60.0 | 60.3 | 60.5 |

We can informally compare the responses from these four treatments graphically.

```
boxplot(reflectance ~ operator, data = pulp)
```

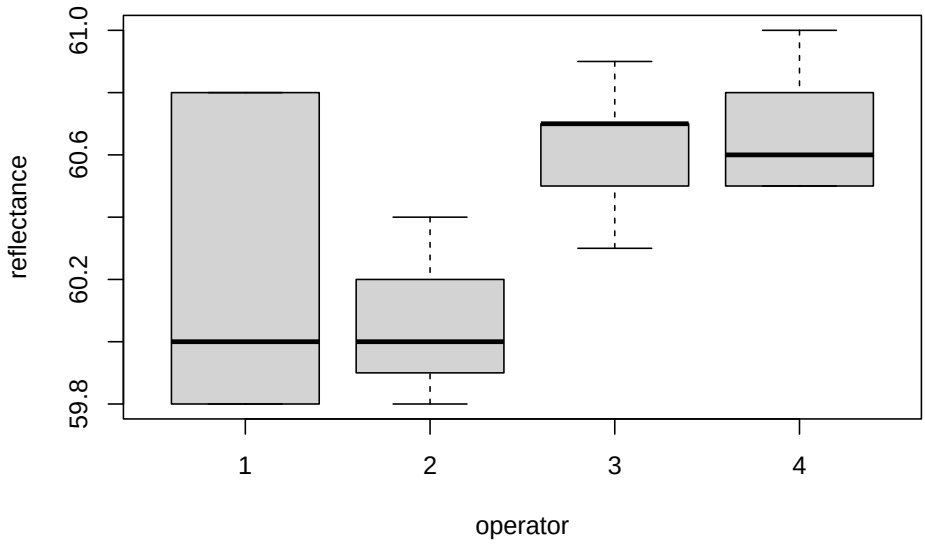



Figure 2.1: Pulp experiment: distributions of reflectance from the four operators.

Figure 2.1 shows that, relative to the variation, there may be a difference in the mean response between treatments 1 and 2, and 3 and 4. In this chapter, we will see how to make this comparison formally using linear models, and to assess how the choice of design impacts our results.

Throughout this chapter we will assume the $i$th treatment is applied to $n_i$ experimental unit, with total number of runs $n = \sum_{i=1}^{t} n_i$ in the experiment.

## 2.1 A unit-treatment linear model

An appropriate, and common, model to describe data from such experiments when the response is continuous is given by

$$y_{ij} = \mu + \tau_i + \varepsilon_{ij}\,, \quad i = 1, \ldots, t; j = 1, \ldots, n_i\,, \tag{2.1}$$

where $y_{ij}$ is the response from the $j$th application of treatment $i$, $\mu$ is a constant parameter, $\tau_i$ is the effect of the $i$th treatment, and $\varepsilon_{ij}$ is the random individual effect from each experimental unit with $E(\varepsilon_{ij}) = 0$ and $\mathrm{Var}(\varepsilon_{ij}) = \sigma^2$. All random errors are assumed independent and here we also assume $\varepsilon_{ij} \sim N(0, \sigma^2)$.

Model (2.1) assumes that each treatment can be randomly allocated to one of the $n$ experimental units, and that the response observed is independent of the allocation of all the other treatments (the stable unit treatment value assumption or SUTVA).

Why is this model appropriate and commonly used? The expected response from the application of the $i$th treatment is

$$E(y_{ij}) = \mu + \tau_i\,.$$

The parameter $\mu$ can be thought of as representing the impact of many different features particular to **this** experiment but common to all units, and $\tau_i$ is the deviation due to applying treatment $i$. From the applicable of two different hypothetical experiments, A and B, the expected response from treatment $i$ may be different due to a different overall mean. From experiment A:

$$E(y_{ij}) = \mu_{\mathrm{A}} + \tau_i\,.$$

From experiment B:

$$E(y_{ij}) = \mu_{\mathrm{B}} + \tau_i\,.$$

But the **difference** between treatments $k$ and $l$ $(k, l = 1, \ldots, t)$

$$\begin{aligned} E(y_{kj}) - E(y_{lj}) &= \mu_A + \tau_k - \mu_A - \tau_l \\ &= \tau_k - \tau_l\,, \end{aligned}$$

is constant across different experiments. This concept of **comparison** underpins most design of experiments, and will be applied throughout this module.

## 2.2 The partitioned linear model

In matrix form, we can write model (2.1) as

$$\boldsymbol{y} = X_1\mu + X_2\boldsymbol{\tau} + \boldsymbol{\varepsilon}\,,$$

where $X_1 = \mathbf{1}_n$, the $n$-vector with every entry equal to one,

$$X_2 = \bigoplus_{i=1}^{t} \mathbf{1}_{n_i} = \begin{bmatrix} \mathbf{1}_{n_1} & \mathbf{0}_{n_1} & \cdots & \mathbf{0}_{n_1} \\ \mathbf{0}_{n_2} & \mathbf{1}_{n_2} & \cdots & \mathbf{0}_{n_2} \\ \vdots & & \ddots & \vdots \\ \mathbf{0}_{n_t} & \mathbf{0}_{n_t} & \cdots & \mathbf{1}_{n_t} \end{bmatrix},$$

with $\bigoplus$ denoting "direct sum", $\mathbf{0}_{n_i}$ is the $n_i$-vector with every entry equal to zero, $\boldsymbol{\tau} = [\tau_1, \ldots, \tau_t]^{\mathrm{T}}$ and $\boldsymbol{\varepsilon} = [\varepsilon_{11}, \ldots, \varepsilon_{tn_t}]^{\mathrm{T}}$.

Why are we partitioning the model? Going back to our discussion of the role of $\mu$ and $\tau_i$, it is clear that we not interested in estimating $\mu$, which represents an experiment-specific contribution to the expected mean. Our only interest is in estimating the (differences between the) $\tau_i$. Hence, we can treat $\mu$ as a nuisance parameter.

If we define $X = [X_1 \,|\, X_2]$ and $\boldsymbol{\beta}^{\mathrm{T}} = [\mu|\boldsymbol{\tau}^{\mathrm{T}}]$, we can write the usual least squares equations

$$X^{\mathrm{T}}X\hat{\boldsymbol{\beta}} = X^{\mathrm{T}}\boldsymbol{y} \tag{2.2}$$

as a system of two matrix equations

$$\begin{aligned} X_1^{\mathrm{T}}X_1\hat{\mu} + X_1^{\mathrm{T}}X_2\hat{\boldsymbol{\tau}} &= X_1^{\mathrm{T}}\boldsymbol{y} \\ X_2^{\mathrm{T}}X_1\hat{\mu} + X_2^{\mathrm{T}}X_2\hat{\boldsymbol{\tau}} &= X_2^{\mathrm{T}}\boldsymbol{y}\,. \end{aligned}$$

Assuming $(X_1^{\mathrm{T}}X_1)^{-1}$ exists, which it does in this case, we can pre-multiply the first of these equations by $X_2^{\mathrm{T}}X_1(X_1^{\mathrm{T}}X_1)^{-1}$ and subtract it from the second equation to obtain

$$\begin{aligned} X_2^{\mathrm{T}}[I_n - X_1(X_1^{\mathrm{T}}X_1)^{-1}X_1^{\mathrm{T}}]X_1\hat{\mu} + X_2^{\mathrm{T}}[I_n - X_1(X_1^{\mathrm{T}}X_1)^{-1}X_1^{\mathrm{T}}]X_2\hat{\boldsymbol{\tau}} \\ = X_2^{\mathrm{T}}[I_n - X_1(X_1^{\mathrm{T}}X_1)^{-1}X_1^{\mathrm{T}}]\boldsymbol{y}\,. \end{aligned}$$

Writing $H_1 = X_1(X_1^{\mathrm{T}}X_1)^{-1}X_1^{\mathrm{T}}$, we obtain

$$X_2^{\mathrm{T}}[I_n - H_1]X_1\hat{\mu} + X_2^{\mathrm{T}}[I_n - H_1]X_2\hat{\boldsymbol{\tau}} = X_2^{\mathrm{T}}[I_n - H_1]\boldsymbol{y}\,. \tag{2.3}$$

The matrix $H_1$ is a "hat" matrix for a linear model containing only the term $\mu$, and hence $H_1X_1 = X_1$ (see MATH2010 or STAT6123). Hence the first term in (2.3) is zero, and we obtain the **reduced normal equations** for $\boldsymbol{\tau}$:

$$X_2^{\mathrm{T}}[I_n - H_1]X_2\hat{\boldsymbol{\tau}} = X_2^{\mathrm{T}}[I_n - H_1]\boldsymbol{y}\,. \tag{2.4}$$

Note that the solutions from (2.4) are not different from the solution to $\hat{\boldsymbol{\tau}}$ that would be obtained from solving (2.2); equation (2.4) is simply a re-expression, where we have eliminated the nuisance parameter $\mu$. This fact means that we rarely need to solve (2.4) explicitly.

Recalling that for a hat matrix, $I_n - H_1$ is idempotent and symmetric (see MATH2010 or MATH6174), if we define

$$X_{2|1} = (I_n - H_1)X_2\,,$$

then we can rewrite equation (2.4) as

$$X_{2|1}^{\mathrm{T}}X_{2|1}\hat{\boldsymbol{\tau}} = X_{2|1}^{\mathrm{T}}\boldsymbol{y}\,, \tag{2.5}$$

which are the normal equations for a linear model with expectation $E(\boldsymbol{y}) = X_{2|1}\boldsymbol{\tau}$.

## 2.3 Reduced normal equations for the CRD

For the CRD discussed in this chapter, $X_1^{\mathrm{T}}X_1 = n$, the total number of runs in the experiment[1]. Hence $(X_1^{\mathrm{T}}X_1)^{-1} = 1/n$ and $H_1 = \frac{1}{n}J_n$, with $J_n$ the $n \times n$ matrix with all entries equal to 1.

The adjusted model matrix then has the form

$$\begin{aligned} X_{2|1} &= (I_n - H_1)X_2 \\ &= X_2 - \frac{1}{n}J_nX_2 \\ &= X_2 - \frac{1}{n}[n_1\mathbf{1}_n|\cdots|n_t\mathbf{1}_n]\,. \end{aligned} \tag{2.6}$$

That is, every column of $X_2$ has been adjusted by the subtraction of the column mean from each entry[2]. Also notice that each row of $X_{2|1}$ has a row-sum equal

[1] In later chapters we will see examples where $X_1$ has $> 1$ columns, and hence $X_1^{\mathrm{T}}X_1$ is a matrix.

[2] Often called "column centred"

to zero ($= 1 - \sum_{i=1}^{t} n_t/n$). Hence, $X_{2|1}$ is not of full column rank, and so the reduced normal equations do not have a unique solution[3].

Although (2.5), and hence, (2.2), have no unique solution[4], it can be shown that both $\widehat{X_{2|1}\boldsymbol{\tau}}$ and $\widehat{X\boldsymbol{\beta}}$ have unique solutions. Hence fitted values $\hat{\boldsymbol{y}} = \widehat{X\boldsymbol{\beta}}$ and the residual sum of squares

$$RSS = \left(\boldsymbol{y} - \widehat{X\boldsymbol{\beta}}\right)^{\mathrm{T}} \left(\boldsymbol{y} - \widehat{X\boldsymbol{\beta}}\right)$$

are both uniquely defined for any solution to (2.2). That is, every solution to the normal equations leads to the same fitted values and residual sum of squares.

In MATH2010 and STAT6123 we fitted models with categorical variables by defining a set of dummy variables and estimating a reduced model. Here, we will take a slightly different approach and study which combinations of parameters from (2.1) are estimable, and in particular which linear combinations of the treatment parameters $\tau_i$ we can estimate.

Let's study equation (2.5) in more detail. We have

$$\begin{aligned} X_{2|1}^{\mathrm{T}} X_{2|1} &= X_2^{\mathrm{T}}(I_n - H_1)X_2 \\ &= X_2^{\mathrm{T}}X_2 - X_2^{\mathrm{T}}H_1X_2 \\ &= \mathrm{diag}(\boldsymbol{n}) - \frac{1}{n}X_2^{\mathrm{T}}J_nX_2 \\ &= \mathrm{diag}(\boldsymbol{n}) - \frac{1}{n}\boldsymbol{n}\boldsymbol{n}^{\mathrm{T}}, \end{aligned}$$

where $\boldsymbol{n}^{\mathrm{T}} = (n_1, \ldots, n_t)$. Hence, the reduced normal equations become

$$\left[\mathrm{diag}(\boldsymbol{n}) - \frac{1}{n}\boldsymbol{n}\boldsymbol{n}^{\mathrm{T}}\right]\hat{\boldsymbol{\tau}} = X_2^{\mathrm{T}}\boldsymbol{y} - \frac{1}{n}X_2^{\mathrm{T}}J_n\boldsymbol{y} \tag{2.7}$$

$$= X_2^{\mathrm{T}}\boldsymbol{y} - \boldsymbol{n}\bar{y}_{\cdot\cdot}, \tag{2.8}$$

where $\bar{y}_{\cdot\cdot} = \frac{1}{n}\sum_{i=1}^{t}\sum_{j=1}^{n_i} y_{ij}$, i.e. the overall average of the observations from the experiment.

From (2.8) we obtain a system of $t$ equations, each having the form

$$\hat{\tau}_i - \hat{\tau}_w = \bar{y}_{i\cdot} - \bar{y}_{\cdot\cdot}, \tag{2.9}$$

[3] If we recalled the material on "dummy" variables from MATH2010 or STAT6123, we would already have realised this.

[4] That is, for any two solutions $\tilde{\boldsymbol{\beta}}_1$ and $\tilde{\boldsymbol{\beta}}_2$, $X\tilde{\boldsymbol{\beta}}_1 = X\tilde{\boldsymbol{\beta}}_2$.

where $\hat{\tau}_w = \frac{1}{n}\sum_{i=1}^{t} n_i\hat{\tau}_i$ and $\bar{y}_{i\cdot} = \frac{1}{n_i}\sum_{j=1}^{n_i} y_{ij}$ $(i = 1, \ldots, t)$.

These $t$ equations are not independent; when multiplied by the $n_i$, they sum to zero due to the linear dependency between the columns of $X_{2|1}$. Hence, there is no unique solution to $\hat{\boldsymbol{\tau}}$ from equation (2.8). However, we can estimate certain linear combinations of the $\tau_i$, called *contrasts*.

## 2.4 Contrasts

**Definition 2.1.** A treatment **contrast** is a linear combination $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{\tau}$ with coefficient vector $\boldsymbol{c}^{\mathrm{T}} = (c_1, \ldots, c_t)$ such that $\boldsymbol{c}^{\mathrm{T}}\mathbf{1} = 0$; that is, $\sum_{i=1}^{t} c_i = 0$.

For example, assume we have $t = 3$ treatments, then the following vectors $\boldsymbol{c}$ all define contrasts:

1. $\boldsymbol{c}^{\mathrm{T}} = (1, -1, 0)$,
2. $\boldsymbol{c}^{\mathrm{T}} = (1, 0, -1)$,
3. $\boldsymbol{c}^{\mathrm{T}} = (0, 1, -1)$.

In fact, they define all $\binom{3}{2} = 3$ pairwise comparisons between treatments. The following are also contrasts:

4. $\boldsymbol{c}^{\mathrm{T}} = (2, -1, -1)$,
5. $\boldsymbol{c}^{\mathrm{T}} = (0.5, -1, 0.5)$,

each comparing the sum, or average, of expected responses from two treatments to the expected response from the remaining treatment.

The following are not contrasts, as $\boldsymbol{c}^{\mathrm{T}}\mathbf{1} \neq 0$:

6. $\boldsymbol{c}^{\mathrm{T}} = (2, -1, 0)$,
7. $\boldsymbol{c}^{\mathrm{T}} = (1, 0, 0)$,

with the final example once again demonstrating that we cannot estimate the individual $\tau_i$.

## 2.5 Treatment contrast estimators in the CRD

We estimate a treatment contrast $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{\tau}$ in the CRD via linear combinations of equations (2.9):

$$\begin{aligned} &\sum_{i=1}^{t} c_i\hat{\tau}_i - \sum_{i=1}^{t} c_i\hat{\tau}_w = \sum_{i=1}^{t} c_i\bar{y}_{i\cdot} - \sum_{i=1}^{t} c_i\bar{y}_{\cdot\cdot} \\ \Rightarrow &\sum_{i=1}^{t} c_i\hat{\tau}_i = \sum_{i=1}^{t} c_i\bar{y}_{i\cdot}\,, \end{aligned}$$

as $\sum_{i=1}^{t} c_i \hat{\tau}_w = \sum_{i=1}^{t} c_i \bar{y}_{..} = 0$, as $\sum_{i=1}^{t} c_i = 0$. Hence, the unique estimator of the contrast $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{\tau}$ has the form

$$\widehat{\boldsymbol{c}^{\mathrm{T}}\boldsymbol{\tau}} = \sum_{i=1}^{t} c_i \bar{y}_{i.} \,.$$

That is, we estimate the contrast in the treatment effects by the corresponding contrast in the treatment means.

The variance of this estimator is straightforward to obtain:

$$\begin{aligned} \mathrm{var}\left(\widehat{\boldsymbol{c}^{\mathrm{T}}\boldsymbol{\tau}}\right) &= \sum_{i=1}^{t} c_i^2 \mathrm{var}(\bar{y}_{i.}) \\ &= \sigma^2 \sum_{i=1}^{t} c_i^2/n_i \,, \end{aligned}$$

as, under our model assumptions, each $\bar{y}_{i.}$ is an average of independent observations with variance $\sigma^2$. Similarly, from model (2.1) we can derive the distribution of $\widehat{\boldsymbol{c}^{\mathrm{T}}\boldsymbol{\tau}}$ as

$$\widehat{\boldsymbol{c}^{\mathrm{T}}\boldsymbol{\tau}} \sim N\left(\boldsymbol{c}^{\mathrm{T}}\boldsymbol{\tau}, \sigma^2 \sum_{i=1}^{t} c_i^2/n_i\right) .$$

Confidence intervals and hypothesis tests for $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{\tau}$ can be constructed/conducted using this distribution, e.g.

- a $100(1-\alpha)\%$ confidence interval:

$$\boldsymbol{c}^{\mathrm{T}}\boldsymbol{\tau} \in \sum_{i=1}^{t} c_i \bar{y}_{i.} \pm t_{n-t,1-\frac{\alpha}{2}}\, s \sqrt{\sum_{i=1}^{t} c_i^2/n_i} \,,$$

where $t_{n-t,1-\frac{\alpha}{2}}$ is the $1-\frac{\alpha}{2}$ quantile of a $t$-distribution with $n-t$ degrees of freedom and

$$s^2 = \frac{1}{n-t} \sum_{i=1}^{t} \sum_{j=1}^{n_i} (y_{ij} - \bar{y}_{i.})^2 \tag{2.10}$$

is the estimate of $\sigma^2$.

- the hypothesis $H_0 : \boldsymbol{c}^{\mathrm{T}}\boldsymbol{\tau} = 0$ against the two-sided alternative $H_1 : \boldsymbol{c}^{\mathrm{T}}\boldsymbol{\tau} \neq 0$ is rejected using a test of with confidence level $1-\alpha$ if

Table 2.2: Pulp experiment: reflectance values (unitless) from four different operators.

| Operator 1 | Operator 2 | Operator 3 | Operator 4 |
|---|---|---|---|
| 59.8 | 59.8 | 60.7 | 61.0 |
| 60.0 | 60.2 | 60.7 | 60.8 |
| 60.8 | 60.4 | 60.5 | 60.6 |
| 60.8 | 59.9 | 60.9 | 60.5 |
| 59.8 | 60.0 | 60.3 | 60.5 |

$$\frac{|\sum_{i=1}^{t} c_i \bar{y}_{i\cdot}|}{s\sqrt{\sum_{i=1}^{t} c_i^2/n_i}} > t_{n-t,1-\frac{\alpha}{2}}.$$

## 2.6 Analysing CRDs in R

Let's return to Example 2.1.

```
knitr::kable(
 tidyr::pivot_wider(pulp, names_from = operator, values_from = reflectance)[, -1],
 col.names = paste("Operator", 1:4),
 caption = "Pulp experiment: reflectance values (unitless) from four different operators."
)
```

Clearly, we could directly calculate, and then compare, mean responses for each operator. However, there are (at least) two other ways we can proceed which use the fact we are fitting a linear model. These will be useful when we consider more complex models.

1. Using `pairwise.t.test`.

```
with(pulp,
 pairwise.t.test(reflectance, operator, p.adjust.method = 'none'))
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  reflectance and operator
##
##   1     2     3
## 2 0.396 -     -
## 3 0.084 0.015 -
## 4 0.049 0.008 0.775
##
## P value adjustment method: none
```

This function performs hypothesis tests for all pairwise treatment comparisons (with a default confidence level of 0.95). Here we can see that operators 1 and 4, 2 and 3, and 2 and 4 have statistically significant differences.

2. Using `lm` and the `emmeans` package.

```
pulp.lm <- lm(reflectance ~ operator, data = pulp)
anova(pulp.lm)
```

```
## Analysis of Variance Table
##
## Response: reflectance
##           Df Sum Sq Mean Sq F value Pr(>F)
## operator   3   1.34   0.447     4.2  0.023 *
## Residuals 16   1.70   0.106
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
pulp.emm <- emmeans::emmeans(pulp.lm, ~ operator)
pairs(pulp.emm, adjust = 'none')
```

```
##  contrast              estimate    SE df t.ratio p.value
##  operator1 - operator2     0.18 0.206 16   0.873  0.3955
##  operator1 - operator3    -0.38 0.206 16  -1.843  0.0839
##  operator1 - operator4    -0.44 0.206 16  -2.134  0.0486
##  operator2 - operator3    -0.56 0.206 16  -2.716  0.0153
##  operator2 - operator4    -0.62 0.206 16  -3.007  0.0083
##  operator3 - operator4    -0.06 0.206 16  -0.291  0.7748
```

Here, we have first fitted the linear model object. The `lm` function, by default, will have set up dummy variables with the first treatment (operator) as a baseline (see MATH2010 or STAT6123). We then take the intermediate step of calculating the ANOVA table for this experiment, and use an F-test to compare the model accounting for operator differences to the null model; there are differences between operators at the 5% significance level,

The choice of dummy variables in the linear model is unimportant; any set could be used[5], as in the next line we use the `emmeans` function (from the package of the same name) to specify that we are interested in estimating contrasts in the factor `operator` (which specifies our treatments in this experiment). Finally, the `pairs` command performs hypothesis tests for all pairwise comparisons between the four treatments. The results are the same as those obtained from using `pairwise.t.test`.

Our preferred approach is using method 2 (`lm` and `emmeans`), for four main reasons:

[5]Recall that although $\mu$ and $\boldsymbol{\tau}$ are not uniquely estimable, fitted values $\hat{y}_i = \hat{\mu} + \hat{\tau}_i$ **are**, and hence it does not matter which dummy variables we use in `lm`.

a. The function `contrasts` in the `emmeans` package can be used to estimate arbitrary treatment contrasts (see `help("contrast-methods")`).

```
# same as `pairs` above
emmeans::contrast(pulp.emm, "pairwise", adjust = "none")
```

```
##  contrast              estimate    SE df t.ratio p.value
##  operator1 - operator2     0.18 0.206 16   0.873  0.3955
##  operator1 - operator3    -0.38 0.206 16  -1.843  0.0839
##  operator1 - operator4    -0.44 0.206 16  -2.134  0.0486
##  operator2 - operator3    -0.56 0.206 16  -2.716  0.0153
##  operator2 - operator4    -0.62 0.206 16  -3.007  0.0083
##  operator3 - operator4    -0.06 0.206 16  -0.291  0.7748
```

```
# estimating single contrast c = (1, -.5, -.5)
# comparing operator 1 with operators 2 and 3
contrast1v23.emmc <- function(levs)
  data.frame('t1 v avg t2 t3' = c(1, -.5, -.5, 0))
emmeans::contrast(pulp.emm, 'contrast1v23')
```

```
##  contrast       estimate    SE df t.ratio p.value
##  t1.v.avg.t2.t3     -0.1 0.179 16  -0.560  0.5832
```

b. It more easily generalises to the more complicated models we will see in Chapter 3.

c. It explicitly acknowledges that we have fitted a linear model, and so encourages us to check the model assumptions (see Exercise 3).

d. It is straightforward to apply adjustments for multiple comparisons.

## 2.7 Multiple comparisons

When we perform hypothesis testing, we choose the critical region (i.e. the rule that decides if we reject $H_0$) to control the probability of a type I error; that is, we control the probability of incorrectly rejecting $H_0$. If we need to test multiple hypotheses, e.g. to test all pairwise differences, we need to consider the overall probability of incorrectly rejecting **one or more** null hypothesis. This is called the **experiment-wise** or **family-wise** error rate.

For Example 2.1, there are $\binom{4}{2} = 6$ pairwise comparisons. Under the assumption that all tests are independent[6], assuming each individual test has type I error 0.05, the experiment-wise type I error rate is given by:

```
alpha <- 0.05
1 - (1 - alpha)^6
```

```
## [1] 0.2649
```

[6]They aren't, but it simplifies the maths!

An experiment-wise error rate of 0.2649 is substantially greater than 0.05. Hence, we would expect to make many more type I errors than may be desirable. xkcd has a fun example:

```
alpha <- 0.05
1 - (1 - alpha)^20
```

```
## [1] 0.6415
```

Therefore it is usually desirable to maintain some control of the experiment-wise type I error rate. We will consider two methods.

1. The **Bonferroni method**. An upper bound on the experiment-wise type I error rate for testing $k$ hypotheses can be shown to be

$$\begin{aligned} P(\text{wrongly reject at least one of } H_0^1, \ldots, H_0^k) &= P\left(\bigcup_{i=1}^{k} \{\text{wrongly reject } H_0^i\}\right) \\ &\leq \sum_{i=1}^{k} \underbrace{P(\text{wrongly reject } H_0^i)}_{\leq \alpha} \\ &\leq k\alpha\,. \end{aligned}$$

   Hence a *conservative*[7] adjustment for multiple comparisons is to test each hypothesis at size $\alpha/k$, i.e. for the CRD compare to the quantile $t_{n-t,1-\frac{\alpha}{2k}}$ (or multiply each individual p-value by $k$).

   For Example 2.1, we can test all pairwise comparisons, each at size $\alpha/k$ using the `adjustment` argument in `pairs`.

```
pairs(pulp.emm, adjust = 'bonferroni')
```

```
##  contrast              estimate    SE df t.ratio p.value
##  operator1 - operator2     0.18 0.206 16   0.873  1.0000
##  operator1 - operator3    -0.38 0.206 16  -1.843  0.5034
##  operator1 - operator4    -0.44 0.206 16  -2.134  0.2918
##  operator2 - operator3    -0.56 0.206 16  -2.716  0.0915
##  operator2 - operator4    -0.62 0.206 16  -3.007  0.0501
##  operator3 - operator4    -0.06 0.206 16  -0.291  1.0000
##
## P value adjustment: bonferroni method for 6 tests
```

   Now, only one comparison is significant at an experiment-wise type I error rate of $\alpha = 0.05$ (operators 2 and 4).

2. **Tukey's method**. An alternative approach that gives an exact experiment-wise error rate of $100\alpha\%$ compares the $t$ statistic to a critical value from

[7] So the experiment-wise type I error will actually be less than the prescribed $\alpha$

the studentised range distribution[8], given by $\frac{1}{\sqrt{2}}q_{t,n-t,1-\alpha}$ with $q_{t,n-t,1-\alpha}$ the $1-\alpha$ quantile from the studentised range distribution (available in R as qtukey).

For Example 2.1:

```
pairs(pulp.emm)
```

```
##  contrast              estimate    SE df t.ratio p.value
##  operator1 - operator2     0.18 0.206 16   0.873  0.8185
##  operator1 - operator3    -0.38 0.206 16  -1.843  0.2903
##  operator1 - operator4    -0.44 0.206 16  -2.134  0.1845
##  operator2 - operator3    -0.56 0.206 16  -2.716  0.0658
##  operator2 - operator4    -0.62 0.206 16  -3.007  0.0377
##  operator3 - operator4    -0.06 0.206 16  -0.291  0.9911
##
## P value adjustment: tukey method for comparing a family of 4 estimates
```

The default adjustment in the pairs function is the Tukey method. Comparing the p-values for each comparison using unadjusted t-tests, the Boneferroni and Tukey methods:

```
pairs.u <- pairs(pulp.emm, adjust = 'none')
pairs.b <- pairs(pulp.emm, adjust = 'bonferroni')
pairs.t <- pairs(pulp.emm)
data.frame(transform(pairs.b)[, 1:5], Bonf.p.value = transform(pairs.b)[, 6], Tukey.p.value = tra
```

```
##                contrast estimate     SE df t.ratio Bonf.p.value Tukey.p.value
## 1 operator1 - operator2     0.18 0.2062 16  0.8731      1.00000       0.81854
## 2 operator1 - operator3    -0.38 0.2062 16 -1.8433      0.50336       0.29030
## 3 operator1 - operator4    -0.44 0.2062 16 -2.1343      0.29182       0.18448
## 4 operator2 - operator3    -0.56 0.2062 16 -2.7164      0.09150       0.06579
## 5 operator2 - operator4    -0.62 0.2062 16 -3.0074      0.05009       0.03767
## 6 operator3 - operator4    -0.06 0.2062 16 -0.2910      1.00000       0.99108
##   unadjust.p.value
## 1         0.395509
## 2         0.083893
## 3         0.048637
## 4         0.015251
## 5         0.008349
## 6         0.774758
```

Although the decision on which hypotheses to reject (comparson 2 - 4) is the same here for both methods, the p-values from the Bonferroni method are all

[8] Given $t$ independent samples of size $n$ from the same normal distribution, the studentised range distribution is the distribution of $\frac{R}{S/\sqrt{n}}$, where $R = \bar{y}_{max} - \bar{y}_{min}$ is the difference between the largest and smallest sample means, and $S = \sqrt{\frac{1}{tn-1}\sum_{i=1}^{t}\sum_{j=1}^{n}(y_{ij}-\bar{y})^2}$ is the pooled sample standard deviation.

larger, reflecting its more conservative nature.

## 2.8 Impact of design choices on estimation

Recall from Section 2.5 that the width of a confidence interval for a contrast $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{\tau}$ is given by $2t_{n-t,1-\frac{\alpha}{2}}s\sqrt{\sum_{i=1}^{t}c_i^2/n_i}$. The expectation of the square of this quantity is given by

$$4t^2_{n-t,1-\frac{\alpha}{2}}\sigma^2\sum_{i=1}^{t}c_i^2/n_i\,,$$

as $E(s^2)=\sigma^2$. It is intuitive that a good design should have small values of the square root of this quantity (divided by $2\sigma$),

$$t_{n-t,1-\frac{\alpha}{2}}\sqrt{\sum_{i=1}^{t}c_i^2/n_i}\,,$$

which can be achieved either by increasing $n$, and hence reducing the size of the $t$-quantile, or for choice of the $n_i$ for a fixed $n$, i.e. through choice of replication of each treatment.

### 2.8.1 Optimal treatment allocation

It is quite common that although the total number, $n$, of runs in the experiment may be fixed, the number $n_1, n_2, \ldots, n_t$ applied to the different treatments is under the experimenter's control. Choosing $n_1, n_2$ subject to $n_1+n_2=n$ was the first **optimal design** problem we encountered in Chapter 1.

Assume interest lies in estimating the set of $p$ contrasts $\boldsymbol{c}_1^{\mathrm{T}}\boldsymbol{\tau},\ldots,\boldsymbol{c}_p^{\mathrm{T}}\boldsymbol{\tau}$, with $\boldsymbol{c}_l^{\mathrm{T}}=(c_{l1},\ldots,c_{lt})$. One useful measure of the overall quality of the estimators of these $p$ contrasts is the average variance, given by

$$\frac{\sigma^2}{p}\sum_{l=1}^{p}\sum_{i=1}^{t}c_{li}^2/n_i\,.$$

So we will minimise this variance by allocating larger values of $n_i$ to the treatments with correspondingly larger values of the contrast coefficients $c_{li}$. Therefore an approach to optimal allocation is to choose $\boldsymbol{n}=(n_1,\ldots,n_t)^{\mathrm{T}}$ so as to

$$\text{minimise}\quad \phi(\boldsymbol{n})=\sum_{l=1}^{p}\sum_{i=1}^{t}c_{li}^2/n_i \qquad \text{subject to}\quad \sum_{i=1}^{t}n_i=n\,. \tag{2.11}$$

This is a discrete optimisation problem (the $n_i$ are integers). It is usually easier to solve the relaxed problem, where we allow continuous $0\le n_i\le n$, and round

the resulting solution to obtain integers. There is no guarantee that such a rounded allocation will actually be the optimal integer-valued solution, but it is usually fairly close.

To solve the continuous version of (2.11) we will use the method of Lagrange mutliplers, where we define the function

$$h(\boldsymbol{n}, \lambda) = \phi(\boldsymbol{n}) + \lambda \left( \sum_{i=1}^{t} n_i - n \right) ,$$

introducing the new scalar variable $\lambda$, and solve the set of $t + 1$ equations:

$$\begin{aligned} \frac{\partial h}{\partial n_1} &= 0 \\ &\vdots \\ \frac{\partial h}{\partial n_t} &= 0 \\ \frac{\partial h}{\partial \lambda} &= 0 \,. \end{aligned}$$

In this case, we have

$$\frac{\partial h}{\partial n_i} = -\sum_{l=1}^{p} c_{li}^2 / n_i^2 + \lambda = 0 \,, \quad i = 1, \ldots t, \tag{2.12}$$

and

$$\frac{\partial h}{\partial \lambda} = \sum_{i=1}^{t} n_i - n = 0 \,.$$

This last equation ensures $\sum_{i=1}^{t} n_i = n$. From the $t$ equations described by (2.12), we get

$$n_i \propto \sqrt{\sum_{l=1}^{p} c_{li}^2}$$

We don't need to explicitly solve for $\lambda$ to find the normalising constant for each $n_i$. As we know $\sum_{i=1}^{t} n_i = n$, we obtain,

$$n_i = \frac{\sqrt{\sum_{l=1}^{p} c_{li}^2}}{\sum_{i=1}^{t} \sqrt{\sum_{l=1}^{p} c_{li}^2}} n \,. \tag{2.13}$$

Let's return to Example 2.1 and calculate the optimal allocations under two different sets of contrasts. First, we define an `R` function for calculating (2.13).

```
opt_ni <- function(C, n) {
  CtC <- t(C) %*% C
  n * sqrt(diag(CtC)) / sum(sqrt(diag(CtC)))
}
```

Checking that the function `opt-ni` matches (2.13) is left as an exercise.

Consider two sets of contrasts:

1. All pairwise comparisons between the four treatments

$$
\begin{aligned}
c_1 &= (-1, 1, 0, 0) \\
c_2 &= (-1, 0, 1, 0) \\
c_3 &= (-1, 0, 0, 1) \\
c_4 &= (0, -1, 1, 0) \\
c_5 &= (0, -1, 0, 1) \\
c_6 &= (0, 0, -1, 1)\,.
\end{aligned}
$$

   Calculating (2.13), we obtain

```
C <- matrix(
  c(
-1, 1, 0, 0,
-1, 0, 1, 0,
-1, 0, 0, 1,
0, -1, 1, 0,
0, -1, 0, 1,
0, 0, -1, 1),
  nrow = 6, byrow = T
)
opt_ni(C, 20)
```

```
## [1] 5 5 5 5
```

   Hence confirming that equal replication of the treatments is optimal for minimising the average variance of estimators of the pairwise treatment differences.

2. If operator 4 is new to the mill, it may be desired to test their output to the average output from the other three operators, using a contrast with coefficients $c = (1/3, 1/3, 1/3, -1)$. The allocation to minimise the variance of the corresponding estimator is given by:

```
C <- matrix(
  c(1/3, 1/3, 1/3, -1),
  nrow = 1
)
opt_ni(C, 20)
```

```
## [1]  3.333  3.333  3.333 10.000
```

So the optimal allocation splits 10 units between operators 1-3, and allocates 10 units to operator 4. There is no exact integer rounding possible, so we will use $n_1 = 4$, $n_2 = n_3 = 3$, $n_4 = 10$ and calculate the efficiency by comparing the variance of this allocation to that from the equally allocated design.

```
crd_var <- function(C, n) {
  CtC <- t(C) %*% C
  sum(diag(CtC) / n)
}
n_equal <- rep(5, 4)
n_opt <- c(4, 3, 3, 10)
crd_var(C, n_opt) / crd_var(C, n_equal)
```

```
## [1] 0.7569
```

So the efficiency of the equally allocated design for estimating this contrast is 75.69 %.

### 2.8.2 Overall size of the experiment

We can also consider the complementary question: suppose the proportion of runs that is to be allocated to each treatment has been fixed in advance, what size of experiment should be performed to meet the objectives? That is, given a fixed proportion, $w_i$, of resource to be allocated to the $i$th treatment, so that $n_i = nw_i$ units will be allocated to that treatment, what value of $n$ should be chosen?

One way of thinking about this question is to consider the ratio

$$\begin{aligned}\frac{|\boldsymbol{c}^{\mathrm{T}}\boldsymbol{\tau}|}{\sqrt{\mathrm{Var}(\widehat{\boldsymbol{c}^{\mathrm{T}}\boldsymbol{\tau}})}} &= \frac{|\boldsymbol{c}^{\mathrm{T}}\boldsymbol{\tau}|}{\sqrt{\frac{\sigma^2}{n}\sum_{i=1}^{t} c_i^2/w_i}} \\ &= \sqrt{n}\frac{|\boldsymbol{c}^{\mathrm{T}}\boldsymbol{\tau}|/\sigma}{\sqrt{\sum_{i=1}^{t} c_i^2/w_i}},\end{aligned}$$

which is analogous to the test statistic for $H_0 : \boldsymbol{c}^{\mathrm{T}}\boldsymbol{\tau} = 0$. For a given value of the signal-to-noise ratio $d = |\boldsymbol{c}^{\mathrm{T}}\boldsymbol{\tau}|/\sigma$, we can choose $n$ to result in a specified value of $T = |\boldsymbol{c}^{\mathrm{T}}\boldsymbol{\tau}|/\sqrt{\mathrm{Var}(\widehat{\boldsymbol{c}^{\mathrm{T}}\boldsymbol{\tau}})}$:

$$n = T^2\frac{\sum_{i=1}^{t} c_i^2/w_i}{d^2}.$$

Returning to Example 2.1, assume are testing a single pairwise comparison and that we require $T = 3$, so that the null hypothesis would be comfortably rejected at the 5% level (cf 1.96 for a standard z-test). For equal allocation of the units to each treatment ($w_1 = \cdots = w_4 = 1/4$) and a variety of different values of the signal-to-noise ratio $d$, we obtained the following optimal experiment sizes:

```
opt_n <- function(cv, prop, snr, target) target ^ 2 * c(t(cv) %*% diag( 1 / prop) %*% c
cv <- c(-1, 1, 0, 0)
w <- rep(1/4, 4)
snr <- c(0.5, 1, 1.5, 2, 2.5, 3)
cbind('Signal-to-noise' = snr, 'n' = opt_n(cv, w, snr, 3))
```

```
##      Signal-to-noise      n
## [1,]             0.5 288.00
## [2,]             1.0  72.00
## [3,]             1.5  32.00
## [4,]             2.0  18.00
## [5,]             2.5  11.52
## [6,]             3.0   8.00
```

So, for example, to achieve $T = 3$ with a signal-to-noise ratio of $d = 0.5$ requires $n = 288$ runs. As would be expected, the number of runs required to achieve this value of $T$ decreases as the signal-to-noise ration increases. For $d = 3$, only a very small experiment with $n = 8$ runs is needed.

## 2.9 Exercises

1. a. For Example 2.1, calculate the mean response for each operator and show that the treatment differences and results from hypothesis tests using the results in Section 2.5 are the same as those found in Section 2.6 using `pairwise.t.test`, and `emmeans`.

   b. Also check the results in Section 2.7 by (i) adjusting individual p-values (for Bonferroni) and (ii) using the `qtukey` command.

Solution

As a reminder, the data from the experiment is as follows.

| Operator 1 | Operator 2 | Operator 3 | Operator 4 |
|---|---|---|---|
| 59.8 | 59.8 | 60.7 | 61.0 |
| 60.0 | 60.2 | 60.7 | 60.8 |
| 60.8 | 60.4 | 60.5 | 60.6 |
| 60.8 | 59.9 | 60.9 | 60.5 |
| 59.8 | 60.0 | 60.3 | 60.5 |

The mean response, and variance, from each treatment is given by

| operator | n_i | mean | variance |
|---|---|---|---|
| 1 | 5 | 60.24 | 0.268 |
| 2 | 5 | 60.06 | 0.058 |
| 3 | 5 | 60.62 | 0.052 |
| 4 | 5 | 60.68 | 0.047 |

The sample variance, $s^2 = 0.106$, from (2.10). As $\sum_{i=1}^{t} c_i^2/n_i = \frac{2}{5}$ for contrast vectors $\boldsymbol{c}$ corresponding to pairwise differences, the standard error of each pairwise difference is given by $\sqrt{\frac{2s^2}{5}} = 0.206$. Hence, we can create a table of pairwise differences, standard errors and test statistics.

| contrast | estimate | SE | df | t.ratio | unadjust.p.value | Bonferroni | Tukey |
|---|---|---|---|---|---|---|---|
| operator1 - operator2 | 0.18 | 0.206 | 16 | 0.873 | 0.396 | 1.000 | 0.819 |
| operator1 - operator3 | -0.38 | 0.206 | 16 | -1.843 | 0.084 | 0.503 | 0.290 |
| operator1 - operator4 | -0.44 | 0.206 | 16 | -2.134 | 0.049 | 0.292 | 0.184 |
| operator2 - operator3 | -0.56 | 0.206 | 16 | -2.716 | 0.015 | 0.092 | 0.066 |
| operator2 - operator4 | -0.62 | 0.206 | 16 | -3.007 | 0.008 | 0.050 | 0.038 |
| operator3 - operator4 | -0.06 | 0.206 | 16 | -0.291 | 0.775 | 1.000 | 0.991 |

Unadjusted p-values are obtained from the t-distribution, as twice the tail probabilities (`2 * (1 - pt(abs(t.ratio), 16))`). For Bonferroni, we simply multiply these p-values by $\binom{t}{2} = 6$, and then take the minimum of this value and 1. For the Tukey method, we use `1 - ptukey(abs(t.ratio) * sqrt(2), 4, 16)` (see `?ptukey`).

Alternatively, to test each hypothesis at the 5% level, we can compare each t.ratio to (i) `qt(0.975, 16) = 2.12` (unadjusted); (ii) `qt(1 - 0.025/6, 16) = 3.008` (Bonferroni); or (iii) `qtukey(0.95, 4, 16) / sqrt(2) = 2.861`.

2. (Adapted from Wu and Hamada, 2009) The bioactivity of four different drugs $A$, $B$, $C$ and $D$ for treating a particular illness was compared in a study and the following ANOVA table was given for the data:

| Source | Degrees of freedom | Sums of squares | Mean square |
|---|---|---|---|
| Treatment | 3 | 64.42 | 21.47 |
| Residual | 26 | 62.12 | 2.39 |
| Total | 29 | 126.54 | |

   i. What considerations should be made when assigning drugs to patients, and why?

   ii. Use an $F$-test to test at the 0.01 level the null hypothesis that the four drugs have the same bioactivity.

   iii. The average response from each treatment is as follows: $\bar{y}_{A\cdot} = 66.10$ ($n_A = 7$ patients), $\bar{y}_{B\cdot} = 65.75$ ($n_B = 8$), $\bar{y}_{C\cdot} = 62.63$ ($n_C = 9$),

and $\bar{y}_{D\cdot} = 63.85$ ($n_D = 6$). Conduct hypothesis tests for all pairwise comparisons using the Bonferroni and Tukey methods for an experiment-wise error rate of 0.05.

iv. In fact, $A$ and $B$ are brand-name drugs and $C$ and $D$ are generic drugs. Test the null hypothesis at the 5% level that brand-name and generic drugs have the same bioactivity.

Solution

i. Each patient should be randomly allocated to one of the drugs. This is to protect against possible bias from lurking variables, e.g. demographic variables or subjective bias from the study administrator (blinding the study can also help to protect against this).

ii. Test statistic = (Treatment mean square)/(Residual mean square) = 21.47/2.39 = 8.98. Under $H_0$: no difference in bioactivity between the drugs, the test statistic follows an $F_{3,26}$ distribution, which has a 1% critical value of `qf(0.99, 3, 26) = 4.6366`. Hence, we can reject $H_0$.

iii. For each difference, the test statistic has the form

$$\frac{|\bar{y}_{i\cdot} - \bar{y}_{j\cdot}|}{s\sqrt{\frac{1}{n_i} + \frac{1}{n_j}}},$$

for $i, j = A, B, C, D$; $i \neq j$. The treatment means and repetitions are given in the question (note that not all $n_i$ are equal). From the ANOVA table, we get $s^2 = 62.12/26 = 2.389$. The following table summarises the differences between drugs:

| | $A-B$ | $A-C$ | $A-D$ | $B-C$ | $B-D$ | $C-D$ |
|---|---|---|---|---|---|---|
| Abs. difference | 0.35 | 3.47 | 2.25 | 3.12 | 1.9 | 1.22 |
| Test statistic | 0.44 | 4.45 | 2.62 | 4.15 | 2.28 | 1.50 |

The Bonferroni critical value is $t_{26,1-0.05/12} = 3.5069$. The Tukey critical value is $q_{4,26,0.95}/\sqrt{2} = 2.7433$ (available `R` as `qtukey(0.95, 4, 26) / sqrt(2)`). Hence under both methods, bioactivity of drugs $A$ and $C$, and $B$ and $C$, are significantly different.

iv. A suitable contrast has $\boldsymbol{c} = (0.5, 0.5, -0.5, -0.5)$, with $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{\tau} = (\tau_A + \tau_B)/2 - (\tau_C + \tau_D)/2$ (the difference in average treatment effects).

An estimate for this contrast is given by $(\bar{y}_{A\cdot} + \bar{y}_{B\cdot})/2 - (\bar{y}_{C\cdot} + \bar{y}_{D\cdot})/2$, with variance

$$\mathrm{Var}\left(\frac{1}{2}(\bar{y}_{A\cdot} + \bar{y}_{B\cdot}) - \frac{1}{2}(\bar{y}_{C\cdot} + \bar{Y}_{D\cdot})\right) = \frac{\sigma^2}{4}\left(\frac{1}{n_A} + \frac{1}{n_B} + \frac{1}{n_C} + \frac{1}{n_D}\right).$$

Table 2.5: Naphthalene black experiment: yields (grams of standard colour) from six different batches of hydrochloric acid.

| Batch 1 | Batch 2 | Batch 3 | Batch 4 | Batch 5 | Batch 6 |
|---|---|---|---|---|---|
| 145 | 140 | 195 | 45 | 195 | 120 |
| 40 | 155 | 150 | 40 | 230 | 55 |
| 40 | 90 | 205 | 195 | 115 | 50 |
| 120 | 160 | 110 | 65 | 235 | 80 |
| 180 | 95 | 160 | 145 | 225 | 45 |

Hence, a test statistic for $H_0 : \frac{1}{2}(\tau_A + \tau_B) - \frac{1}{2}(\tau_C + \tau_D) = 0$ is given by

$$\frac{\frac{1}{2}(\bar{y}_{A\cdot} + \bar{y}_{B\cdot}) - \frac{1}{2}(\bar{y}_{C\cdot} + \bar{y}_{D\cdot})}{\sqrt{\frac{s^2}{4}\left(\frac{1}{n_A} + \frac{1}{n_B} + \frac{1}{n_C} + \frac{1}{n_D}\right)}} = \frac{2.685}{\frac{\sqrt{2.389}}{2}\sqrt{\frac{1}{7} + \frac{1}{8} + \frac{1}{9} + \frac{1}{6}}} = 4.70\,.$$

The critical value is $t_{26,1-0.05/2} = 2.0555$. Hence, we can reject $H_0$ and conclude there is a difference between brand-name and generic drugs.

3. The below table gives data from a completely randomised design to compare six different batches of hydrochloric acid on the yield of a dye (naphthalene black 12B).

```
napblack <- data.frame(batch = rep(factor(1:6), rep(5, 6)),
                       repetition = rep(1:5, 6),
                       yield = c(145, 40, 40, 120, 180, 140, 155, 90, 160, 95,
                                 195, 150, 205, 110, 160, 45, 40, 195, 65, 145,
                                 195, 230, 115, 235, 225, 120, 55, 50, 80, 45)
                         )
knitr::kable(
tidyr::pivot_wider(napblack, names_from = batch, values_from = yield)[, -1],
 col.names = paste("Batch", 1:6),
 caption = "Naphthalene black experiment: yields (grams of standard colour) from six differe
)
```

Conduct a full analysis of this experiment, including

   a. exploratory data analysis;
   b. fitting a linear model, and conducting an F-test to compare to a model that explains variation using the six batches to the null model;
   c. usual linear model diagnostics;
   d. multiple comparisons of all pairwise differences between treatments.

Solution

a. Two of the simplest ways of examining the data are to calculate basic descriptive statistics, e.g. the mean and standard deviation of the yield in

each batch, and to plot the data in the different batches using a simple graphical display, e.g. a stripchart of the yields in each batch. Notice that in both `aggregate` and `stripchart` we use the formula `yield ~ batch`. This formula splits the data into groups defined by batch.

```
aggregate(yield ~ batch, data = napblack, FUN = function(x) c(mean = mean(x),
                                                              st.dev = sd(x)))
```

```
##   batch yield.mean yield.st.dev
## 1     1     105.00        63.05
## 2     2     128.00        33.28
## 3     3     164.00        37.98
## 4     4      98.00        68.70
## 5     5     200.00        50.00
## 6     6      70.00        31.02
```

```
boxplot(yield ~ batch, data = napblack)
```

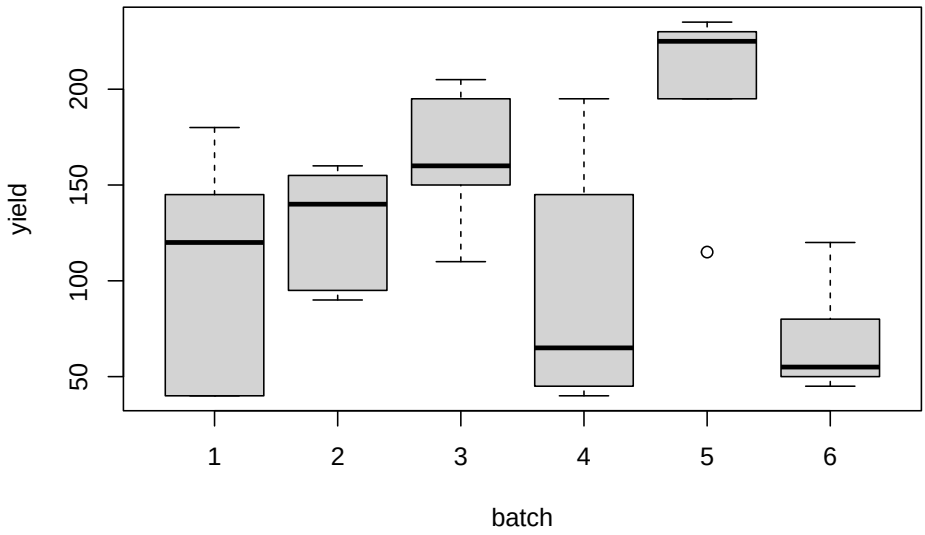



Figure 2.2: Naphthalene black experiment: distributions of dye yields from the six batches.

Notice that even within any particular batch, the number of grams of standard dyestuff colour determined by the dye trial varies from observation to observation. This *within-group* variation is considered to be random or residual variation. This cannot be explained by any differences between batches. However, a second source of variation in the overall data set can be explained by variation between the batches, i.e. between the different batch means themselves. We can see from the box plots (Figure 2.2) and the mean yields in each batch that observations from batch number five appear to have given higher yields (in grams of standard colour) than those

from the other batches.

b. When we fit linear models and compare them using analysis of variance (ANOVA), it enables us to decide whether the differences that seem to be evident in these simple plots and descriptive statistics are statistically significant or whether this kind of variation could have arisen by chance, even though there are no real differences between the batches.

An ANOVA table may be used to compare a linear model including differences between the batches to the null model. The linear model we will fit is a simple unit-treatment model:

$$Y_{ij} = \mu + \tau_i + \varepsilon_{ij}\,, \qquad i = 1, \ldots, 6;\ j = 1, \ldots, 5\,, \tag{2.14}$$

where $Y_{ij}$ is the response obtained from the $j$th repetition of the $i$th batch, $\mu$ is a constant term, $\tau_i$ is the expected effect due to the observation being in the $i$th batch ($i = 1, \ldots, 5$) and $\varepsilon_{ij}$ are the random errors.

A test of the hypothesis that the group means are all equal is equivalent to a test that the $\tau_i$ are all equal to 0 ($H_0 : \tau_1 = \tau_2 = \cdots = \tau_6 = 0$). We can use `lm` to fit model (2.14), and `anova` to test the hypothesis. Before we fit the linear model, we need to make sure `batch` has type `factor`[9].

```
napblack$batch <- as.factor(napblack$batch)
napblack.lm <- lm(yield ~ batch, data = napblack)
anova(napblack.lm)
```

```
## Analysis of Variance Table
##
## Response: yield
##           Df Sum Sq Mean Sq F value Pr(>F)
## batch      5  56358   11272     4.6 0.0044 **
## Residuals 24  58830    2451
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value of 0.0044 indicates significant differences between at least two of the batch means. Therefore $H_0$ is rejected and a suitable multiple comparison test should be carried out.

c. To perform our analysis, we have fitted a linear model. Therefore, we should use some plots of the residuals $y_{ij} - \hat{y}_{ij}$ to check the model assumptions, particularly that the errors are independently and identically normally distributed. The function `rstandard` which produces residuals which have been standardised to have variance equal to 1.

[9] Factors are variables in `R` which take on a limited number of different values (e.g. categorical variables). We need to define a categorical variable, like `batch` as a `factor` to ensure they are treated correctly by functions such as `lm`.

```
standres <- rstandard(napblack.lm)
fitted <- fitted(napblack.lm)
par(mfrow = c(1, 2), pty = "s")
with(napblack, {
  plot(batch, standres, xlab = "Batch", ylab = "Standarised residuals")
  plot(fitted, standres, xlab = "Fitted value", ylab = "Standarised residuals")
})
```

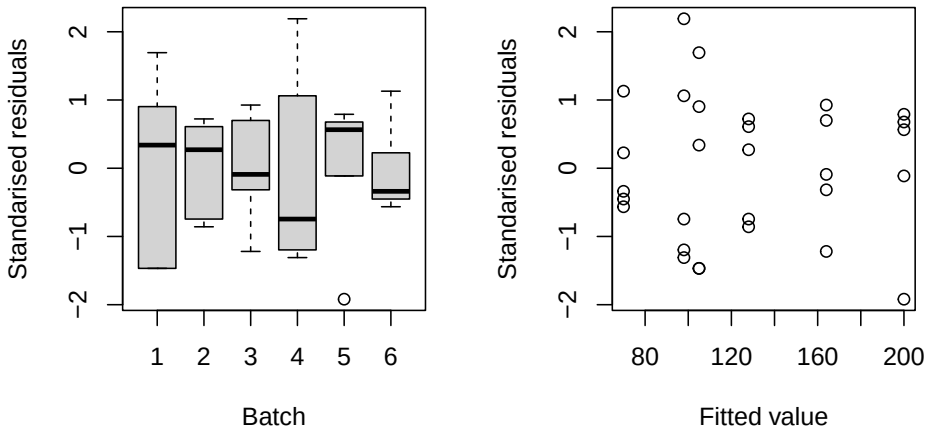



Figure 2.3: Residuals against batch (left) and fitted values (right) for the linear model fit to the naphthalene black data.

The plots (Figure 2.3) show no large standardised residuals ($> 2$ in absolute value[10]). While there is some evidence of unequal variation across batches, there is no obvious pattern with respect to fitted values (e.g. no "funnelling").

We can also plot the standardised residuals against the quantiles of a standard normal distribution to assess the assumption of normality.

```
par(pty = "s")
qqnorm(standres, main = "")
```

The points lie quite well on a straight line (see Figure 2.4), suggesting the assumption of normality is valid. Overall, the residual plots look reasonable; some investigation of transformations to correct for non-constant variance could be investigated (see MATH2010/STAT6123).

d. When a significant difference between the treatments has been indicated, the next stage is to try to determine which treatments differ. In some cases a specific difference is of interest, a control versus a new treatment for

[10]We would anticipate 95% of the standardised residuals to lie in [-1.96, 1.96], as they will follow a standard normal distribution if the model assumptions are correct.

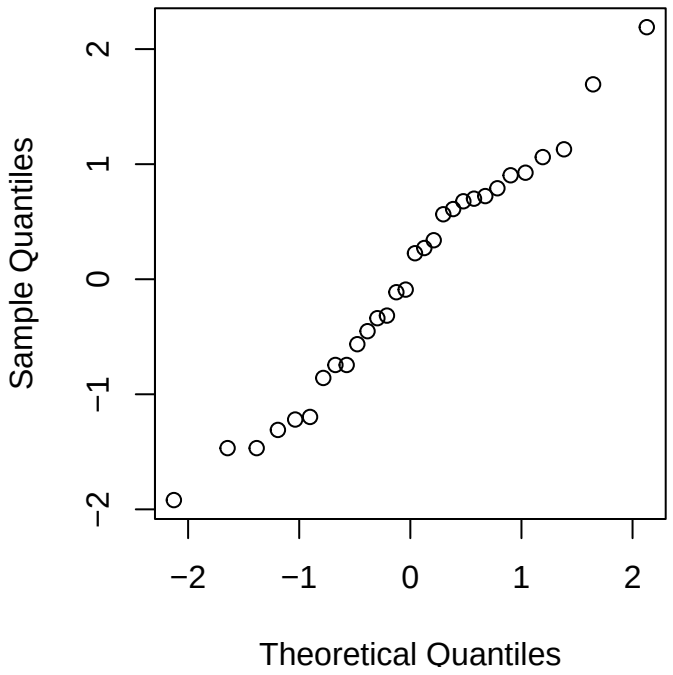



Figure 2.4: Normal probability plot for the standardised residuals for the linear model fit to the naphthalene black data.

instance, in which case that difference could now be inspected. However, usually no specific differences are to be considered a priori, and *any* difference is of practical importance. A multiple comparison procedure is required to investigate all possible differences, which takes account of the number of possible differences available amongst the treatments (15 differences between the six batches here).

We will use Tukey's method for controlling the experiment-wise type I error rate, fixed here at 5%, as implemented by `emmeans`.

```
napblack.emm <- emmeans::emmeans(napblack.lm, 'batch')
pairs(napblack.emm)
```

```
##  contrast        estimate   SE df t.ratio p.value
##  batch1 - batch2      -23 31.3 24  -0.735  0.9755
##  batch1 - batch3      -59 31.3 24  -1.884  0.4351
##  batch1 - batch4        7 31.3 24   0.224  0.9999
##  batch1 - batch5      -95 31.3 24  -3.034  0.0566
##  batch1 - batch6       35 31.3 24   1.118  0.8692
##  batch2 - batch3      -36 31.3 24  -1.150  0.8555
##  batch2 - batch4       30 31.3 24   0.958  0.9266
##  batch2 - batch5      -72 31.3 24  -2.299  0.2329
##  batch2 - batch6       58 31.3 24   1.852  0.4535
##  batch3 - batch4       66 31.3 24   2.108  0.3167
##  batch3 - batch5      -36 31.3 24  -1.150  0.8555
##  batch3 - batch6       94 31.3 24   3.002  0.0606
##  batch4 - batch5     -102 31.3 24  -3.257  0.0348
##  batch4 - batch6       28 31.3 24   0.894  0.9442
##  batch5 - batch6      130 31.3 24   4.152  0.0043
##
## P value adjustment: tukey method for comparing a family of 6 estimates
```

We have two significant differences, between batches 4-5 and 5-6.

```
subset(transform(pairs(napblack.emm)), p.value < 0.05)
```

```
##           contrast estimate    SE df t.ratio  p.value
## 13 batch4 - batch5     -102 31.31 24  -3.257 0.034820
## 15 batch5 - batch6      130 31.31 24   4.152 0.004295
```

4. (Adapted from Morris, 2011) Consider a completely randomised design with $t = 5$ treatments and $n = 50$ units. The contrasts

$$\tau_2 - \tau_1, \quad \tau_3 - \tau_2, \quad \tau_4 - \tau_3, \tau_5 - \tau_4$$

are of primary interest to the experimenter.

   a. Find an allocation of the 50 units to the 5 treatments, i.e. find $n_1, \ldots, n_5$,

that minimises the average variance of the corresponding contrast estimators.

b. Fixing the proportions of experimental effort applied to each treatment to those found in part (a), i.e. to $w_i = n_i/50$, find the value of $n$ required to make the ratio $T = |\boldsymbol{c}^{\mathrm{T}}\boldsymbol{\tau}|/\sqrt{\mathrm{var}\left(\widehat{\boldsymbol{c}^{\mathrm{T}}\boldsymbol{\tau}}\right)} = 2$ assuming a signal-to-noise ratio of 1.

Solution

a. We can use the function `opt_ni` given in Section 2.8.1:

```
n <- 50
C <- matrix(
  c(
  -1, 1, 0, 0, 0,
  0, -1, 1, 0, 0,
  0, 0, -1, 1, 0,
  0, 0, 0, -1, 1
  ), nrow = 4, byrow = T
)
opt_ni(C, n)
```

```
## [1]  8.009 11.327 11.327 11.327  8.009
```

Rounding, we obtain a solution of the form $n_1 = n_5 = 8$, $n_2 = n_4 = 11$ and $n_3 = 12$. Any of $n_2, n_3, n_4$ may be rounded up to 12 to form a design with the same variance.

```
nv <- c(8, 11, 11, 11, 8)
crd_var(C, nv + c(0, 1, 0, 0, 0))
crd_var(C, nv + c(0, 0, 1, 0, 0))
crd_var(C, nv + c(0, 0, 0, 1, 0))
```

```
## [1] 0.7803
## [1] 0.7803
## [1] 0.7803
```

b. The optimal ratios for each treatment from part (a) are $w_1 = w_5 = 0.1602$ and $w_2 = w_3 = w_4 = 0.2265$. Fixing these, we can use code from Section 2.8.2 to find the required value of $n$ for each contrast.

```
nv <- NULL
for(i in 1:4) nv[i] <- opt_n(C[i, ], opt_ni(C, n) / n, 1, 2) # snr = 1, target = 2
nv
```

```
## [1] 42.63 35.31 35.31 42.63
```

Hence, we need $n = 43$ for to achieve $T = 2$ for the first and last contrasts, and $n = 36$ for the second and third. The differences are due to the

different proportions $w_i$ assumed for each treatment. To achieve $T = 2$ for all contrasts, we pick the larger number, $n = 43$.

# Chapter 3

# Blocking

The completely randomised design (CRD) works well when there is sufficient homogeneous experimental units to perform the whole experiment under the same, or very similar, conditions and there are no restrictions on the randomisation of treatments to units. The only systematic (non-random) differences in the observed responses result from differences between the treatments. While such designs are commonly and successfully used, especially in smaller experiments, their application can often be unrealistic or impractical in many settings.

A common way in which the CRD fails is a lack of sufficiently similar experimental units. If there are systemtic differences between different batches, or **blocks** of units, these differences should be taken into account in both the allocation of treatments to units and the modelling of the resultant data. Otherwise, block-to-block differences may bias treatment comparisons and/or inflate our estimate of the background variability and hence reduce our ability to detect important treatment effects.

**Example 3.1.** Steel bar experiment (Morris, 2011, ch. 4)

Kocaoz et al. (2005) described an experiment to assess the strength of steel reinforcement bars from $t = 4$ coatings[1] (treatments). In total $n = 32$ different bars (units) were available, but the testing process meant sets of four bars were tested together. To account for potential test-specific features (e.g. environmental or operational), these different test sets were assumed to form $b = 8$ blocks of size $k = 4$. The data are shown in Table 3.1 below.

```
bar <- data.frame(coating = rep(factor(1:4), 8),
                  block = rep(factor(1:8), rep(4, 8)),
                  strength = c(136, 147, 138, 149, 136, 143, 122, 153, 150, 142, 131, 136,
```

[1]The four coatings were all made from Engineering Thermoplastic Polyurethane (ETPU); coating one was solely made from ETPU, coatings 2-4 had additional glass fibre, carbon fibre or aramid fibre added, respectively.

Table 3.1: Steel bar experiment: tensile strength values (kliograms per square inch, ksi) from steel bars with four different coatings.

| Block | Coating 1 | Coating 2 | Coating 3 | Coating 4 |
|---|---|---|---|---|
| 1 | 136 | 147 | 138 | 149 |
| 2 | 136 | 143 | 122 | 153 |
| 3 | 150 | 142 | 131 | 136 |
| 4 | 155 | 148 | 130 | 129 |
| 5 | 145 | 149 | 136 | 139 |
| 6 | 150 | 149 | 147 | 144 |
| 7 | 147 | 150 | 125 | 140 |
| 8 | 148 | 149 | 118 | 145 |

```
                                    155, 148, 130, 129, 145, 149, 136, 139, 150, 149, 14
                                    147, 150, 125, 140, 148, 149, 118, 145)
                    )
knitr::kable(
 tidyr::pivot_wider(bar, names_from = coating, values_from = strength),
 col.names = c("Block", paste("Coating", 1:4)),
 caption = "Steel bar experiment: tensile strength values (kliograms per square inch, k
)
```

Here, each block has size 4, which is equal to the number of treatments in the experiment, and each treatment is applied in each block. This is an example of a **randomised complete block design**.

We can study the data graphically, plotting by treatment and by block.

```
boxplot(strength ~ block, data = bar)
boxplot(strength ~ coating, data = bar)
```

The box plots within each plot in Figure 3.1 are comparable, as every treatment has occured with every block the same number of times (once). For example, when we compare the box plots for treatments 1 and 3, we know each of then display one observation from each block. Therefore, differences between treatments will not be influenced by large differences between blocks. This **balance** makes our analysis more straighforward. By eye, it appears here there may be differences between coating 3 and the other three coatings.

**Example 3.2.** Tyre experiment (Wu and Hamada, 2009, ch. 3)

Davies (1954), p.200, examined the effect of $t = 4$ different rubber compounds (treatments) on the lifetime of a tyre. Each tyre is only large enough to split into $k = 3$ segments whilst still containing a representative amount of each compound. When tested, each tyre is subjected to the same road conditions, and hence is treated as a block. A design with $b = 4$ blocks was used, as displayed in Table

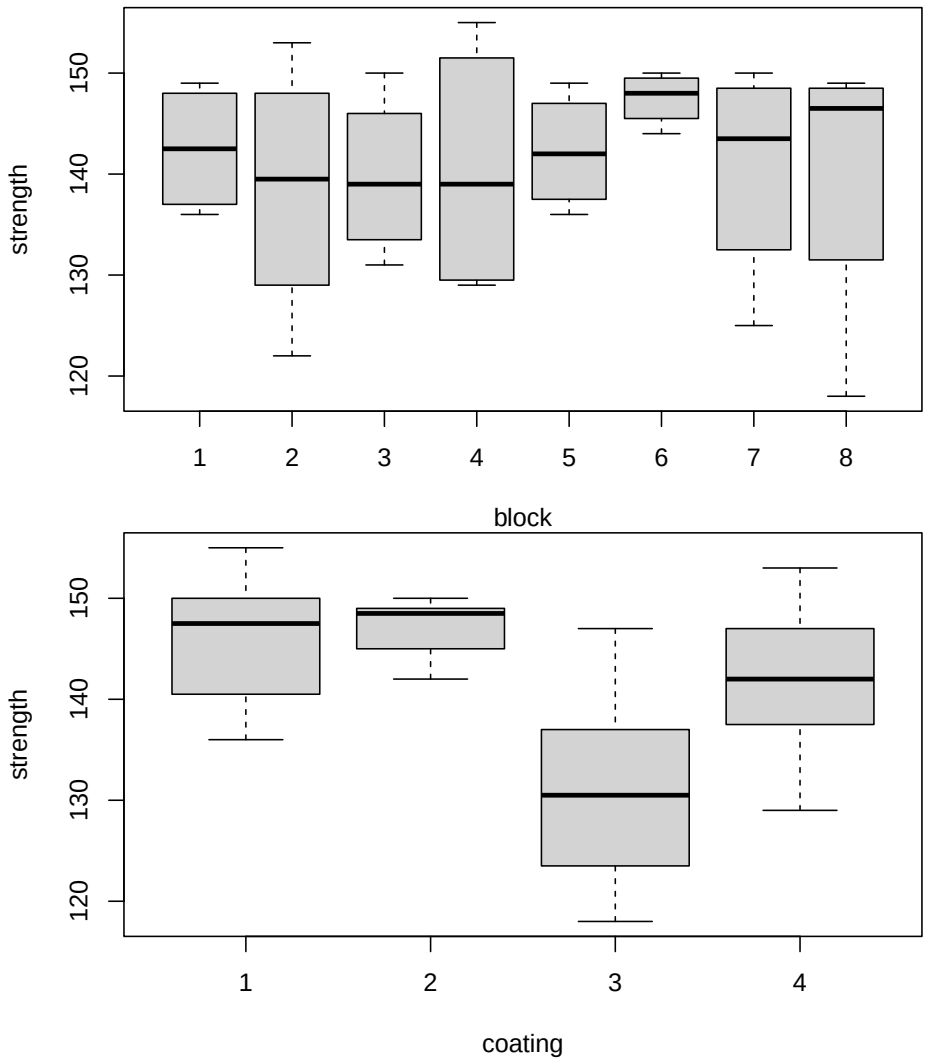



Figure 3.1: Steel bar experiment: distributions of tensile strength (ksi) from the eight blocks (top) and the four coatings (bottom).

Table 3.2: Tyre experiment: relative wear measurements (unitless) from tires made with four different rubber compounds.

| Block | Compound 1 | Compound 2 | Compound 3 | Compound 4 |
|---|---|---|---|---|
| 1 | 238 | 238 | 279 | |
| 2 | 196 | 213 | | 308 |
| 3 | 254 | | 334 | 367 |
| 4 | | 312 | 421 | 412 |

3.2.

```
tyre <- data.frame(compound = as.factor(c(1, 2, 3, 1, 2, 4, 1, 3, 4, 2, 3, 4)),
                   block = rep(factor(1:4), rep(3, 4)),
                   wear = c(238, 238, 279, 196, 213, 308, 254, 334, 367, 312, 421, 412)
                     )
options(knitr.kable.NA = '')
knitr::kable(
 tidyr::pivot_wider(tyre, names_from = compound, values_from = wear),
 col.names = c("Block", paste("Compound", 1:4)),
 caption = "Tyre experiment: relative wear measurements (unitless) from tires made with
)
```

Here, each block has size $k = 3$, which is smaller than the number of treatments ($t = 4$). Hence, each block cannot contain an application of each treatment. This is an example of an **incomplete block design**.

Graphical exploration of the data is a little more problematic in this example. As each treatment does not occur in each block, box plots such as Figure 3.2 are not as informative. Do compounds three and four have higher average wear because they were the only compounds to both occur in blocks 3 and 4? Or do blocks 3 and 4 have a higher mean because they contain both compounds 3 and 4? The design cannot help us entirely disentangle the impact of blocks and treatments[2]. In our modelling, we will assume variation should first be described by blocks (which are generally fixed aspects of the experiment) and then treatments (which are more directly under the experimenter's control).

```
boxplot(wear ~ block, data = tyre)
boxplot(wear ~ compound, data = tyre)
```

## 3.1 Unit-block-treatment model

If $n_{ij}$ is the number of times treatment $j$ occurs in block $i$, a common statistical model to describe data from a blocked experiment has the form

[2]This is our first example of (partial) confounding, which we will see again in Chapters 5 and 6

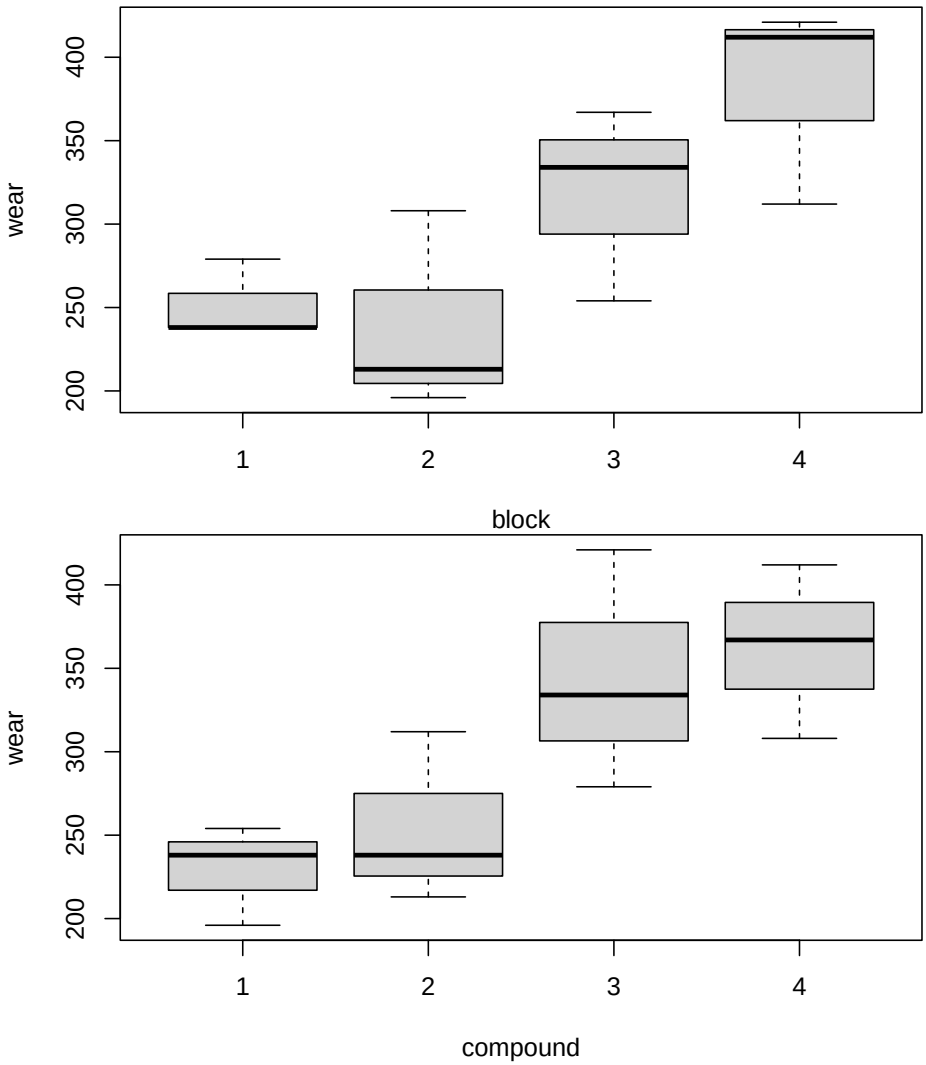



Figure 3.2: Tyre experiment: distributions of wear from the four blocks (top) and the four compounds (bottom).

$$y_{ijl} = \mu + \beta_i + \tau_j + \varepsilon_{ijl}\,, \qquad i = 1, \ldots, b; j = 1, \ldots, t; l = 1, \ldots, n_{ij}\,, \tag{3.1}$$

where $y_{ijl}$ is the response from the $l$th application of the $j$th treatment in the $i$th block, $\mu$ is a constant parameter, $\beta_i$ is the effect of the $i$th block, $\tau_j$ is the effect of treatment $j$, and $\varepsilon_{ijl} \sim N(0, \sigma^2)$ are once again random individual effects from each experimental unit, assumed independent. The total number of runs in the experiment is given by $n = \sum_{i=1}^{b} \sum_{j=1}^{t} n_{ij}$.

For Example 3.1, there are $t = 4$ experiments, $b = 8$ blocks and each treatment occurs once in each block, so $n_{ij} = 1$ for all $i, j$. In Example 3.2, there are again $t = 4$ treatments but now only $b = 4$ blocks and not every treatment occurs in every block. In fact, we have $n_{11} = n_{12} = n_{13} = 1$, $n_{14} = 0$, $n_{21} = n_{22} = n_{24} = 1$, $n_{23} = 0$, $n_{31} = n_{33} = n_{34} = 1$, $n_{32} = 0$, $n_{41} = 0$ and $n_{42} = n_{43} = n_{44} = 1$.

Writing model (3.1) is matrix form as a partitioned linear model, we obtain

$$\boldsymbol{y} = \mu \mathbf{1}_n + X_1 \boldsymbol{\beta} + X_2 \boldsymbol{\tau} + \boldsymbol{\varepsilon}\,, \tag{3.2}$$

with $\boldsymbol{y}$ the $n$-vector of responses, $X_1$ and $X_2$ $n \times b$ and $n \times t$ model matrices for blocks and treatments, respectively, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_b)^{\mathrm{T}}$, $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_t)^{\mathrm{T}}$ and $\boldsymbol{\varepsilon}$ the $n$-vector of errors.

In equation(3.2), assuming without loss of generality that runs of the experiment are ordered by block, the matrix $X_1$ has the form

$$X_1 = \bigoplus_{i=1}^{b} \mathbf{1}_{k_i} = \begin{bmatrix} \mathbf{1}_{k_1} & \mathbf{0}_{k_1} & \cdots & \mathbf{0}_{k_1} \\ \mathbf{0}_{k_2} & \mathbf{1}_{k_2} & \cdots & \mathbf{0}_{k_2} \\ \vdots & & \ddots & \vdots \\ \mathbf{0}_{k_b} & \mathbf{0}_{k_b} & \cdots & \mathbf{1}_{k_b} \end{bmatrix}\,,$$

where $k_i = \sum_{j=1}^{t} n_{ij}$, the number of units in the $i$th block. The structure of matrix $X_2$ is harder to describe so succinctly, but each row includes a single non-zero entry, equal to one, indicating which treatment was applied in that run of the experiment. The first $k_1$ rows correspond to block 1, the second $k_2$ to block 2, and so on. We will see special cases later.

## 3.2 Normal equations

Writing as a partitioned model $\boldsymbol{y} = W\boldsymbol{\alpha} + \boldsymbol{\varepsilon}$, with $W = [\mathbf{1}|X_1|X_2]$ and $\boldsymbol{\alpha}^{\mathrm{T}} = [\mu|\boldsymbol{\beta}^{\mathrm{T}}|\boldsymbol{\tau}^{\mathrm{T}}]$, the least squares normal equations

$$W^{\mathrm{T}} W \hat{\boldsymbol{\alpha}} = W^{\mathrm{T}} \boldsymbol{y} \tag{3.3}$$

can be written as a set of three matrix equations:

$$n\hat{\mu} + \mathbf{1}_n^{\mathrm{T}} X_1 \hat{\boldsymbol{\beta}} + \mathbf{1}_n^{\mathrm{T}} X_2 \hat{\boldsymbol{\tau}} = \mathbf{1}_n^{\mathrm{T}} \boldsymbol{y}\,, \tag{3.4}$$

$$X_1^{\mathrm{T}} \mathbf{1}_n \hat{\mu} + X_1^{\mathrm{T}} X_1 \hat{\boldsymbol{\beta}} + X_1^{\mathrm{T}} X_2 \hat{\boldsymbol{\tau}} = X_1^{\mathrm{T}} \boldsymbol{y}\,, \tag{3.5}$$

$$X_2^{\mathrm{T}} \mathbf{1}_n \hat{\mu} + X_2^{\mathrm{T}} X_1 \hat{\boldsymbol{\beta}} + X_2^{\mathrm{T}} X_2 \hat{\boldsymbol{\tau}} = X_2^{\mathrm{T}} \boldsymbol{y}\,. \tag{3.6}$$

$$\tag{3.7}$$

Above, the matrices $X_1^{\mathrm{T}} X_1 = \mathrm{diag}(k_1, \ldots, k_b)$ and $X_2^{\mathrm{T}} X_2 = \mathrm{diag}(n_1, \ldots, n_t)$ have simple forms as diagonal matrices with entries equal to the size of each block and the number of replications of each treatment, respectively.

The $t \times b$ matrix $N = X_2^{\mathcal{T}} X_1$ is particularly important in block designs, and is called the **incidence** matrix. Each of the $i$th row of $N$ indicates in which blocks the $i$th treatment occurs.

We can eliminate the explicit dependence on $\mu$ and $\boldsymbol{\beta}$ to find reduced normal equations for $\boldsymbol{\tau}$ by multiplying the middle equation by $X_2^{\mathrm{T}} X_1 (X_1^{\mathrm{T}} X_1)^{-1}$:

$$\begin{aligned} X_2^{\mathrm{T}} X_1 (X_1^{\mathrm{T}} X_1)^{-1} X_1^{\mathrm{T}} \mathbf{1}_n \hat{\mu} + X_2^{\mathrm{T}} X_1 (X_1^{\mathrm{T}} X_1)^{-1} X_1^{\mathrm{T}} X_1 \hat{\boldsymbol{\beta}} + X_2^{\mathrm{T}} X_1 (X_1^{\mathrm{T}} X_1)^{-1} X_1^{\mathrm{T}} X_2 \hat{\boldsymbol{\tau}} \\ = X_2^{\mathrm{T}} X_1 (X_1^{\mathrm{T}} X_1)^{-1} X_1^{\mathrm{T}} \mathbf{1}_n \hat{\mu} + X_2^{\mathrm{T}} X_1 \hat{\boldsymbol{\beta}} + X_2^{\mathrm{T}} X_1 (X_1^{\mathrm{T}} X_1)^{-1} X_1^{\mathrm{T}} X_2 \hat{\boldsymbol{\tau}} \\ = X_2^{\mathrm{T}} X_1 (X_1^{\mathrm{T}} X_1)^{-1} X_1^{\mathrm{T}} \boldsymbol{y} \end{aligned} \tag{3.8}$$

and subtracting from the final equation:

$$\begin{aligned} X_2^{\mathrm{T}} \left( \mathbf{1}_n - X_1 (X_1^{\mathrm{T}} X_1)^{-1} X_1^{\mathrm{T}} \mathbf{1}_n \right) \hat{\mu} + \left( X_2^{\mathrm{T}} X_1 - X_2^{\mathrm{T}} X_1 \right) \hat{\boldsymbol{\beta}} \\ + X_2^{\mathrm{T}} \left( I_n - X_1 (X_1^{\mathrm{T}} X_1)^{-1} X_1^{\mathrm{T}} \right) X_2 \hat{\boldsymbol{\tau}} \\ = X_2^{\mathrm{T}} \left( I_n - X_1 (X_1^{\mathrm{T}} X_1)^{-1} X_1^{\mathrm{T}} \right) \boldsymbol{y}\,. \end{aligned} \tag{3.9}$$

Clearly, a zero matrix is multiplying the block effects $\hat{\boldsymbol{\beta}}$. Also,

$$X_1 (X_1^{\mathrm{T}} X_1)^{-1} X_1^{\mathrm{T}} \mathbf{1}_n = \mathbf{1}_n\,,$$

as

$$X_1 (X_1^{\mathrm{T}} X_1)^{-1} = \bigoplus_{i=1}^{b} \frac{1}{k_i} \mathbf{1}_{k_i} = \begin{bmatrix} \frac{1}{k_1} \mathbf{1}_{k_1} & \mathbf{0}_{k_1} & \cdots & \mathbf{0}_{k_1} \\ \mathbf{0}_{k_2} & \frac{1}{k_2} \mathbf{1}_{k_2} & \cdots & \mathbf{0}_{k_2} \\ \vdots & & \ddots & \vdots \\ \mathbf{0}_{k_b} & \mathbf{0}_{k_b} & \cdots & \frac{1}{k_b} \mathbf{1}_{k_b} \end{bmatrix},$$

and hence

$$X_1(X_1^{\mathrm{T}}X_1)^{-1}X_1^{\mathrm{T}} = \bigoplus_{i=1}^{b} \frac{1}{k_i} J_{k_i} = \begin{bmatrix} \frac{1}{k_1}J_{k_1} & \mathbf{0}_{k_1\times k_2} & \cdots & \mathbf{0}_{k_1\times k_b} \\ \mathbf{0}_{k_2\times k_1} & \frac{1}{k_2}J_{k_2} & \cdots & \mathbf{0}_{k_2\times k_b} \\ \vdots & & \ddots & \vdots \\ \mathbf{0}_{k_b\times k_1} & \mathbf{0}_{k_b\times k_2} & \cdots & \frac{1}{k_b}J_{k_b} \end{bmatrix}.$$

Writing $H_1 = X_1(X_1^{\mathrm{T}}X_1)^{-1}X_1^{\mathrm{T}}$, we then get the reduced normal equations for $\boldsymbol{\tau}$:

$$X_2^{\mathrm{T}}\left(I_n - H_1\right)X_2\hat{\boldsymbol{\tau}} = X_2^{\mathrm{T}}\left(I_n - H_1\right)\boldsymbol{y}\,. \tag{3.10}$$

We can demonstrate the form of these matrices through our two examples.

For Example 3.1:

```
one <- rep(1, 4)
X1 <- kronecker(diag(1, nrow = 8), one)
X2 <- diag(1, nrow = 4)
X2 <- do.call("rbind", replicate(8, X2, simplify = FALSE))
#incidence matrix
N <- t(X2) %*% X1
X1tX1 <- t(X1) %*% X1 # diagonal
X2tX2 <- t(X2) %*% X2 # diagonal
H1 <- X1 %*% solve(t(X1) %*% X1) %*% t(X1)
ones <- H1 %*% rep(1, 32) # H1 times vector of 1s is also a vector of 1s
A <- t(X2) %*% X2 - t(X2) %*% H1 %*% X2 # X2t(I - H1)X2
qr(A)$rank # rank 3
X2tH1 <- t(X2) %*% H1 # adjustment to y
W <- cbind(ones, X1, X2) # overall model matrix
qr(W)$rank # rank 11 (t+b - 1)
```

For Example 3.2:

```
one <- rep(1, 3)
X1 <- kronecker(diag(1, nrow = 4), one)
X2 <- matrix(
        c(1, 0, 0, 0,
          0, 1, 0, 0,
          0, 0, 1, 0,
          1, 0, 0, 0,
          0, 1, 0, 0,
          0, 0, 0, 1,
          1, 0, 0, 0,
          0, 0, 1, 0,
          0, 0, 0, 1,
```

```
          0, 1, 0, 0,
          0, 0, 1, 0,
          0, 0, 0, 1), nrow = 12, byrow = T
)
#incidence matrix
N <- t(X2) %*% X1
X1tX1 <- t(X1) %*% X1 # diagonal
X2tX2 <- t(X2) %*% X2 # diagonal
H1 <- X1 %*% solve(t(X1) %*% X1) %*% t(X1)
ones <- H1 %*% rep(1, 12) # H1 times vector of 1s is also a vector of 1s
A <- t(X2) %*% X2 - t(X2) %*% H1 %*% X2 # X2t(I - H1)X2
qr(A)$rank # rank 3
X2tH1 <- t(X2) %*% H1 # adjustment to y
W <- cbind(ones, X1, X2) # overall model matrix
qr(W)$rank # rank 7 (t+b - 1)
```

Notice that if we write $X_{2|1} = (I_n - H_1)X_2$, then the reduced normal equations become

$$X_{2|1}^{\mathrm{T}} X_{2|1} \hat{\boldsymbol{\tau}} = X_{2|1}^{\mathrm{T}} \boldsymbol{y}\,,$$

which have the same form as the CRD in Chapter 2 albeit with a different $X_{2|1}$ matrix as we are adjusting for more complex nuisance parameters.

In general, the solution of these equations will depend on the exact form of the design. For the randomised complete block design, the solution turns out to be straighforward (see Section @ref(#rcbd) below). By default, to fit model (3.2), the `lm` function in `R` applies the constraint $\tau_t = \beta_b = 0$, and removes the corresponding columns from $X_1$ and $X_2$, to leave a $W$ matrix with full column rank. Clearly, this solution is not unique but, as with CRDs, we will identify uniquely estimatable combinations of the model parameters (and use `emmeans` to extract these estimates from an `lm` object).

## 3.3 Analysis of variance

As was the case with the CRD, it can be shown that any solution to the normal equations (3.3) will produce a unique solution to $\widehat{W\alpha}$, and hence a unique analysis of variance decomposition can be obtained.

For a block experiment, the ANOVA table is comparing the full model (3.2), the model containing the block effects

$$\boldsymbol{y} = \mu\mathbf{1} + X_1\boldsymbol{\beta} + \boldsymbol{\varepsilon} \tag{3.11}$$

and the null model

$$\boldsymbol{y} = \mu\mathbf{1} + \boldsymbol{\varepsilon}\,, \tag{3.12}$$

and has the form:

| Source | Degrees of freedom | Sums of squares | Mean square |
|---|---|---|---|
| Blocks | $b-1$ | RSS (3.12) - RSS (3.11) | |
| Treatments | $t-1$ | RSS (3.11) - RSS (3.2) | [RSS (3.11) - RSS (3.2)] / $(t-1)$ |
| Residual | $n-b-t+1$ | RSS (3.2) | RSS (3.2) / $(n-b-t+1)$ |
| Total | $n-1$ | RSS (3.12) | |

We test the hypothesis $H_0 : \tau_1 = \cdots = \tau_t = 0$ at the $100\alpha\%$ significance level by comparing the ratio of treatment and residual mean squares to the $1-\alpha$ quantile of an $F$ distribution with $t-1$ and $n-b-t+1$ degrees of freedom.

For Example 3.1, we obtain the following ANOVA.

```
bar.lm <- lm(strength ~ block + coating, data = bar)
anova(bar.lm)
```

```
## Analysis of Variance Table
##
## Response: strength
##           Df Sum Sq Mean Sq F value Pr(>F)
## block      7    215      31    0.55 0.7903
## coating    3   1310     437    7.75 0.0011 **
## Residuals 21   1184      56
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Clearly, the null hypothesis of no treatment effect is rejected. The `anova` function also compares the block mean square to the residual mean square to perform a test of the hypothesis $H_0 : \beta_1 = \cdots = \beta_b = 0$. This is not a hypothesis that should usually be tested. The blocks are a nuisance factor and are generally a feature of the experimental process that has not been subject to randomisation; we are not interested in testing for block-to-block differences.[3]

For Example 3.2, we get the ANOVA table:

```
tyre.lm <- lm(wear ~ block + compound, data = tyre)
anova(tyre.lm)
```

[3] `R` and `anova` don't, of course, know that this is a block design or that a blocking factor is being tested.

```
## Analysis of Variance Table
##
## Response: wear
##           Df Sum Sq Mean Sq F value  Pr(>F)
## block      3  39123   13041    37.2 0.00076 ***
## compound   3  20729    6910    19.7 0.00335 **
## Residuals  5   1751     350
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Again, the null hypothesis is rejected, and hence we should investigate which tyre compounds differ in their mean response.

The residual mean square for model (3.2) also provides an unbiased estimate, $s^2$, of $\sigma^2$, the variability of the $\varepsilon_{ijl}$, *assuming the unit-block-treatment model is correct.*

```
bar.s2 <- summary(bar.lm)$sigma^2
tyre.s2 <- summary(tyre.lm)$sigma^2
```

For Example 3.1, $s^2 = 56.3869$ and for Example 3.2, $s^2 = 350.1833$.

## 3.4 Randomised complete block designs

A randomised complete block design (RCBD) has each treatment replicated exactly once in each block, that is $n_{ij} = 1$ for $i = 1, \ldots, b; j = 1, \ldots, t$. Therefore each block has common size $k_1 = \cdots = k_b = t$. The $t$ treatments are randomised to the $t$ units in each block. We can drop the index $l$ from our unit-block-treatment model, as every treatment is replicated just once:

$$y_{ij} = \mu + \beta_i + \tau_j + \varepsilon_{ij}\,, \qquad i = 1, \ldots, b; j = 1, \ldots, t\,.$$

For an RCBD, the matrix $X_{2|1}$ has the form

$$\begin{aligned} X_{2|1} &= (I_n - H_1)X_2 \\ &= X_2 - H_1 X_2 \\ &= X_2 - \frac{1}{t} J_{n\times t}\,, \end{aligned} \tag{3.13}$$

following from the fact that

$$H_1 X_2 = X_1 (X_1^{\mathrm{T}} X_1)^{-1} X_1^{\mathrm{T}} X_2 \tag{3.14}$$
$$= \frac{1}{t} X_1 X_1^{\mathrm{T}} X_2 \tag{3.15}$$
$$= \frac{1}{t} X_1 N^{\mathrm{T}} \tag{3.16}$$
$$= \frac{1}{t} X_1 J_{b \times t} \tag{3.17}$$
$$= \frac{1}{t} J_{n \times t}\,, \tag{3.18}$$

as for a RCBD $X_1^{\mathrm{T}} X_1 = \mathrm{diag}(k_1, \ldots, k_b) = tI_b$ and $X_2^{\mathrm{T}} X_1 = N = J_{t \times b}$.

Comparing (3.13) to the form of $X_{2|1}$ for a CRD, equation (2.6), we see that for the RCBD, $X_{2|1}$ has the same form as a CRD with $b$ replicates of each treatment (that is, $n_i = b$ for $i = 1, \ldots, t$). This is a powerful result, as it tells us:

- The reduced normal equations for the RCBD take the same form as for the CRD,

  $$\hat{\tau}_j - \hat{\tau}_w = \bar{y}_{.j} - \bar{y}_{..}\,,$$

  with $\hat{\tau}_w = \frac{1}{t}\sum_{j=1}^{t} \hat{\tau}_j$, $\bar{y}_{.j} = \frac{1}{b}\sum_{i=1}^{b} y_{ij}$ and $\bar{y}_{..} = \frac{1}{n}\sum_{i=1}^{b}\sum_{j=1}^{t} y_{ij}$. Hence, as with a CRD, we can estimate any contrast $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{\tau}$, having $\sum_{j=1}^{t} c_j = 0$, with estimator

  $$\widehat{\boldsymbol{c}^{\mathrm{T}}\boldsymbol{\tau}} = \sum_{j=1}^{t} c_j \bar{y}_{.j}\,.$$

  Hence, the **point estimate** for a contrast $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{\tau}$ is exactly the same as would be obtained by ignoring blocks and treating the experiment as a CRD with $n = bt$ and $n_i = b$, for $i = 1, \ldots, t$.

- Inference for a contrast takes exactly the same form as for a CRD (Section 2.5), with in particular:

  $$\mathrm{var}\left(\widehat{\boldsymbol{c}^{\mathrm{T}}\boldsymbol{\tau}}\right) = \frac{\sigma^2}{b}\sum_{j=1}^{t} c_j^2\,,$$

  and

  $$\widehat{\boldsymbol{c}^{\mathrm{T}}\boldsymbol{\tau}} \sim N\left(\boldsymbol{c}^{\mathrm{T}}\boldsymbol{\tau}, \frac{\sigma^2}{b}\sum_{j=1}^{t} c_j^2\right)\,.$$

Although these equations have the same form as for a CRD, note that $\sigma^2$ is representing different quantities in each case.

- In a CRD, $\sigma^2$ is the uncontrolled variation in the response *among all experimental units*.
- In a RCBD, $\sigma^2$ is the uncontrolled variation in the response *among all units within a common block*.

Block-to-block differences are modelled via inclusion of the block effects $\beta_i$ in the model, and hence if blocking is effective, we would expect $\sigma^2$ from a RCBD to be substantially smaller than from a corresponding CRD with $n_i = b$.

Example 3.1 is a RCBD. We can estimate the contrasts

$$\tau_1 - \tau_2\tau_1 - \tau_3\tau_1 - \tau_4$$

between coatings[4] using `emmeans`.

```
bar.emm <- emmeans::emmeans(bar.lm, ~ coating)
contrastv1.emmc <- function(levs)
  data.frame('t1 v t2' = c(1, -1, 0, 0), 't1 v t3' = c(1, 0, -1, 0),
  't1 v t4' = c(1, 0, 0, -1))
emmeans::contrast(bar.emm, 'contrastv1')
```

```
##  contrast estimate   SE df t.ratio p.value
##  t1.v.t2     -1.25 3.75 21  -0.333  0.7425
##  t1.v.t3     15.00 3.75 21   3.995  0.0007
##  t1.v.t4      4.00 3.75 21   1.065  0.2988
##
## Results are averaged over the levels of: block
```

It is important to once again adjust for mulitple comparisons. Here we can use a Bonferroni adjustment, and multiply each p-value by the number of tests (3). We obtain p-values of 1 (coating 1 versus 2), 0.002 (1 versus 3) and 0.8964 (2 versus 3). Hence, there is a significant difference between coatings 1 and 3, with $H_0 : \tau_1 = \tau_3$ rejected at the 1% significant level.

We can demonstrate the equivalence of the contrast point estimates between a RCBD and a CRD by fitting a unit-treatment model that ignores blocks:

```
bar_crd.lm <- lm(strength ~ coating, data = bar)
bar_crd.emm <- emmeans::emmeans(bar_crd.lm, ~ coating)
emmeans::contrast(bar_crd.emm, 'contrastv1')
```

```
##  contrast estimate   SE df t.ratio p.value
##  t1.v.t2     -1.25 3.54 28  -0.354  0.7263
```

[4]These contrasts measure the difference between the coating only made from ETPU and the three coatings with added fibres.

```
##  t1.v.t3     15.00 3.54 28   4.243  0.0002
##  t1.v.t4      4.00 3.54 28   1.132  0.2674
```

```
crd.s2 <- summary(bar_crd.lm)$sigma^2
rcbd.s2 <- summary(bar.lm)$sigma^2
```

As expected the point estimates of the three contrasts are identical. In this case, the standard error of each contrast is actually smaller assuming a CRD without blocks, suggesting block-to-block differences were actually small here (further evidence is provided by the small block sums of squares in the ANOVA table). Here the estimate of $\sigma$ from the RCBD is $s_{RCBD} = 7.5091$ and from the CRD is $s_{CRD} = 7.0698$, so for this example the unit-to-unit variation within and between blocks is not so different, and actually estimated to be slightly smaller in the CRD[5].

## 3.5 Orthogonal blocking

The equality of the point estimates from the RCBD and the CRD is a consequence of the block and treatment parameters in model (3.1) being **orthogonal**. That is, the least squares estimators for $\boldsymbol{\beta}$ and $\boldsymbol{\tau}$ are independent in the sense that the estimators obtained from model (3.2) are the same as those obtained from the sub-models

$$\boldsymbol{y} = \mu \mathbf{1}_n + X_1 \boldsymbol{\beta} + \boldsymbol{\varepsilon}\,,$$

and

$$\boldsymbol{y} = \mu \mathbf{1}_n + X_2 \boldsymbol{\tau} + \boldsymbol{\varepsilon}\,.$$

That is, the presence or absence of the block parameters does not affect the estimator of the treatment parameters (and vice versa).

A condition for $\boldsymbol{\beta}$ and $\boldsymbol{\tau}$ to be estimated orthogonally can be derived from normal equations (3.4) - (3.5). Firstly. we premultiply (3.4) by $\frac{1}{n} X_1^{\mathrm{T}} \mathbf{1}_n$ and substract it from (3.5):

$$\begin{aligned}
&\left(X_1^{\mathrm{T}} \mathbf{1}_n - X_1^{\mathrm{T}} \mathbf{1}_n\right) \hat{\mu} + \left(X_1^{\mathrm{T}} X_1 - \frac{1}{n} X_1^{\mathrm{T}} \mathbf{1}_n \mathbf{1}_n^{\mathrm{T}} X_1\right) \hat{\boldsymbol{\beta}} + \left(X_1^{\mathrm{T}} X_2 - \frac{1}{n} X_1^{\mathrm{T}} \mathbf{1}_n \mathbf{1}_n^{\mathrm{T}} X_2\right) \hat{\boldsymbol{\tau}} \\
&= X_1^{\mathrm{T}} \left(I_n - \frac{1}{n} J_n\right) X_1 \hat{\boldsymbol{\beta}} + X_1^{\mathrm{T}} \left(I_n - \frac{1}{n} J_n\right) X_2 \hat{\boldsymbol{\tau}} \\
&= X_1^{\mathrm{T}} \left(I_n - \frac{1}{n} J_n\right) \boldsymbol{y}\,. \qquad (3.19)
\end{aligned}$$

[5] Of course, the CRD has seven more degrees of freedom for estimating $\sigma^2$ as block effects do not require estimation.

Secondly, we premultiply (3.4) by $\frac{1}{n}X_2^{\mathrm{T}}\mathbf{1}_n$ and substract it from (3.6):

$$X_2^{\mathrm{T}}\left(I_n - \frac{1}{n}J_n\right)X_1\hat{\boldsymbol{\beta}} + X_2^{\mathrm{T}}\left(I_n - \frac{1}{n}J_n\right)X_2\hat{\boldsymbol{\tau}} = X_2^{\mathrm{T}}\left(I_n - \frac{1}{n}J_n\right)\boldsymbol{y}\,. \quad (3.20)$$

For equations (3.19) and (3.20) to be independent, we require

$$X_2^{\mathrm{T}}\left(I_n - \frac{1}{n}J_n\right)X_1 = \mathbf{0}_{t\times b}\,.$$

Hence, we obtain the following condition on the incidence matrix $N = X_2^{\mathrm{T}}X_1$ for a block design to be orthogonal:

$$N = \frac{1}{n}X_2^{\mathrm{T}}J_nX_1 \quad (3.21)$$

$$= \frac{1}{n}\boldsymbol{n}\boldsymbol{k}^{\mathrm{T}}\,, \quad (3.22)$$

where $\boldsymbol{n}^{\mathrm{T}} = (n_1, \ldots, n_t)$ is the vector of treatment replications and $\boldsymbol{k}^{\mathrm{T}} = (k_1, \ldots, k_b)$ is the vector of block sizes.

The most common orthogonal block design for unstructured treatments is the RCBD, which has $n = bt$, $\boldsymbol{n} = b\mathbf{1}_t$, $\boldsymbol{k} = t\mathbf{1}_b$, and

$$N = J_{t\times b} \qquad = \frac{1}{bt}\boldsymbol{n}\boldsymbol{k}^{\mathrm{T}}\,. \quad (3.23)$$

Hence, the condition for orthogonality is met. In an orthogonal design, such as a RCBD, all information about the treatment comparisons is contained in comparisons made within blocks. For more complex blocking structures, such as incomplete block designs, this is not the case. We shall see orthogonal blocking again in Chapter 5.

## 3.6 Balanced incomplete block designs

When the blocks sizes are less than the number of treatments, i.e. $k_i < t$ for all $i = 1, \ldots, b$, by necessity the design is incomplete, in that not all treatments can be allocated to every block. We will restrict ourselves now to considering binary designs with common block size $k < t$. In a binary design, each treatment occurs within a block either 0 or 1 times ($n_{ij} = 0$ or $n_{ij} = 1$).

Example 3.2 is an example of an incomplete design with $k = 3 < t = 4$. For incomplete designs, it is often useful to study the *treatment concurrence* matrix, given by $NN^{\mathrm{T}}$.

```
N <- matrix(
  c(1, 1, 1, 0,
    1, 1, 0, 1,
    1, 0, 1, 1,
    0, 1, 1, 1),
  nrow = 4, byrow = T
)
N %*% t(N)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    3    2    2    2
## [2,]    2    3    2    2
## [3,]    2    2    3    2
## [4,]    2    2    2    3
```

This matrix has the number of treatment replications, $n_j$, on the diagonal and the off-diagonal elements are equal to the number of blocks within which each pair of treatments occurs together. We will denote as $\lambda_{ij}$ the number of blocks that contain both treatment $i$ and treatment $j$ ($i \neq j$). For Example 3.2, $\lambda_{ij} = 2$ for all $i, j = 1, \ldots, 4$; that is, each pair of treatments occurs together in two blocks.

**Definition 3.1.** A **balanced incomplete block design** (BIBD) is an incomplete block design with $k < t$ that meets three requirements:

1. The design is binary.
2. Each treatment is applied to a unit in the same number of blocks. It follows that the common number of units applied to each treatment must be $r = n_j = bk/t$ ($j = 1, \ldots, t$), where $n = bk$. (Sometimes referred to as first-order balance).
3. Each pair of treatments is applied to two units in the same number of blocks, that is $\lambda_{ij} = \lambda$. (Sometimes referred to as second-order balance).

   In fact, we can deduce that $\lambda(t-1) = r(k-1)$. To see this, focus on treatment 1. This treatment occurs in $r$ blocks, and in each of these blocks, it occurs together with $k-1$ other treatments. But also, treatment 1 occurs $\lambda$ times with each of the other $t-1$ treatments. Hence $\lambda(t-1) = r(k-1)$, or $\lambda = r(k-1)/(t-1)$.

The design in Example 3.2 is a BIBD with $b = 4$, $k = 3$, $t = 4$, $r = 4 \times 3/4 = 3$, $\lambda = 3 \times (3-1)/(4-1) = 2$.

### 3.6.1 Construction of BIBDs

BIBDs do not exist for all combinations of values of $t$, $k$ and $b$. In particular, we must ensure

- $r = bk/t$ is integer, and

- $\lambda = r(k-1)/(t-1)$ is integer.

In general, we can always construct a BIBD for $t$ treatments in $b = \binom{t}{k}$ blocks of size $k$, although it may not be the smallest possible BIBD. Each of the possible choices of $k$ treatments from the total $t$ forms one block. Such a design will have $r = \binom{t-1}{k-1}$ and $\lambda = \binom{t-2}{k-2}$. The design in Example 3.2 was constructed this way, with $b = 4$, $r = 3$ and $\lambda = 2$.

Sometimes, smaller BIBDs that satisfy the two conditions above can be constructed. Finding these designs is an combinatorial problem, and tables of designs are available in the literature[6]. A large collection of BIBDs has also been catalogued in the `R` package `ibd`. The function `bibd` generates BIBDs for given values of $t$, $b$, $r$, $k$ and $\lambda$, or returns a message that a design is not available for those values. We can use the function to find the design used in Example 3.2.

```
tyre.bibd <- ibd::bibd(v = 4, b = 4, r = 3, k = 3, lambda = 2) # note, v is the notation for the
tyre.bibd$N # incidence matrix
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    1    0    1
## [2,]    0    1    1    1
## [3,]    1    0    1    1
## [4,]    1    1    1    0
```

We can also use the package to find a design for bigger experiments, for example $t = 8$ treatments in $b = 14$ blocks of size $k = 4$. Here, $r = 14 \times 4/8 = 7$ and $\lambda = 7 \times 3/7 = 3$.

```
larger.bibd <- ibd::bibd(v = 8, b = 14, r = 7, k = 4, lambda = 3)
larger.bibd$N
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14]
## [1,]    1    0    0    1    0    1    0    1    1     0     0     0     1     1
## [2,]    0    0    0    0    0    0    1    0    1     1     1     1     1     1
## [3,]    0    0    0    1    1    1    0    1    0     1     1     1     0     0
## [4,]    1    0    1    1    1    0    1    0    0     0     0     1     0     1
## [5,]    1    1    1    1    0    0    0    0    0     1     1     0     1     0
## [6,]    1    1    0    0    1    0    1    1    1     1     0     0     0     0
## [7,]    0    1    1    0    0    1    1    1    0     0     1     0     0     1
## [8,]    0    1    1    0    1    1    0    0    1     0     0     1     1     0
```

Although larger than the examples we have considered before, this design is small compared to the design that would be obtained from the naive construction above with $b = \binom{t}{k} = \binom{8}{4} = 70$ blocks.

[6] See Cochran and Cox (1957) and Fisher and Yates (1963)

### 3.6.2 Reduced normal equations

It can be shown that the reduced normal equations (3.10) for a BIBD can be written as

$$\left(I_t - \frac{1}{t}J_t\right)\hat{\boldsymbol{\tau}} = \frac{k}{\lambda t}\left(X_2^{\mathrm{T}} - \frac{1}{k}NX_1^{\mathrm{T}}\right)\boldsymbol{y}\,. \tag{3.24}$$

Equation (3.24) defines a series of $t$ equations of the form

$$\begin{aligned}\hat{\tau}_j - \hat{\tau}_w &= \frac{k}{\lambda t}\left(\sum_{i=1}^{b} n_{ij}y_{ij} - \frac{1}{k}\sum_{i=1}^{b} n_{ij}\sum_{j=1}^{t} n_{ij}y_{ij}\right)\\ &= \frac{k}{\lambda t}q_j\,,\end{aligned}$$

with $q_j = \sum_{i=1}^{b} n_{ij}y_{ij} - \frac{1}{k}\sum_{i=1}^{b} n_{ij}\sum_{j=1}^{t} n_{ij}y_{ij}$.

Notice that unlike for the RCBD, the reduced normal equations for a BIBD do not correspond to the equations for a CRD. Although the first term in $q_i$ is the sum of the responses for the $j$th treatment (mirroring the CRD), the second term is no longer the overall sum (or average) of the responses. In fact, for $q_j$ this second term is an adjusted total, just involving observations from those blocks that contain treatment $j$.

### 3.6.3 Estimation and inference

As with the CRD and RCD we can estimate contrasts in the $\tau_i$, with estimator

$$\widehat{\boldsymbol{c}^{\mathrm{T}}\boldsymbol{\tau}} = \frac{k}{\lambda t}\sum_{j=1}^{t} c_j q_j\,.$$

Due to the form of the reduced normal equations for the BIBD, the estimator is no longer just a linear combination of the treatment means; $q_j$ includes a term that adjusts for the blocks in which treatment $j$ has occurred.

The simplest way to derive the variance of this estimator is to rewrite it in the form

$$\widehat{\boldsymbol{c}^{\mathrm{T}}\boldsymbol{\tau}} = \frac{k}{\lambda t}\boldsymbol{c}^{\mathrm{T}}\boldsymbol{q}\,,$$

with $\boldsymbol{q} = \left(X_2^{\mathrm{T}} - \frac{1}{k}NX_1^{\mathrm{T}}\right)\boldsymbol{y}$. Then

$$\mathrm{var}\left(\widehat{\boldsymbol{c}^{\mathrm{T}}\boldsymbol{\tau}}\right) = \frac{k^2}{\lambda^2 t^2}\boldsymbol{c}^{\mathrm{T}}\mathrm{var}(\boldsymbol{q})\boldsymbol{c}\,.$$

Recalling that $NN^{\mathrm{T}}$ is the treatment coincidence matrix, the variance-covariance matrix of $\boldsymbol{q}$ is given by

$$\begin{aligned}
\mathrm{var}(\boldsymbol{q}) &= \left(X_2^{\mathrm{T}} - \frac{1}{k}NX_1^{\mathrm{T}}\right)\left(X_2 - \frac{1}{k}X_1N\right)\sigma^2 \\
&= \sigma^2\left\{rI_t - \frac{1}{k}NN^{\mathrm{T}}\right\} \\
&= \sigma^2\left\{rI_t - \frac{1}{k}\left[(r-\lambda)I_t + \lambda J_t\right]\right\} \\
&= \sigma^2\left\{\left(\frac{r(k-1)+\lambda}{k}\right)I_t - \frac{\lambda}{k}J_t\right\} \\
&= \sigma^2\left\{\left(\frac{\lambda(t-1)+\lambda}{k}\right)I_t - \frac{\lambda}{k}J_t\right\} \\
&= \sigma^2\left\{\frac{\lambda t}{k}I_t - \frac{\lambda}{k}J_t\right\}.
\end{aligned}$$

Hence

$$\begin{aligned}
\mathrm{var}\left(\widehat{\boldsymbol{c}^{\mathrm{T}}\boldsymbol{\tau}}\right) &= \frac{k^2}{\lambda^2 t^2}\boldsymbol{c}^{\mathrm{T}}\left(\frac{\lambda t}{k}I_t - \frac{\lambda}{k}J_t\right)\boldsymbol{c}\sigma^2 \\
&= \frac{k\sigma^2}{\lambda t}\boldsymbol{c}^{\mathrm{T}}\boldsymbol{c} \\
&= \frac{k\sigma^2}{\lambda t}\sum_{j=1}^{t} c_i^2,
\end{aligned}$$

as $\boldsymbol{c}^{\mathrm{T}}J_t = \boldsymbol{0}$ as $\sum_{j=1}^{t} c_j = 0$. The estimator is also unbiased ($E(\widehat{\boldsymbol{c}^{\mathrm{T}}\boldsymbol{\tau}}) = \boldsymbol{c}^{\mathrm{T}}\boldsymbol{\tau}$), and hence the sampling distribution, upon which inference can be based, is given by

$$\widehat{\boldsymbol{c}^{\mathrm{T}}\boldsymbol{\tau}} \sim N\left(\boldsymbol{c}^{\mathrm{T}}\boldsymbol{\tau}, \frac{k\sigma^2}{\lambda t}\sum_{j=1}^{t} c_j^2\right).$$

Returning to Example 3.2, we can use these results to estimate all pairwise differences between the four treatments. Firstly, we directly calculate the $q_j$ from treatment and block sums, using the incidence matrix.

```
trtsum <- aggregate(wear ~ compound, data = tyre, FUN = sum)[, 2]
blocksum <- aggregate(wear ~ block, data = tyre, FUN = sum)[, 2]
q <- trtsum - N %*% blocksum / 3
```

```
C <- matrix(
  c(1, -1, 0, 0,
    1, 0, -1, 0,
    1, 0, 0, -1,
    0, 1, -1, 0,
    0, 1, 0, -1,
    0, 0, 1, -1),
  ncol = 4, byrow = T
)
k <- 3; lambda <- 2; t <- 4
pe <- k * C %*% q / (lambda * t) # point estimates
se <- sqrt(2 * k * tyre.s2 / (lambda * t)) # st error (same for each contrast)
t.ratio <- pe / se
p.value <- 1 - ptukey(abs(t.ratio) * sqrt(2), 4, 5)
data.frame(Pair = c('1v2', '1v3', '1v4', '2v3', '2v4', '3v4'),
  Estimate = pe, St.err = se, t.ratio = t.ratio,
           p.value = p.value, reject = p.value < 0.05)
```

```
##   Pair Estimate St.err t.ratio  p.value reject
## 1  1v2   -4.375  16.21  -0.270 0.992273  FALSE
## 2  1v3  -76.250  16.21  -4.705 0.019509   TRUE
## 3  1v4 -100.875  16.21  -6.225 0.005912   TRUE
## 4  2v3  -71.875  16.21  -4.435 0.024757   TRUE
## 5  2v4  -96.500  16.21  -5.955 0.007188   TRUE
## 6  3v4  -24.625  16.21  -1.519 0.491534  FALSE
```

Secondly, we can use `emmeans` to generate the same output from an `lm` object.

```
tyre.emm <- emmeans::emmeans(tyre.lm, ~ compound)
pairs(tyre.emm)
```

```
##  contrast              estimate   SE df t.ratio p.value
##  compound1 - compound2    -4.37 16.2  5  -0.270  0.9923
##  compound1 - compound3   -76.25 16.2  5  -4.705  0.0195
##  compound1 - compound4  -100.87 16.2  5  -6.225  0.0059
##  compound2 - compound3   -71.88 16.2  5  -4.435  0.0248
##  compound2 - compound4   -96.50 16.2  5  -5.955  0.0072
##  compound3 - compound4   -24.62 16.2  5  -1.519  0.4915
##
## Results are averaged over the levels of: block
## P value adjustment: tukey method for comparing a family of 4 estimates
```

For this experiment, treatments 1 and 3, 1 and 4, 2 and 3, and 2 and 4 are significantly different at an experiment-wise type I error rate of 5%.

## 3.7 Exercises

1. Consider the below randomised complete block design for comparing two catalysts, $A$ and $B$, for a chemical reaction using six batches of material. The response is the yield (%) from the reaction.

| Catalyst | Batch 1 | Batch 2 | Batch 3 | Batch 4 | Batch 5 | Batch 6 |
|---|---|---|---|---|---|---|
| A | 9 | 19 | 28 | 22 | 18 | 8 |
| B | 10 | 22 | 30 | 21 | 23 | 12 |

   i. Write down a unit-block-treatment model for this example.
   ii. Test if there is a significant difference between catalysts at the 5% level.
   iii. Fit a unit-treatment model ignoring blocks and test again for a difference between catalysts. Comment on difference between this analysis and the one including blocks.

Solution

i. The unit-block-treatment model for this RCBD is given by

$$y_{ij} = \mu + \beta_i + \tau_j + \varepsilon_{ij} \qquad i = 1, \ldots, 6; j = \mathrm{A}, \mathrm{B}\,, \tag{3.25}$$

where $y_{ij}$ is the yield from the application of catalyst $j$ to block $i$, $\mu$ is a constant parameter, $\beta_i$ is the effect of block $i$ and $\tau_j$ is the effect of treatment $j$. The errors follow a normal distribution $\varepsilon_{ij} \sim N(0, \sigma^2)$ with mean 0 and constant variance, and are assumed independent for different experimental units.

ii. To test if there is a difference between catalysts, we compare model (3.25) with the model that only includes block effects:

$$y_{ij} = \mu + \beta_i + \varepsilon_{ij} \qquad i = 1, \ldots, 6; j = \mathrm{A}, \mathrm{B}\,. \tag{3.26}$$

The relative difference in the residual mean squares between these two models follows an F distribution under $H_0 : \tau_1 = \tau_2 = 0$, see Section 3.3. These test statistic and associated p-value can be calculated in `R` using `anova`.

```
reaction <- data.frame(
  catalyst = factor(rep(c('A', 'B'), 6)),
  batch = factor(rep(1:6, rep(2, 6))),
  yield = c(9, 10, 19, 22, 28, 30, 22, 21, 18, 23, 8, 12)
)
reaction.lm <- lm(yield ~ batch + catalyst, data = reaction)
anova(reaction.lm)
```

```
## Analysis of Variance Table
##
## Response: yield
##           Df Sum Sq Mean Sq F value  Pr(>F)
## batch      5    561   112.2    48.1 0.00031 ***
## catalyst   1     16    16.3     7.0 0.04566 *
## Residuals  5     12     2.3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value is (just) less than 0.05, and so we can reject $H_0$ (no treatment difference) at the 5% level.

iii. The unit-treatment model is given by

$$y_{ij} = \mu + \tau_j + \varepsilon_{ij}\,, \qquad i = 1, \ldots, 6; j = 1, 2\,,$$

where now **all** the between unit differences, including any block-to-block differences, are modelled by the unit errors $\varepsilon_{ij}$. We can fit this model in R.

```
reaction2.lm <- lm(yield ~ catalyst, data = reaction)
anova(reaction2.lm)
```

```
## Analysis of Variance Table
##
## Response: yield
##           Df Sum Sq Mean Sq F value Pr(>F)
## catalyst   1     16    16.3    0.29    0.6
## Residuals 10    573    57.3
```

There is no longer evidence to reject the null hypothesis of no treatment difference. This is because the residual mean square is now so much larger (57.2667 versus 2.3333). The residual mean square is also an unbiased estimate of $\sigma^2$, and our estimate of $\sigma^2$ from the unit-treatment model is clearly much larger, as block-to-block variation has also been included.

2. Consider the data below obtained from an agricultural experiment in which six different fertilizers were given to a crop of blackcurrants in a field. The field was divided into four equal areas of land so that the land in each area was fairly homogeneous. Each area of land was further subdivided into six plots and one of the fertilizers, chosen by a random procedure, was applied to each plot. The yields of blackcurrants obtained from each plot were recorded (in lbs) and are given in Table 3.5. In this randomised block design the treatments are the six fertilizers and the blocks are the four areas of land.

```
blackcurrent <- data.frame(fertilizer = rep(factor(1:6), 4),
                block = rep(factor(1:4), rep(6, 4)),
```

Table 3.5: Blackcurrent experiment: yield (lbs) from six different fertilizers.

| Block | Ferilizer 1 | Ferilizer 2 | Ferilizer 3 | Ferilizer 4 | Ferilizer 5 | Ferilizer 6 |
|---|---|---|---|---|---|---|
| 1 | 14.5 | 13.5 | 11.5 | 13.0 | 15 | 12.5 |
| 2 | 12.0 | 10.0 | 11.0 | 13.0 | 12 | 13.5 |
| 3 | 9.0 | 9.0 | 14.0 | 13.5 | 8 | 14.0 |
| 4 | 6.5 | 8.5 | 10.0 | 7.5 | 7 | 8.0 |

```
                yield = c(14.5, 13.5, 11.5, 13.0, 15.0, 12.5,
                          12.0, 10.0, 11.0, 13.0, 12.0, 13.5,
                          9.0, 9.0, 14.0, 13.5, 8.0, 14.0,
                          6.5, 8.5, 10.0, 7.5, 7.0, 8.0)
)
knitr::kable(
 tidyr::pivot_wider(blackcurrent, names_from = fertilizer, values_from = yield),
 col.names = c("Block", paste("Ferilizer", 1:6)),
 caption = "Blackcurrent experiment: yield (lbs) from six different fertilizers."
)
```

Conduct a full analysis of this experiment, including

a. exploratory data analysis;
b. fitting an appropriate linear model, and conducting an F-test to compare a model that explains variation between the six fertilizers to the model only containing blocks;
c. Linear model diagnostics;
d. if appropriate, multiple comparisons of all pairwise differences between treatments.

Solution

a. We start with exploratory tabular and graphical analysis.

```
aggregate(yield ~ fertilizer, data = blackcurrent,
          FUN = function(x) c(mean = mean(x), sd = sd(x)))
aggregate(yield ~ block, data = blackcurrent,
          FUN = function(x) c(mean = mean(x), sd = sd(x)))
boxplot(yield ~ fertilizer, data = blackcurrent)
boxplot(yield ~ block, data = blackcurrent)
```

```
##   fertilizer yield.mean yield.sd
## 1          1     10.500    3.488
## 2          2     10.250    2.255
## 3          3     11.625    1.702
## 4          4     11.750    2.843
## 5          5     10.500    3.697
```

```
## 6          6     12.000    2.739
##   block yield.mean yield.sd
## 1     1     13.333    1.291
## 2     2     11.917    1.281
## 3     3     11.250    2.859
## 4     4      7.917    1.242
```

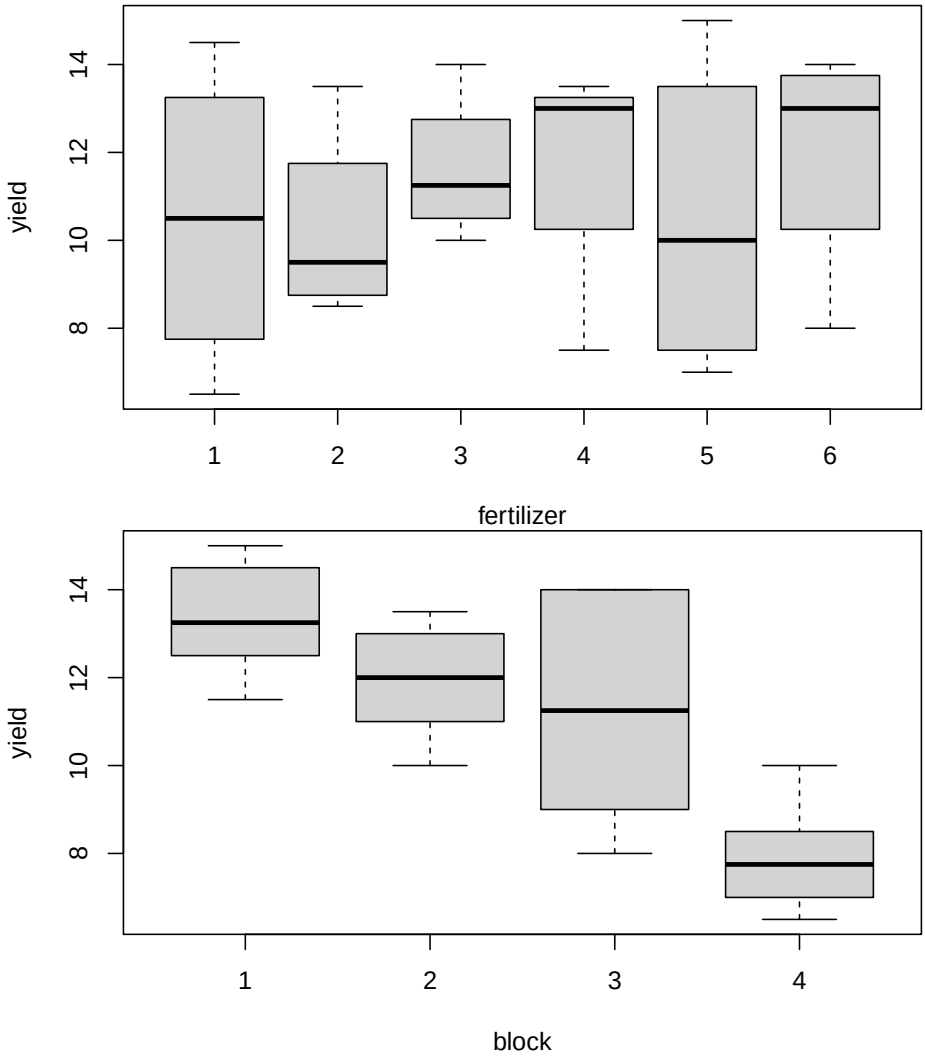



Figure 3.3: Blackcurrent experiment: yield against treatment and block.

There are substantial differences in average responses between blocks; differences between treatment means are smaller. These plots are only meaningful because this design is an RCBD, and each treatment occurs in

each block.

The unit-block-treatment model is *additive*; that is, we assume the effect of each treatment does not vary for each block. Therefore we also need to check that there are no substantive interactions between treatments and blocks. We will do this graphically.

```
with(blackcurrent,
  interaction.plot(fertilizer, block, yield, xlab = 'Treatment', ylab = 'Yield', trace.label
)
```

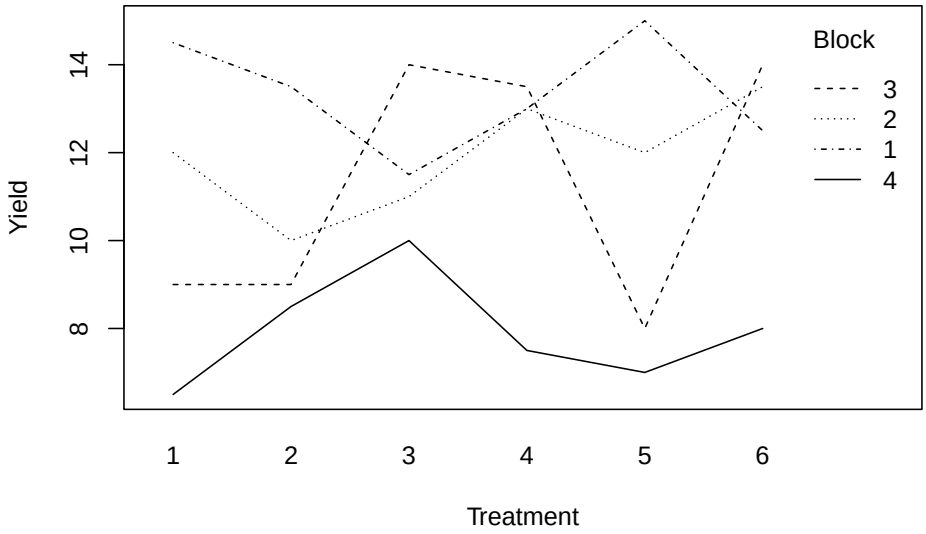



Figure 3.4: Blackcurrent experiment: treatment-block interaction plot.

Figure 3.4 plots the treatment means within each block. While there are clear differences between blocks, the differences between treatments do not seem to vary systematically between blocks.

b. We now fit a linear model, and perform an F-test for the hypothesis $H_0 : \tau_1 = \tau_2 = \tau_3 = \tau_4 = \tau_5 = \tau_6 = 0$.

```
blackcurrent.lm <- lm(yield ~ block + fertilizer, data = blackcurrent)
anova(blackcurrent.lm)
```

```
## Analysis of Variance Table
##
## Response: yield
##            Df Sum Sq Mean Sq F value Pr(>F)
## block       3   94.9   31.62    8.90 0.0013 **
## fertilizer  5   11.8    2.36    0.66 0.6564
## Residuals  15   53.3    3.55
## ---
```

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

The "treatment" row of the ANOVA table compares the model including blocks and treatments to that only containing blocks. This comparison tests the above null hypothesis. We can see here that there is no evidence to reject $H_0$ (p-value = 0.6564). This outcome is not surprising, given the tabular and graphical summaries we saw above. The block sum of squares is large, but we do not perform formal hypothesis testing for blocks.

Out of curiousity, we can also assess the "efficiency" of blocking by comparing the estimate of $\sigma^2$ from the block design with the estimate that would result from ignoring the blocks, treating the experiment as a CRD and fitting a unit-treatment model.

```
blackcurrent_crd.lm <- lm(yield ~ fertilizer, data = blackcurrent)
summary(blackcurrent_crd.lm)$sig^2 / summary(blackcurrent.lm)$sig^2
```

```
## [1] 2.316
```

The estimate of $\sigma^2$ from the CRD is more the two times greater than the estimate from the block design, meaning about 100% more observations would be needed in the CRD to get the same level of precision as the RCBD.

c. We now examine residual diagnostics, to check the assumptions of our model:

- constant variance, with respect to the mean response, the treatment and the block
- normality of residuals
- additive treatment and block effects (already assessed in Fig 3.4).

```
standres <- rstandard(blackcurrent.lm)
fitted <- fitted(blackcurrent.lm)
par(mfrow = c(1, 3), pty = "s")
with(blackcurrent, {
  plot(fertilizer, standres, xlab = "Treatment", ylab = "Standarised residuals")
  plot(block, standres, xlab = "Block", ylab = "Standarised residuals")
  plot(fitted, standres, xlab = "Fitted value", ylab = "Standarised residuals")
})
```

The plots in Figure 3.5 do not show any serious evidence of non-constant variance (maybe very slightly for blocks), and no large outliers.

```
par(pty = "s")
qqnorm(standres, main = "")
```

Figure 3.6 shows the residuals lie roughly on a straight line when plotted against theoretical normal quantiles, and hence the assumption of normally distributed errors seems reasonable.

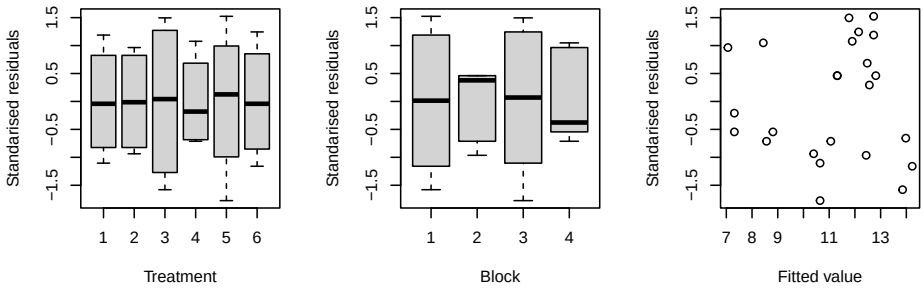



Figure 3.5: Blackcurrent experiment: Residuals against treatments (left), blocks (middle) and fitted values (right).

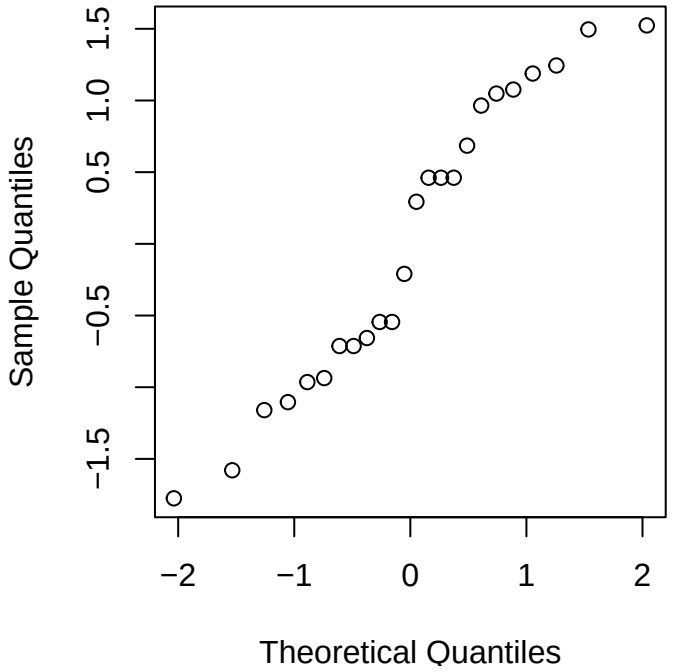



Figure 3.6: Blackcurrent experiment: Normal probability plot for standardised residuals.

d. There is no evidence of a difference between treatments, so we would not normally test each pairwise difference. However, if we did, we could use the following code.

```
blackcurrent.emm <- emmeans::emmeans(blackcurrent.lm, ~ fertilizer)
pairs(blackcurrent.emm)
```

```
##  contrast                  estimate   SE df t.ratio p.value
##  fertilizer1 - fertilizer2    0.250 1.33 15   0.188  1.0000
##  fertilizer1 - fertilizer3   -1.125 1.33 15  -0.844  0.9541
##  fertilizer1 - fertilizer4   -1.250 1.33 15  -0.938  0.9303
##  fertilizer1 - fertilizer5    0.000 1.33 15   0.000  1.0000
##  fertilizer1 - fertilizer6   -1.500 1.33 15  -1.125  0.8635
##  fertilizer2 - fertilizer3   -1.375 1.33 15  -1.031  0.9000
##  fertilizer2 - fertilizer4   -1.500 1.33 15  -1.125  0.8635
##  fertilizer2 - fertilizer5   -0.250 1.33 15  -0.188  1.0000
##  fertilizer2 - fertilizer6   -1.750 1.33 15  -1.313  0.7741
##  fertilizer3 - fertilizer4   -0.125 1.33 15  -0.094  1.0000
##  fertilizer3 - fertilizer5    1.125 1.33 15   0.844  0.9541
##  fertilizer3 - fertilizer6   -0.375 1.33 15  -0.281  0.9997
##  fertilizer4 - fertilizer5    1.250 1.33 15   0.938  0.9303
##  fertilizer4 - fertilizer6   -0.250 1.33 15  -0.188  1.0000
##  fertilizer5 - fertilizer6   -1.500 1.33 15  -1.125  0.8635
##
## Results are averaged over the levels of: block
## P value adjustment: tukey method for comparing a family of 6 estimates
```

3. Prove that $2k/\lambda t > 2/b$, and use this result to compare the precision of a pairwise treatment comparison from a BIBD with block size $k$ and an RCBD, both with $t$ treatments in $b$ blocks.

Solution

From the notes, $r = bk/t$ and hence $k = rt/b$. Therefore

$$\begin{aligned}\frac{2k}{\lambda t} &= \frac{2rt}{\lambda bt}\\ &= \frac{2r}{\lambda b}\\ &= \frac{2}{b} \times \frac{r}{\lambda}\\ &> \frac{2}{b},\end{aligned}$$

as $r/\lambda > 1$. The variance of the estimator of a pairwise comparison is

Table 3.6: BIBD for $t = 5$ treatments in $b = 5$ blocks of size $k = 4$.

| | | | | |
|---|---|---|---|---|
| Block 1 | 1 | 2 | 3 | 4 |
| Block 2 | 1 | 2 | 3 | 5 |
| Block 3 | 1 | 2 | 4 | 5 |
| Block 4 | 1 | 3 | 4 | 5 |
| Block 5 | 2 | 3 | 4 | 5 |

$$\text{var}_{\text{RCBD}}\left(\tau_i - \tau_j\right) = \frac{2\sigma^2}{b}\,,$$

for a RCBD with $b$ blocks, and

$$\text{var}_{\text{BIBD}}\left(\tau_i - \tau_j\right) = \frac{2k\sigma^2}{\lambda t}\,,$$

for a BIBD. Hence, we can see that the ratio of these variances is

$$\frac{kb}{\lambda t} > 1\,,$$

and hence the variance for a BIBD is larger, i.e. incomplete blocks results in less precise treatment comparisons.

4. Construct a BIBD for $t = 5$ treatments in $b = 5$ blocks of size $k = 4$ units. What are $r$ and $\lambda$ for your design? Compare your design to a RCBD via the efficiency for estimating a pairwise treatment difference.

Solution

Here, $b = \binom{t}{k} = \binom{5}{4} = 5$, and hence we can use the "naive" construction method from the notes, where we can form one block from each possible subset of $k = 4$ treatments. This design will have $r = \binom{4}{3} = 4$ and $\lambda = \binom{3}{2} = 3$.

```
bibd <- t(combn(1:5, 4))
rownames(bibd) <- paste("Block", 1:5)
knitr::kable(bibd, caption = "BIBD for $t = 5$ treatments in $b=5$ blocks of size $k=4$.")
```

From Question 3, the efficiency of this BIBD compared to a RCBD with $b = 5$ blocks of size $k = 5$ is given by $\frac{\lambda t}{kb} = 0.75$. So variances will be 25% larger under the BIBD.

5. Consider an experiment for testing the abrasion resistance of rubber-coated fabric. There are four types of material, denoted A - D. The response is the loss in weight in 0.1 milligrams (mg) over a standard period of time. The testing machine has four positions, so four samples of material can be tested at a time. Past experience suggests that there may be differences between

these positions, and there may be differences between each application of the testing machine (due to changes in set-up). Therefore, we have two blocking variables, "Position" and "Application". For this experiment, we use a **latin square** design, as follows.

```
fabric <- data.frame(material = factor(c('C', 'D', 'B', 'A', 'A', 'B', 'D', 'C',
                                         'D', 'C', 'A', 'B', 'B', 'A', 'C', 'D')),
                     position = rep(factor(1:4), 4),
                     application = rep(factor(1:4), rep(4, 4)),
                     weight = c(235, 236, 218, 268,
                                251, 241, 227, 229,
                                234, 273, 274, 226,
                                195, 270, 230, 225)
                       )
knitr::kable(
 tidyr::pivot_wider(fabric, id_cols = application, names_from = position,
                    values_from = material),
 col.names = c("Application", paste("Position", 1:4)))
```

| Application | Position 1 | Position 2 | Position 3 | Position 4 |
|---|---|---|---|---|
| 1 | C | D | B | A |
| 2 | A | B | D | C |
| 3 | D | C | A | B |
| 4 | B | A | C | D |

The blocking variables are represented as the rows and columns of the square; the latin letters represent the different treatments. A latin square of order $k$ is a $k \times k$ square of $k$ latin letters arranged so that each letter appears exactly once in each row and column (Sudoko squares are also examples of latin squares). To perform the experiment, the levels of the blocking factors are randomly assigned to the rows and the columns, and the different treatments to the letters. A latin square design is a special case of the more general class of **row column** designs for two blocking factors.

A suitable unit-block-treatment model for a latin square design has the form

$$y_{ijk} = \mu + \beta_i + \gamma_j + \tau_k + \varepsilon_{ijk}\,, \qquad i, j, k = 1, \ldots, t\,,$$

with $\mu$ a constant parameter, $\beta_i$ row block effects, $\gamma_j$ column block effects and $\tau_k$ the treatment effects. As usual, $\varepsilon_{ijk} \sim N(0, \sigma^2)$, with errors from different units assumed independent. Note that not all combinations of $i, j, k$ actually occur in the design; at the intersection of the $i$th row and $j$th column, only one of the $t$ treatments is applied.

a. Write down a set of normal equations for the model parameters.

b. It can be shown that the reduced normal equations for the treatment parameters $\tau_1, \ldots, \tau_t$ have the form

$$\hat{\tau}_k - \hat{\tau}_w = \bar{y}_{\cdot\cdot j} - \bar{y}_{\cdots},$$

with $\hat{\tau}_w = \frac{1}{t}\sum_{k=1}^{t} \hat{\tau}_k$, $\bar{y}_{\cdot\cdot k} = \frac{1}{t}\sum_{i=1}^{t}\sum_{j=1}^{t} n_{ijk} y_{ijk}$ (mean for treatment $k$) and $\bar{y}_{\cdots} = \frac{1}{n}\sum_{i=1}^{t}\sum_{j=1}^{t}\sum_{k=1}^{t} n_{ij} y_{ijk}$ (overall mean) where $n_{ijk} = 1$ if treatment $k$ occurs at the intersection of row $i$ and column $j$ and zero otherwise, and $\sum_{i=1}^{t}\sum_{j=1}^{t} n_{ijk} = t$ for all $k = 1, \ldots, t$.

Demonstrate that any contrast can therefore be estimated from this design, and derive the variance of the estimator of $\sum_{k=1}^{t} c_k \tau_k$.

The data for this experiment is as follows, where the entries in the table give the response for the corresponding treatment:

```
knitr::kable(
 tidyr::pivot_wider(fabric, id_cols = application, names_from = position,
                    values_from = weight),
 col.names = c("Application", paste("Position", 1:4)))
```

| Application | Position 1 | Position 2 | Position 3 | Position 4 |
|---|---|---|---|---|
| 1 | 235 | 236 | 218 | 268 |
| 2 | 251 | 241 | 227 | 229 |
| 3 | 234 | 273 | 274 | 226 |
| 4 | 195 | 270 | 230 | 225 |

c. For this data, test if there is a significant difference between materials. If there is, conduct multiple comparisons of all pairs at an experimentwise type I error rate of 5%.

Solution

a. We start by writing the model in matrix form:

$$\begin{aligned} \boldsymbol{y} &= \mathbf{1}_n \mu + X_1 \boldsymbol{\beta} + X_2 \boldsymbol{\gamma} + X_3 \boldsymbol{\tau} + \boldsymbol{\varepsilon} \\ &= W \boldsymbol{\alpha} + \boldsymbol{\varepsilon}, \end{aligned}$$

with $W = (\mathbf{1}_n, X_1, X_2, X_3)$, $X_1$, $X_2$ and $X_3$ being $n \times t$ model matrices for row blocks, column blocks and treatments, respectively, $\boldsymbol{\alpha}^{\mathrm{T}} = (\mu, \boldsymbol{\beta}^{\mathrm{T}}, \boldsymbol{\gamma}^{\mathrm{T}}, \boldsymbol{\tau}^{\mathrm{T}})$ being a $(1+3t)$-vector of parameters, and $\boldsymbol{\varepsilon} \sim N(\mathbf{0}_n, I_n \sigma^2)$.

The normal equations are given by $W^{\mathrm{T}} W \hat{\boldsymbol{\alpha}} = W^{\mathrm{T}} \boldsymbol{y}$, which can be expanded out to give the following four matrix equations:

$$n\hat{\mu} + \mathbf{1}_n^{\mathrm{T}} X_1 \hat{\boldsymbol{\beta}} + \mathbf{1}_n^{\mathrm{T}} X_2 \hat{\boldsymbol{\gamma}} + \mathbf{1}_n^{\mathrm{T}} X_3 \hat{\boldsymbol{\tau}} = \mathbf{1}_n^{\mathrm{T}} \boldsymbol{y}\,, \tag{3.27}$$

$$X_1^{\mathrm{T}} \mathbf{1}_n \hat{\mu} + X_1^{\mathrm{T}} X_1 \hat{\boldsymbol{\beta}} + X_1^{\mathrm{T}} X_2 \hat{\boldsymbol{\gamma}} + X_1^{\mathrm{T}} X_3 \hat{\boldsymbol{\tau}} = X_1^{\mathrm{T}} \boldsymbol{y}\,, \tag{3.28}$$

$$X_2^{\mathrm{T}} \mathbf{1}_n \hat{\mu} + X_2^{\mathrm{T}} X_1 \hat{\boldsymbol{\beta}} + X_2^{\mathrm{T}} X_2 \hat{\boldsymbol{\gamma}} + X_2^{\mathrm{T}} X_3 \hat{\boldsymbol{\tau}} = X_2^{\mathrm{T}} \boldsymbol{y}\,, \tag{3.29}$$

$$X_3^{\mathrm{T}} \mathbf{1}_n \hat{\mu} + X_3^{\mathrm{T}} X_1 \hat{\boldsymbol{\beta}} + X_3^{\mathrm{T}} X_2 \hat{\boldsymbol{\gamma}} + X_3^{\mathrm{T}} X_3 \hat{\boldsymbol{\tau}} = X_3^{\mathrm{T}} \boldsymbol{y}\,. \tag{3.30}$$

$$\tag{3.31}$$

b. From the reduced normal equations given in the question, we have

$$\begin{aligned}\sum_{k=1}^{t} c_k \left(\hat{\tau}_k - \hat{\tau}_w\right) &= \sum_{k=1}^{t} c_k \hat{\tau}_k \\ &= \sum_{k=1}^{t} c_k \left(\bar{y}_{..k} - \bar{y}_{...}\right) \\ &= \sum_{k=1}^{t} c_k \bar{y}_{..k}\,,\end{aligned}$$

as $\sum_{k=1}^{t} c_k \hat{\tau}_w = \sum_{k=1}^{t} c_k \bar{y}_{...} = 0$, as neither $\hat{\tau}_w$ or $\bar{y}_{...}$ depend on $k$. Hence,

$$\widehat{\boldsymbol{c}^{\mathrm{T}}\boldsymbol{\tau}} = \sum_{k=1}^{t} c_k \bar{y}_{..k}\,.$$

Or notice that the reduced normal equations for this design are the same as for a CRD with each treatment replicated $t$ times.

c. We conduct these hypothesis tests using `R`.

```
fabric.lm <- lm(weight ~ position + application + material, data = fabric)
anova(fabric.lm)
```

```
## Analysis of Variance Table
##
## Response: weight
##             Df Sum Sq Mean Sq F value  Pr(>F)
## position     3   1468     489    7.99 0.01617 *
## application  3    986     329    5.37 0.03901 *
## material     3   4621    1540   25.15 0.00085 ***
## Residuals    6    368      61
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The "material" line of the ANOVA table compares the models with and without the effects for material (but both models including the blocking factors). There is clearly a significant effect of material.

```
fabric.emm <- emmeans::emmeans(fabric.lm, ~ material)
fabric.pairs <- transform(pairs(fabric.emm))
dplyr::mutate(fabric.pairs, reject = p.value < 0.05)
```

```
##   contrast estimate    SE df t.ratio  p.value reject
## 1    A - B    45.75 5.534  6   8.267 0.000703   TRUE
## 2    A - C    24.00 5.534  6   4.337 0.019036   TRUE
## 3    A - D    35.25 5.534  6   6.370 0.002866   TRUE
## 4    B - C   -21.75 5.534  6  -3.930 0.029477   TRUE
## 5    B - D   -10.50 5.534  6  -1.897 0.320631  FALSE
## 6    C - D    11.25 5.534  6   2.033 0.274277  FALSE
```

Using an experiment-wise error rate of 5%, we see significant differences between materials A and B, A and C, A and D, and B and C.

# Chapter 4

# Factorial experiments

In Chapters 2 and 3, we assumed the objective of the experiment was to investigate $t$ **unstructured** treatments, defined only as a collection of distinct entities (drugs, advertisements, receipes, etc.). That is, there was not necessarily any explicit relationship between the treatments (although we could clearly choose which paticular comparisons between treatments were of interest via choice of contrast).

In many experiments, particularly in industry, engineering and the physical sciences, the treatments are actually defined via the choice of a **level** relating to each of a set of **factors**. We will focus on the commonly occurring case of factors at **two levels**. For example, consider the below experiment from the pharmaceutical industry.

**Example 4.1.** Desilylation experiment (Owen et al., 2001)

In this experiment, performed at GlaxoSmithKline, the aim was to optimise the desilylation[1] of an ether into an alcohol, which was a key step in the synthesis of a particular antibiotic. There were $t = 16$ treatments, defined via the settings of four different factors, as given in Table 4.1.

```
desilylation <- FrF2::FrF2(nruns = 16, nfactors = 4, randomize = F,
                           factor.names = list(temp = c(10, 20), time = c(19, 25),
                                               solvent = c(5, 7), reagent = c(1, 1.33)))
yield <- c(82.93, 94.04, 88.07, 93.97, 77.21, 92.99, 83.60, 94.38,
           88.68, 94.30, 93.00, 93.42, 84.86, 94.26, 88.71, 94.66)
desilylation <- data.frame(desilylation, yield = yield)
rownames(desilylation) <- paste("Trt", 1:16)
knitr::kable(desilylation,
             col.names = c("Temp (degrees C)", "Time (hours)", "Solvent (vol.)",
                           "Reagent (equiv.)", "Yield (%)"),
```

[1] Desilylation is a process of removing silyl, $SiH_3$ a silicon hydride, from a compound.

Table 4.1: Desilylation experiment: 16 treatments defined by settings of four factors, with response (yield).

| | Temp (degrees C) | Time (hours) | Solvent (vol.) | Reagent (equiv.) | Yield (%) |
|---|---|---|---|---|---|
| Trt 1 | 10 | 19 | 5 | 1 | 82.93 |
| Trt 2 | 20 | 19 | 5 | 1 | 94.04 |
| Trt 3 | 10 | 25 | 5 | 1 | 88.07 |
| Trt 4 | 20 | 25 | 5 | 1 | 93.97 |
| Trt 5 | 10 | 19 | 7 | 1 | 77.21 |
| Trt 6 | 20 | 19 | 7 | 1 | 92.99 |
| Trt 7 | 10 | 25 | 7 | 1 | 83.60 |
| Trt 8 | 20 | 25 | 7 | 1 | 94.38 |
| Trt 9 | 10 | 19 | 5 | 1.33 | 88.68 |
| Trt 10 | 20 | 19 | 5 | 1.33 | 94.30 |
| Trt 11 | 10 | 25 | 5 | 1.33 | 93.00 |
| Trt 12 | 20 | 25 | 5 | 1.33 | 93.42 |
| Trt 13 | 10 | 19 | 7 | 1.33 | 84.86 |
| Trt 14 | 20 | 19 | 7 | 1.33 | 94.26 |
| Trt 15 | 10 | 25 | 7 | 1.33 | 88.71 |
| Trt 16 | 20 | 25 | 7 | 1.33 | 94.66 |

```
            caption = "Desilylation experiment: 16 treatments defined
            by settings of four factors, with response (yield).")
```

Each treatment is defined by the choice of one of two levels for each of the four factors. In the `R` code above, we have used the function `FrF2` (from the package of the same name) to generate all $t = 2^4 = 16$ combinations of the two levels of these four factors. We come back to this function later in the chapter.

This **factorial treatment structure** lends itself to certain treatment contrasts being of natural interest.

## 4.1 Factorial contrasts

Throughout this chapter, we will assume there are no blocks or other restrictions on randomisation, and so we will assume a completely randomised design can be used. We start by assuming the same unit-treatment model as Chapter 2:

$$y_{ij} = \mu + \tau_i + \varepsilon_{ij}\,, \quad i = 1, \ldots, t; j = 1, \ldots, n_i\,, \tag{4.1}$$

where $y_{ij}$ is the response from the $j$th application of treatment $i$, $\mu$ is a constant parameter, $\tau_i$ is the effect of the $i$th treatment, and $\varepsilon_{ij}$ is the random individual

effect from each experimental unit with $\varepsilon_{ij} \sim N(0, \sigma^2)$ independent of other errors.

Now, the number of treatments $t = 2^f$, where $f$ equals the number of factors in the experiment.

For Example 4.1, we have $t = 2^4 = 16$ and $n_i = 1$ for all $i = 1, \ldots, 16$; that is, each of the 16 treatments are replicated once. In general, we shall assume common treatment replication $n_i = r \geq 1$.

If we fit model (4.1) and compute the ANOVA table, we notice a particular issue with this design.

```
desilylation <- data.frame(desilylation, trt = factor(1:16))
desilylation.lm <- lm(yield ~ trt, data = desilylation)
anova(desilylation.lm)
```

```
## Analysis of Variance Table
##
## Response: yield
##           Df Sum Sq Mean Sq F value Pr(>F)
## trt       15    427    28.5     NaN    NaN
## Residuals  0      0     NaN
```

All available degrees of freedom are being used to estimate parameters in the mean ($\mu$ and the treatment effects $\tau_i$). There are no degrees of freedom left to estimate $\sigma^2$. This is due to a lack of treatment replication. Without replication in the design, model (4.1) is **saturated**, with as many treatments as there are observations and an unbiased estimate of $\sigma^2$ cannot be obtained. We will return to this issue later.

### 4.1.1 Main effects

Studying Table 4.1, there are some comparisons between treatments which are obviously of interest. For example, comparing the average effect from the first 8 treatments with the average effect of the second 8, using

$$\boldsymbol{c}^{\mathrm{T}}\boldsymbol{\tau} = \sum_{i=1}^{t} c_i \tau_i \,,$$

with

$$\boldsymbol{c}^{\mathrm{T}} = (-\mathbf{1}_8^{\mathrm{T}}, \mathbf{1}_8^{\mathrm{T}})/8 \,.$$

```
desilylation.emm <- emmeans::emmeans(desilylation.lm, ~ trt)
reagent_me.emmc <- function(levs) data.frame('reagent m.e.' = rep(c(-1, 1), rep(8, 2)) / 8)
emmeans::contrast(desilylation.emm, 'reagent_me')
```

```
##  contrast     estimate  SE df t.ratio p.value
##  reagent.m.e.     3.09 NaN  0     NaN     NaN
```

This contrast compares the average treatment effect from the 8 treatments which have `reagent` set to its low level (1 equiv.) to the average effect from the 8 treatments which have `reagent` set to its high level. This is a "fair" comparison, as both of these sets of treatments have each of the combinations of the factors `temp`, `time` and `solvent` occuring equally often (twice here). Hence, the **main effect** of `reagent` is averaged over the levels of the other three factors.

As in Chapter 2, we can estimate this treatment contrast by applying the same contrast coefficients to the treatment means,

$$\widehat{\boldsymbol{c}^{\mathrm{T}}\boldsymbol{\tau}} = \sum_{i=1}^{t} c_i \bar{y}_{i\cdot} \,,$$

where, for this experiment, each $\bar{y}_{i\cdot}$ is the mean of a single observation (as there is no treatment replication). We see that inference about this contrast is not possible, as no standard error can be obtained.

**Definition 4.1.** The **main effect** of a factor $A$ is defined as the difference in the average response from the high and low levels of the factor

$$\mathrm{ME}(A) = \bar{y}(A+) - \bar{y}(A-) \,,$$

where $\bar{y}(A+)$ is the average response when factor $A$ is set to its high level, averaged across all combinations of levels of the other factors (with $\bar{y}(A+)$ defined similarly for the low level of $A$).

As we have averaged the response across the levels of the other factors, the intepretation of the main effect extends beyond this experiment. That is, we can use it to infer something about the system under study. Assuming model (4.1) is correct, any variation in the main effect can only come from random error in the observations. In fact,

$$\begin{aligned} \mathrm{var}\{ME(A)\} &= \frac{\sigma^2}{n/2} + \frac{\sigma^2}{n/2} \\ &= \frac{4\sigma^2}{n} \,, \end{aligned}$$

and assuming $r > 1$,

$$\hat{\sigma}^2 = \frac{1}{2^f(r-1)} \sum_{i=1}^{2^f} \sum_{j=1}^{r} (y_{ij} - \bar{y}_{i\cdot})^2 \,, \tag{4.2}$$

Table 4.2: Desilylation experiment: main effect contrast coefficients

| | Temperature | Time | Solvent | Reagent |
|---|---|---|---|---|
| Trt 1 | -0.125 | -0.125 | -0.125 | -0.125 |
| Trt 2 | 0.125 | -0.125 | -0.125 | -0.125 |
| Trt 3 | -0.125 | 0.125 | -0.125 | -0.125 |
| Trt 4 | 0.125 | 0.125 | -0.125 | -0.125 |
| Trt 5 | -0.125 | -0.125 | 0.125 | -0.125 |
| Trt 6 | 0.125 | -0.125 | 0.125 | -0.125 |
| Trt 7 | -0.125 | 0.125 | 0.125 | -0.125 |
| Trt 8 | 0.125 | 0.125 | 0.125 | -0.125 |
| Trt 9 | -0.125 | -0.125 | -0.125 | 0.125 |
| Trt 10 | 0.125 | -0.125 | -0.125 | 0.125 |
| Trt 11 | -0.125 | 0.125 | -0.125 | 0.125 |
| Trt 12 | 0.125 | 0.125 | -0.125 | 0.125 |
| Trt 13 | -0.125 | -0.125 | 0.125 | 0.125 |
| Trt 14 | 0.125 | -0.125 | 0.125 | 0.125 |
| Trt 15 | -0.125 | 0.125 | 0.125 | 0.125 |
| Trt 16 | 0.125 | 0.125 | 0.125 | 0.125 |

which is the residual mean square.

For Example 4.1, we can also calculate main effect estimates for the other three factors by defining appropriate contrasts in the treatments.

```
contrast.mat <- FrF2::FrF2(nruns = 16, nfactors = 4, randomize = F,
                           factor.names = c("temp", "time", "solvent", "reagent"))
fac.contrasts.emmc <- function(levs)
  dplyr::mutate_all(data.frame(contrast.mat), function(x) scale(as.numeric(as.character(x)), scal
main_effect_contrasts <- fac.contrasts.emmc()
rownames(main_effect_contrasts) <- paste("Trt", 1:16)
knitr::kable(main_effect_contrasts, caption = 'Desilylation experiment: main effect contrast coef
```

Estimates can be obtained by applying these coefficients to the observed treatment means.

```
t(as.matrix(main_effect_contrasts)) %*% yield
```

```
##           [,1]
## temp     8.120
## time     2.567
## solvent -2.218
## reagent  3.087
```

```
emmeans::contrast(desilylation.emm, 'fac.contrasts')
```

```
##  contrast estimate  SE df t.ratio p.value
```

```
##  temp         8.12 NaN  0     NaN     NaN
##  time         2.57 NaN  0     NaN     NaN
##  solvent     -2.22 NaN  0     NaN     NaN
##  reagent      3.09 NaN  0     NaN     NaN
```

Main effects are often displayed graphically, using **main effect plots** which simply plot the average response for each factor level, joined by a line. The larger the main effect, the larger the slope of the line (or the bigger the difference between the averages). Figure 4.1 presents the four main effect plots for Example 4.1.

```
## calculate the means
temp_bar <- aggregate(yield ~ temp, data = desilylation, FUN = mean)
time_bar <- aggregate(yield ~ time, data = desilylation, FUN = mean)
solvent_bar <- aggregate(yield ~ solvent, data = desilylation, FUN = mean)
reagent_bar <- aggregate(yield ~ reagent, data = desilylation, FUN = mean)

## convert factors to numeric
fac_to_num <- function(x) as.numeric(as.character(x))
temp_bar$temp <- fac_to_num(temp_bar$temp)
time_bar$time <- fac_to_num(time_bar$time)
solvent_bar$solvent <- fac_to_num(solvent_bar$solvent)
reagent_bar$reagent <- fac_to_num(reagent_bar$reagent)

## main effect plots
plotmin <- min(temp_bar$yield, time_bar$yield, solvent_bar$yield, reagent_bar$yield)
plotmax <- max(temp_bar$yield, time_bar$yield, solvent_bar$yield, reagent_bar$yield)
par(cex = 2)
layout(matrix(c(1, 2, 3, 4), nrow = 2, ncol = 2, byrow = TRUE), respect = T)
plot(temp_bar, pch = 16, type = "b", ylim = c(plotmin, plotmax))
plot(time_bar, pch = 16, type = "b", ylim = c(plotmin, plotmax))
plot(solvent_bar, pch = 16, type = "b", ylim = c(plotmin, plotmax))
plot(reagent_bar, pch = 16, type = "b", ylim = c(plotmin, plotmax))
```

### 4.1.2 Interactions

Another contrast that could be of interest in Example 4.1 has coefficients

$$\boldsymbol{c}^{\mathrm{T}} = (\mathbf{1}_4^{\mathrm{T}}, -\mathbf{1}_8^{\mathrm{T}}, \mathbf{1}_4^{\mathrm{T}})/8\,.$$

This contrast measures the difference between the average treatment effect from treatments 1-4, 13-16 and treatments 5-12. Checking back against Table 4.1, we see this is comparing those treatments where `solvent` and `reagent` are both set to their low (1-4) or high (13-16) level against those treatments where one of the two factors is set high and the other is set low (5-12).

Focusing on `reagent`, if the effect of this factor on the response was independent

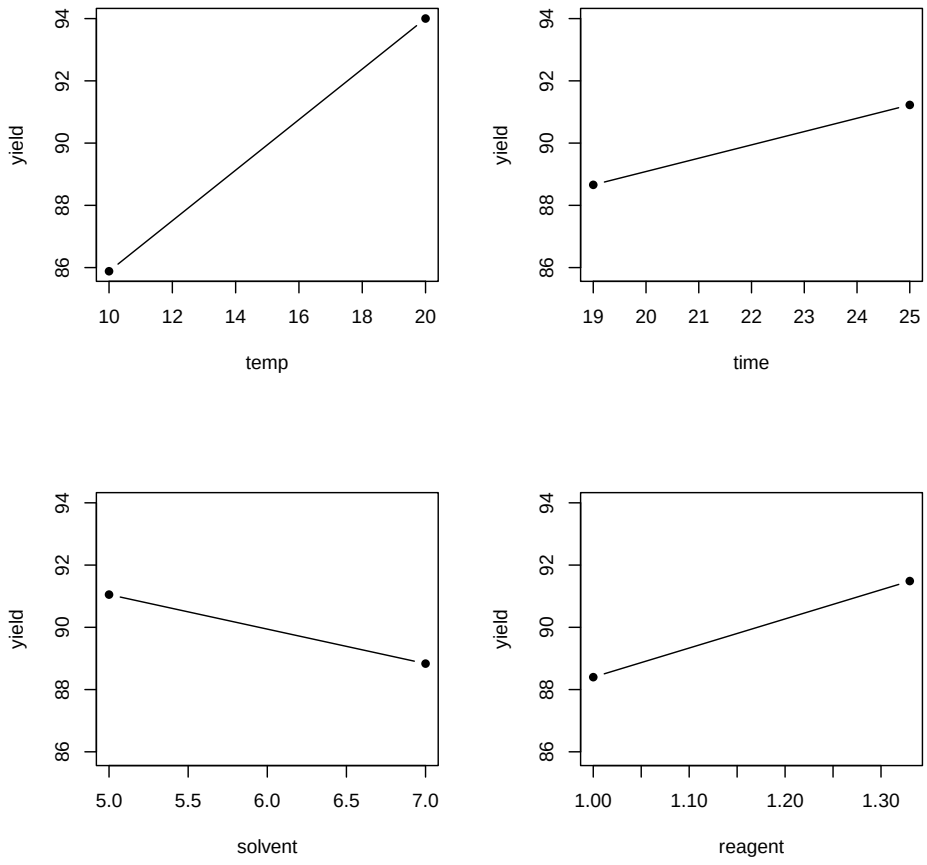



Figure 4.1: Desilylation experiment: main effect plots

of the level to which `solvent` has been set, you would expect this contrast to be zero - changing from the high to low level of `reagent` should affect the response in the same way, regardless of the setting of `solvent`. This argument can be reversed, focussing on the effect of `solvent`. Therefore, if this contrast is large, we say the two factors **interact**.

```
sol_reg_int.emmc <- function(levels)
  data.frame('reagent x solvent' = .125 * c(rep(1, 4), rep(-1, 8), rep(1, 4)))
emmeans::contrast(desilylation.emm, 'sol_reg_int')
```

```
##  contrast          estimate  SE df t.ratio p.value
##  reagent.x.solvent     0.49 NaN  0     NaN     NaN
```

For Example 4.1, this interaction contrast seems quite small, although of course without an estimate of the standard error we are still lacking a formal method to judge this.

It is somewhat more informative to consider the above interaction contrast as the average difference in two "sub-contrasts"

$$\boldsymbol{c}^{\mathrm{T}}\boldsymbol{\tau} = \frac{1}{2}\left\{\frac{1}{4}\left(\tau_{13}+\tau_{14}+\tau_{15}+\tau_{16}-\tau_5-\tau_6-\tau_7-\tau_8\right) - \frac{1}{4}\left(\tau_9+\tau_{10}+\tau_{11}+\tau_{12}-\tau_1-\tau_2-\tau_3-\tau_4\right)\right\},$$

The first component in the above expression is the effect of changing `reagent` from high to low **given `solvent` is set to it's high level**. The second component the effect of changing `reagent` from high to low **given `solvent` is set to it's low level**. This leads to our definition of a two-factor interaction.

**Definition 4.2.** The **two-factor interaction** between factors $A$ and $B$ is defined as the average difference in main effect of factor $A$ when computed at the high and low levels of factor $B$.

$$\begin{aligned}\mathrm{Int}(A,B) &= \frac{1}{2}\left\{\mathrm{ME}(A \mid B+) - \mathrm{ME}(A \mid B-)\right\} \\ &= \frac{1}{2}\left\{\mathrm{ME}(B \mid A+) - \mathrm{ME}(B \mid A-)\right\} \\ &= \frac{1}{2}\left\{\bar{y}(A+,B+) - \bar{y}(A-,B+) - \bar{y}(A+,B-) + \bar{y}(A-,B-)\right\},\end{aligned}$$

where $\bar{y}(A+,B-)$ is the average response when factor $A$ is set to its high level and factor $B$ is set to its low level, averaged across all combinations of levels of the other factors, and other averages are defined similarly. The conditional main effects of factor $A$ when factor $B$ is set to its high level is defined as

$$\mathrm{ME}(A \mid B+) = \bar{y}(A+,B+) - \bar{y}(A-,B+),$$

with similar definitions for other conditional main effects.

As the sum of the squared contrast coefficients is the same for two-factor interactions as for main effects, the variance of the contrast estimator is also the same.

$$\text{var}\left\{\text{Int}(A,B)\right\} = \frac{4\sigma^2}{n}\,.$$

For Example 4.1 we can calculate two-factor interactions for all $\binom{4}{2} = 6$ pairs of factors. The simplest way to calculate the contrast coefficients is as the elementwise, or Hadamard, product[2] of the unscaled main effect contrasts (before dividing by $n/2$).

```
fac.contrasts.int.emmc <- function(levs) {
  with(sqrt(8) * main_effect_contrasts, {
    data.frame('tem_x_tim' = temp * time,
               'tem_x_sol' = temp * solvent,
               'tem_x_rea' = temp * reagent,
               'tim_x_sol' = time * solvent,
               'tim_x_rea' = time * reagent,
               'sol_x_rea' = solvent * reagent)
  })
}
two_fi_contrasts <- fac.contrasts.int.emmc()
rownames(two_fi_contrasts) <- paste("Trt", 1:16)
knitr::kable(two_fi_contrasts, caption = 'Desilylation experiment: two-factor interaction contras
```

Estimates of the interaction contrasts can again by found by considering the equivalent contrasts in the observed treatment means.

```
t(as.matrix(two_fi_contrasts)) %*% yield
```

```
##             [,1]
## tem_x_tim -2.357
## tem_x_sol  2.358
## tem_x_rea -2.773
## tim_x_sol  0.440
## tim_x_rea -0.645
## sol_x_rea  0.490
```

```
emmeans::contrast(desilylation.emm, 'fac.contrasts.int')
```

```
##  contrast  estimate  SE df t.ratio p.value
##  tem_x_tim   -2.357 NaN  0     NaN     NaN
##  tem_x_sol    2.357 NaN  0     NaN     NaN
##  tem_x_rea   -2.772 NaN  0     NaN     NaN
```

[2]For two matrices $A$ and $B$ of the same dimension $m \times n$, the Hadamard product $A \odot B$ is a matrix of the same dimension with elements given by the elementwise product, $(A \odot B)_{ij} = A_{ij}B_{ij}$.

Table 4.3: Desilylation experiment: two-factor interaction contrast coefficients

| | tem_x_tim | tem_x_sol | tem_x_rea | tim_x_sol | tim_x_rea | sol_x_rea |
|---|---|---|---|---|---|---|
| Trt 1 | 0.125 | 0.125 | 0.125 | 0.125 | 0.125 | 0.125 |
| Trt 2 | -0.125 | -0.125 | -0.125 | 0.125 | 0.125 | 0.125 |
| Trt 3 | -0.125 | 0.125 | 0.125 | -0.125 | -0.125 | 0.125 |
| Trt 4 | 0.125 | -0.125 | -0.125 | -0.125 | -0.125 | 0.125 |
| Trt 5 | 0.125 | -0.125 | 0.125 | -0.125 | 0.125 | -0.125 |
| Trt 6 | -0.125 | 0.125 | -0.125 | -0.125 | 0.125 | -0.125 |
| Trt 7 | -0.125 | -0.125 | 0.125 | 0.125 | -0.125 | -0.125 |
| Trt 8 | 0.125 | 0.125 | -0.125 | 0.125 | -0.125 | -0.125 |
| Trt 9 | 0.125 | 0.125 | -0.125 | 0.125 | -0.125 | -0.125 |
| Trt 10 | -0.125 | -0.125 | 0.125 | 0.125 | -0.125 | -0.125 |
| Trt 11 | -0.125 | 0.125 | -0.125 | -0.125 | 0.125 | -0.125 |
| Trt 12 | 0.125 | -0.125 | 0.125 | -0.125 | 0.125 | -0.125 |
| Trt 13 | 0.125 | -0.125 | -0.125 | -0.125 | -0.125 | 0.125 |
| Trt 14 | -0.125 | 0.125 | 0.125 | -0.125 | -0.125 | 0.125 |
| Trt 15 | -0.125 | -0.125 | -0.125 | 0.125 | 0.125 | 0.125 |
| Trt 16 | 0.125 | 0.125 | 0.125 | 0.125 | 0.125 | 0.125 |

```
##  tim_x_sol    0.440 NaN  0     NaN     NaN
##  tim_x_rea   -0.645 NaN  0     NaN     NaN
##  sol_x_rea    0.490 NaN  0     NaN     NaN
```

As with main effects, interactions are often displayed graphically using **interaction** plots, plotting average responses for each pairwise combination of factors, joined by lines.

```
plotmin <- min(desilylation$yield)
plotmax <- max(desilylation$yield)
par(cex = 2)
layout(matrix(c(1, 2, 3, 4, 5, 6), nrow = 3, ncol = 2, byrow = TRUE), respect = T)

with(desilylation, {
  interaction.plot(temp, time, yield, type = "b", pch = 16, legend = F,
                   ylim = c(plotmin, plotmax))
  legend("bottomright", legend = c("Time = 19", "Time = 25"), lty = 2:1, cex = .75)
  interaction.plot(temp, solvent, yield, type = "b", pch = 16, legend = F,
                   ylim = c(plotmin, plotmax))
  legend("bottomright", legend = c("Solvent = 5", "Solvent = 7"), lty = 2:1, cex = .75)
  interaction.plot(temp, reagent, yield, type = "b", pch = 16, legend = F,
                   ylim = c(plotmin, plotmax))
  legend("bottomright", legend = c("Reagent = 1", "Reagent = 1.33"), lty = 2:1, cex =
  interaction.plot(time, solvent, yield, type = "b", pch = 16, legend = F,
                   ylim = c(plotmin, plotmax))
```

```
  legend("bottomright", legend = c("Solvent = 5", "Solvent = 7"), lty = 2:1, cex = .75)
  interaction.plot(time, reagent, yield, type = "b", pch = 16, legend = F,
                   ylim = c(plotmin, plotmax))
  legend("bottomright", legend = c("Reagent = 1", "Reagent = 1.33"), lty = 2:1, cex = .75)
  interaction.plot(solvent, reagent, yield, type = "b", pch = 16, legend = F,
                   ylim = c(plotmin, plotmax))
  legend("bottomright", legend = c("Reagent = 1", "Reagent = 1.33"), lty = 2:1, cex = .75)
})
```

Parallel lines in an interaction plot indicate no (or very small) interaction (`time` and `solvent`, `time` and `reagent`, `solvent` and `reagent`). The three interactions with `temp` all demonstrate much more robust behaviour at the high level; changing `time`, `solvent` or `reagent` makes little difference to the response at the high level of `temp`, and much less difference than at the low level of `temp`.

If a system displays important interactions, the main effects of factors involved in those interactions should no longer be interpreted. For example, it makes little sense to discuss the main effect of `temp` when is changes so much with the level of `reagent` (from strongly positive when `reagent` is low to quite small when `reagent` is high).

Higher order interactions can be defined similarly, as average differences in lower-order effects. For example, a three-factor interaction measures how a two-factor interaction changes with the levels of a third factor.

$$\begin{aligned}\text{Int}(A,B,C) &= \frac{1}{2}\left\{\text{Int}(A,B \mid C+) - \text{Int}(A,B \mid C-)\right\} \\ &= \frac{1}{2}\left\{\text{Int}(A,C \mid B+) - \text{Int}(A,C \mid B-)\right\} \\ &= \frac{1}{2}\left\{\text{Int}(B,C \mid A+) - \text{Int}(B,C \mid A-)\right\}\ ,\end{aligned}$$

where

$$\text{Int}(A,B \mid C+) = \frac{1}{2}\left\{\bar{y}(A+,B+,C+) - \bar{y}(A-,B+,C+) - \bar{y}(A+,B-,C+) + \bar{y}(A-,B-,C+)\right\}$$

is the interaction between factors $A$ and $B$ using only those treatments where factor $C$ is set to it's high level. Higher order interaction contrasts can again be constructed by (multiple) hadamard products of (unscaled) main effect contrasts.

**Definition 4.3.** A **factorial effect** is a main effect or interaction contrast defined on a factorial experiment. For a $2^f$ factorial experiment with $f$ factors, there are $2^f - 1$ factorial effects, ranging from main effects to the interaction between all $f$ factors.

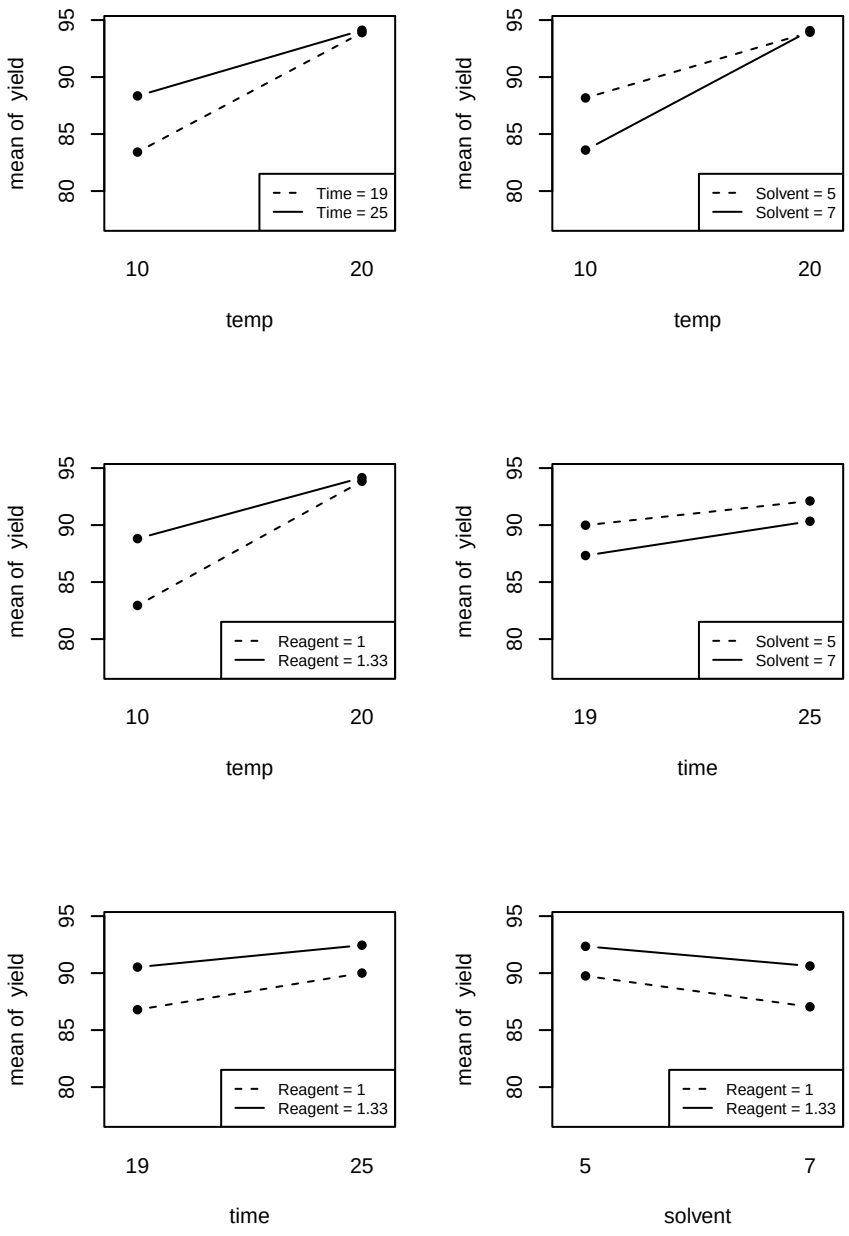



Figure 4.2: Desilylation experiment: two-factor interaction plots

For Example 4.1, we can now calculate all the factorial effects.

```
## hadamard products
unscaled_me_contrasts <- 8 * main_effect_contrasts
factorial_contrasts <- model.matrix(~.^4, unscaled_me_contrasts)[, -1] / 8

## all factorial effects - directly, as there is no treatment replication
t(factorial_contrasts) %*% yield
```

```
##                               [,1]
## temp                        8.1200
## time                        2.5675
## solvent                    -2.2175
## reagent                     3.0875
## temp:time                  -2.3575
## temp:solvent                2.3575
## temp:reagent               -2.7725
## time:solvent                0.4400
## time:reagent               -0.6450
## solvent:reagent             0.4900
## temp:time:solvent           0.2450
## temp:time:reagent           0.1950
## temp:solvent:reagent       -0.0300
## time:solvent:reagent       -0.2375
## temp:time:solvent:reagent   0.1925
```

```
## using emmeans
factorial_contrasts.emmc <- function(levs) data.frame(factorial_contrasts)
desilylation.effs <- emmeans::contrast(desilylation.emm, 'factorial_contrasts')
desilylation.effs
```

```
##  contrast                  estimate  SE df t.ratio p.value
##  temp                         8.120 NaN  0     NaN     NaN
##  time                         2.567 NaN  0     NaN     NaN
##  solvent                     -2.218 NaN  0     NaN     NaN
##  reagent                      3.087 NaN  0     NaN     NaN
##  temp.time                   -2.357 NaN  0     NaN     NaN
##  temp.solvent                 2.358 NaN  0     NaN     NaN
##  temp.reagent                -2.772 NaN  0     NaN     NaN
##  time.solvent                 0.440 NaN  0     NaN     NaN
##  time.reagent                -0.645 NaN  0     NaN     NaN
##  solvent.reagent              0.490 NaN  0     NaN     NaN
##  temp.time.solvent            0.245 NaN  0     NaN     NaN
##  temp.time.reagent            0.195 NaN  0     NaN     NaN
##  temp.solvent.reagent        -0.030 NaN  0     NaN     NaN
##  time.solvent.reagent        -0.238 NaN  0     NaN     NaN
##  temp.time.solvent.reagent    0.193 NaN  0     NaN     NaN
```

## 4.2 Three principles for factorial effects

Empirical study of many experiments (Box and Meyer, 1986; Li et al., 2006) have demonstrated that the following three principles **often** hold when analysing factorial experiments.

**Definition 4.4. Effect hierarchy**: lower-order factorial effects are more likely to be important than higher-order effects; factorial effects of the same order are equally likely to be important.

For example, we would anticipate more large main effects from the analysis of a factorial experiment than two-factor interactions.

**Definition 4.5. Effect sparsity**: the number of large factorial effects is likely to be small, relative to the total number under study.

This is sometimes called the **pareto principle**.

**Definition 4.6. Effect heredity**: interactions are more likely to be important if at least one parent factor also has a large main effect.

These three principles will provide us with some useful guidelines when analysing, and eventually constructing, factorial experiments.

## 4.3 Normal effect plots for unreplicated factorial designs

The lack of an estmate for $\sigma^2$ means alternatives to formal inference methods (e.g. hypothesis tests) must be found to assess the size of factorial effects. We will discuss a method that essentially treats the identification of large factorial effects as an outlier identification problem.

Let $\hat{\theta}_j$ be the $j$th estimated factorial effect, with $\hat{\theta}_j = \sum_{i=1}^{t} c_{ij}\bar{y}_{i\cdot}$ for $\boldsymbol{c}_j^{\mathrm{T}} = (c_{1j}, \ldots, c_{tj})$ a vector of factorial contrast coefficients (defining a main effect or interaction). Then the estimator follows a normal distribution

$$\hat{\theta}_j \sim N\left(\theta_j, \frac{4\sigma^2}{n}\right), \qquad j = 1, \ldots, 2^f - 1,$$

for $\theta_j$ the true, unknown, value of the factorial effect, $j = 1, \ldots, 2^f$. Further more, for $j, l = 1, \ldots 2^f - 1$; $j \neq l$,

$$\begin{aligned}\text{cov}(\hat{\theta}_j, \hat{\theta}_l) &= \text{cov}\left(\sum_{i=1}^{t} c_{ij}\bar{y}_{i\cdot}, \sum_{i=1}^{t} c_{il}\bar{y}_{i\cdot}\right) \\ &= \sum_{i=1}^{t} c_{ij}c_{il}\text{var}(\bar{y}_{i\cdot}) \\ &= \frac{\sigma^2}{r}\sum_{i=1}^{t} c_{ij}c_{il} \\ &= 0\,,\end{aligned}$$

as $\sum_{i=1}^{t} c_{ij}c_{il} = 0$ for $j \neq l$. That is, the factorial contrasts are independent as the contrast coefficient vectors are orthogonal.

Hence, under the null hypothesis $H_0 : \theta_1 = \cdots = \theta_{2^f-1} = 0$ (all factorial effects are zero), the $\hat{\theta}_j$ form a sample from **independent** normally distributed random variables from the distribution

$$\hat{\theta}_j \sim N\left(0, \frac{4\sigma^2}{n}\right)\,, \qquad j = 1, \ldots, 2^f - 1\,. \tag{4.3}$$

To assess evidence against $H_0$, we can plot the ordered estimates of the factorial effects against the ordered quantiles of a standard normal distribution. Under $H_0$, the points in this plot should lie on a straightline (the slope of the line will depend on the unknown $\sigma^2$). We anticipate that the majority of the effects will be small (*effect sparsity*), and hence any large effects that lie away from the line are unlikely to come from distribution (4.3) and may be significantly different from zero. We have essentially turned the problem into an outlier identification problem.

For Example 4.1, we can easily produce this plot in R. Table 4.4 gives the ordered factorial effects, which are then plotted against standard normal quantiles in Figure 4.3.

```
effs <- dplyr::arrange(transform(desilylation.effs)[,1:2], dplyr::desc(estimate))
knitr::kable(effs, caption = "Desilylation experiment: sorted estimated factorial effects")
qqnorm(effs[ ,2], ylab = "Factorial effects", main = "") # note that qqnorm/qqline/qqplot don't r
qqline(effs[ ,2])
```

In fact, it is more usual to use a **half-normal** plot to assess the size of factorial effects, where we plot the sorted absolute values of the estimated effects against the quantiles of a half-normal distribution[3]

[3]The absolute value of a normally distributed random variable follows a half-normal distribution.

Table 4.4: Desilylation experiment: sorted estimated factorial effects

| contrast | estimate |
|---|---|
| temp | 8.1200 |
| reagent | 3.0875 |
| time | 2.5675 |
| temp.solvent | 2.3575 |
| solvent.reagent | 0.4900 |
| time.solvent | 0.4400 |
| temp.time.solvent | 0.2450 |
| temp.time.reagent | 0.1950 |
| temp.time.solvent.reagent | 0.1925 |
| temp.solvent.reagent | -0.0300 |
| time.solvent.reagent | -0.2375 |
| time.reagent | -0.6450 |
| solvent | -2.2175 |
| temp.time | -2.3575 |
| temp.reagent | -2.7725 |

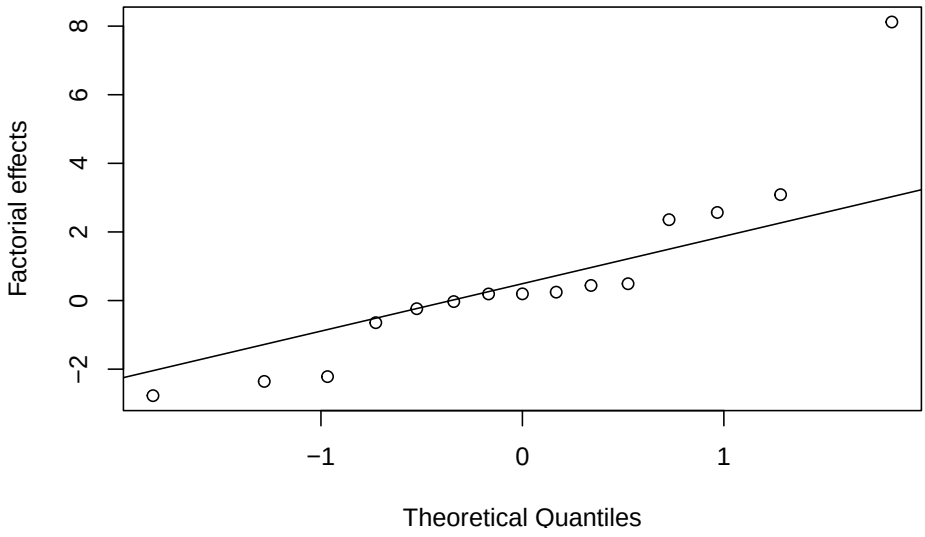



Figure 4.3: Desilylation experiment: normal effects plot

```
p <- .5 + .5 * (1:16 - .5) /16 # probabilities we will plot against
qqplot(x = qnorm(p), y = abs(effs[,2]), ylab = "Absolute factorial effects",
       xlab = "Half-normal quantiles")
```

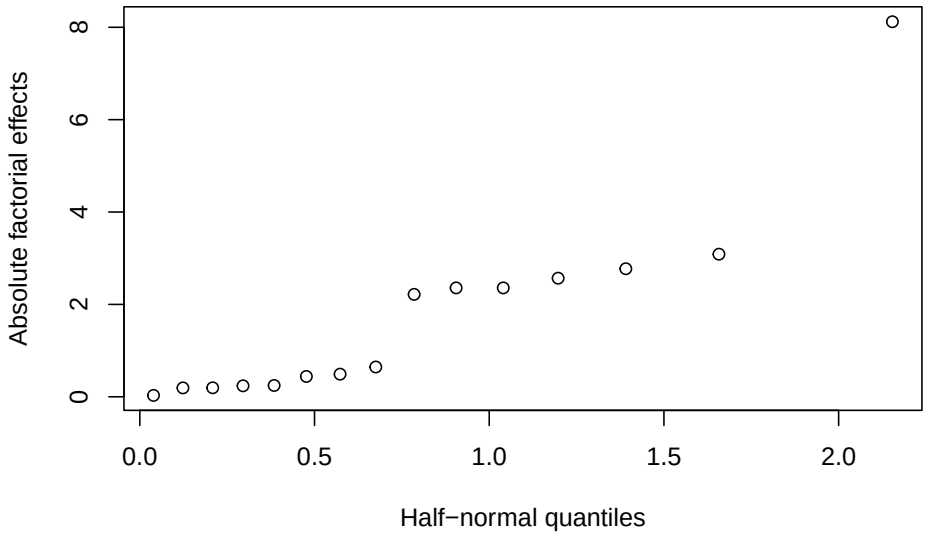



Figure 4.4: Desilylation experiment: half-normal effects plot

The advantage of a half-normal plot such as Figure 4.4 is that we only need to look at effects appearing in the top right corner (significant effects will always appear "above" a hypothetical straight line) and we do not need to worry about comparing large positive and negative values. For these reason, they are usually preferred over normal plots.

For the desilylation experiment, we can see the effects fall into three groups: one effect standing well away from the line, and almost certainly significant (`temp`, from Table **??**), then a group of six effects (`reagent`, `time`, `temp.solvent`, `solvent`, `temp.time`, `temp.reagent`) which may be significant, and then a group of 8 much smaller effects.

### 4.3.1 Lenth's method for approximate hypothesis testing

The assessment of normal or half-normal effect plots can be quite subjective. Lenth (1989) introduced a simple method for conducting more formal hypothesis testing in unreplicated factorial experiments.

Lenth's method uses a **pseudo standard error** (PSE):

$$\text{PSE} = 1.5 \times \text{median}_{|\hat{\theta}_i|<2.5s_0} |\hat{\theta}_i| \,,$$

where $s_0 = 1.5 \times \text{median}|\hat{\theta}_i|$ is a *consistent*[4] estimator of the standard deviation of the $\hat{\theta}_i$ under $H_0 : \theta_1 = \cdots = \theta_{2^f-1} = 0$. The PSE trims approximately 1%[5] of the $\hat{\theta}_i$ to produce a **robust** estimator of the standard deviation, in the sense that it is not influenced by large $\hat{\theta}_i$ belonging to important effects.

For Example 4.1, we can construct the PSE as follows.

```
s0 <- 1.5 * median(abs(effs[, 2]))
trimmed <- abs(effs[, 2]) < 2.5 * s0
pse <- 1.5 * median(abs(effs[trimmed, 2]))
pse
```

```
## [1] 0.66
```

The PSE can be used to construct test statistics

$$t_{\text{PSE},i} = \frac{\hat{\theta}_i}{\text{PSE}},$$

which mimic the usual $t$-statistics used when an estimate of $\sigma^2$ is available. These quantities can be compared to reference distribution which was tabulated by Lenth (1989) and simulated in `R` using the `unrepx` package.

```
eff_est <- effs[, 2]
names(eff_est) <- effs[, 1]
lenth_tests <- unrepx::eff.test(eff_est, method = "Lenth")
knitr::kable(lenth_tests, caption = "Desilylation experiment: hypothesis tests using Le
```

The function `eff.test` calculates unadjusted p-values (`p.value`) and simultaneous p-values (`simult.pval`) adjusted to account for multiple testing. Using the latter, from Table 4.5 we see that the main effects of `temp` and `reagent` are significant at the experiment-wise 5% level and, obeying **effect heredity**, their interaction (the p-value is borderline, and hovers around 0.05 depending on simulation error).

The package `unrepx` also provides the function `hnplot` to display these results graphically by adding a reference line to a half-normal plot; see Figure **??**. The `ME` and `SME` lines indicate the absolute size of effects that would be required to reject $H_0 : \theta_i = 0$ at an individual or experimentwise $100\alpha\%$ level, respectively.

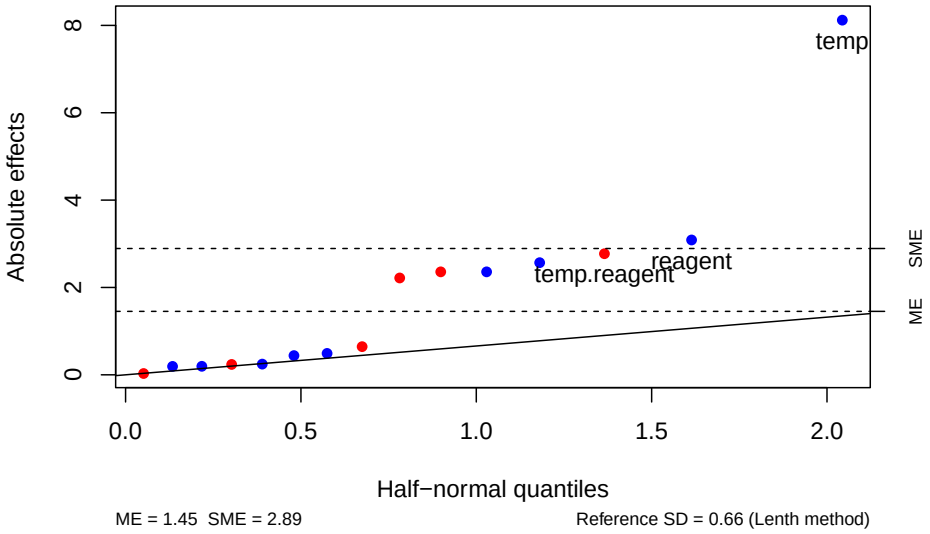
```
unrepx::hnplot(eff_est, method = "Lenth", horiz = F, ID = 2.7, alpha = 0.05)
```

[4] Essentially, $s_0$ tends in probability to $\sigma$ as the number of factorial effects tends to infinity.

[5] Under $H_0$, the $\hat{\theta}_i$ come from a mean-zero normal distribution, and about 1% of deviates fall outside $\pm 2.57\sigma^2$.

Table 4.5: Desilylation experiment: hypothesis tests using Lenth's method.

| | effect | Lenth_PSE | t.ratio | p.value | simult.pval |
|---|---|---|---|---|---|
| temp | 8.1200 | 0.66 | 12.303 | 0.0002 | 0.0022 |
| reagent | 3.0875 | 0.66 | 4.678 | 0.0044 | 0.0397 |
| temp.reagent | -2.7725 | 0.66 | -4.201 | 0.0063 | 0.0581 |
| time | 2.5675 | 0.66 | 3.890 | 0.0078 | 0.0690 |
| temp.solvent | 2.3575 | 0.66 | 3.572 | 0.0107 | 0.0956 |
| temp.time | -2.3575 | 0.66 | -3.572 | 0.0107 | 0.0956 |
| solvent | -2.2175 | 0.66 | -3.360 | 0.0140 | 0.1252 |
| time.reagent | -0.6450 | 0.66 | -0.977 | 0.3054 | 0.9951 |
| solvent.reagent | 0.4900 | 0.66 | 0.742 | 0.4316 | 1.0000 |
| time.solvent | 0.4400 | 0.66 | 0.667 | 0.5406 | 1.0000 |
| temp.time.solvent | 0.2450 | 0.66 | 0.371 | 0.7353 | 1.0000 |
| time.solvent.reagent | -0.2375 | 0.66 | -0.360 | 0.7430 | 1.0000 |
| temp.time.reagent | 0.1950 | 0.66 | 0.295 | 0.7876 | 1.0000 |
| temp.time.solvent.reagent | 0.1925 | 0.66 | 0.292 | 0.7896 | 1.0000 |
| temp.solvent.reagent | -0.0300 | 0.66 | -0.045 | 0.9666 | 1.0000 |




Informally, factorial effects with estimates greater than `SME` are thought highly likely to be significant, and effects between `ME` and `SME` are considered somewhat likely to be significant (and still worthy of further investigation if the budget allows).

## 4.4 Regression modelling for factorial experiments

We have identified $d = 2^f - 1$ factorial effects that we wish to estimate from our experiment. As $d < t = 2^f$, we can estimate these factorial effects using a full-rank linear regression model.

Let $t \times d$ matrix $C$ hold each factorial contrast as a column. Then

$$\hat{\boldsymbol{\theta}} = C^{\mathrm{T}} \bar{\boldsymbol{y}} \,,$$

with $\hat{\boldsymbol{\theta}}^{\mathrm{T}} = (\hat{\theta}_1, \ldots, \hat{\theta}_d)$ being the vector of estimated factorial effects and $\bar{\boldsymbol{y}}^{\mathrm{T}} = (\bar{y}_{1\cdot}, \ldots, \bar{y}_{t\cdot})$ being the vector of treatment means.

We can define an $n \times d$ expanded contrast matrix as $\tilde{C} = C \otimes \mathbf{1}_r$, where each row of $\tilde{C}$ gives the contrast coefficients for each run of the experiment. Then,

$$\hat{\boldsymbol{\theta}} = \frac{1}{r} \tilde{C}^{\mathrm{T}} \boldsymbol{y} \,.$$

To illustrate, we will imagine a hypothetical version of Example 4.1 where each treatment was repeated three times (with $y_{i1} = y_{i2} = y_{i3}$).

```
y <- kronecker(desilylation$yield, rep(1, 3)) # hypothetical response vector
C <- factorial_contrasts
Ctilde <- kronecker(C, rep(1, 3))
t(Ctilde) %*% y / 3 # to check
```

```
##          [,1]
##  [1,]  8.1200
##  [2,]  2.5675
##  [3,] -2.2175
##  [4,]  3.0875
##  [5,] -2.3575
##  [6,]  2.3575
##  [7,] -2.7725
##  [8,]  0.4400
##  [9,] -0.6450
## [10,]  0.4900
## [11,]  0.2450
## [12,]  0.1950
## [13,] -0.0300
## [14,] -0.2375
## [15,]  0.1925
```

If we define a model matrix $X = \frac{2^f}{2} \tilde{C}$, then $X$ is a $n \times d$ matrix with entries $\pm 1$ and columns equal to unscaled factorial contrasts. Then

$$\left(X^{\mathrm{T}}X\right)^{-1}X^{\mathrm{T}}\boldsymbol{y} = \frac{1}{n} \times \frac{2^f}{2}\tilde{C}^{\mathrm{T}}\boldsymbol{y} \tag{4.4}$$

$$= \frac{1}{2r}\tilde{C}^{\mathrm{T}}\boldsymbol{y} \tag{4.5}$$

$$= \frac{1}{2}\hat{\boldsymbol{\theta}}\,. \tag{4.6}$$

$$\tag{4.7}$$

The left-hand side of equation (4.4) is the least squares estimator $\hat{\boldsymbol{\beta}}$ from the model

$$\boldsymbol{y} = \mathbf{1}_n\beta_0 + X\boldsymbol{\beta} + \boldsymbol{\varepsilon}\,,$$

where $\boldsymbol{y}$ is the response vector and $\boldsymbol{\varepsilon}$ the error vector from unit-treatment model (4.1). We have simply re-expressed the mean response as $\mu + \tau_i = \beta_0 + \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\beta}$, where $d$-vector $\boldsymbol{x}_i$ holds the unscaled contrast coefficients for the main effects and interactions.

We can illustrate these connections for Example 4.1.

```
X <- 8 * C
Xt <- t(X)
XtX <- Xt %*% X
2 * solve(XtX) %*% t(X) %*% desilylation$yield
```

```
##                              [,1]
## temp                       8.1200
## time                       2.5675
## solvent                   -2.2175
## reagent                    3.0875
## temp:time                 -2.3575
## temp:solvent               2.3575
## temp:reagent              -2.7725
## time:solvent               0.4400
## time:reagent              -0.6450
## solvent:reagent            0.4900
## temp:time:solvent          0.2450
## temp:time:reagent          0.1950
## temp:solvent:reagent      -0.0300
## time:solvent:reagent      -0.2375
## temp:time:solvent:reagent  0.1925
```

The more usual way to think about this modelling approach is as a regression model with $f$ (quantitative[6]) variables, labelled $x_1, \ldots, x_{2^f-1}$, scaled to lie in the

[6]When qualitative factors only have two levels, each regression term only has 1 degree of freedom, and so there is practically little difference from a quantitative variable.

Table 4.6: Desilylation example: factorial effects calculated using a regression model.

| | x |
|---|---|
| temp | 8.1200 |
| time | 2.5675 |
| solvent | -2.2175 |
| reagent | 3.0875 |
| temp:time | -2.3575 |
| temp:solvent | 2.3575 |
| temp:reagent | -2.7725 |
| time:solvent | 0.4400 |
| time:reagent | -0.6450 |
| solvent:reagent | 0.4900 |
| temp:time:solvent | 0.2450 |
| temp:time:reagent | 0.1950 |
| temp:solvent:reagent | -0.0300 |
| time:solvent:reagent | -0.2375 |
| temp:time:solvent:reagent | 0.1925 |

interval $[-1, 1]$ (in fact, they just take values $\pm 1$). We can then fit a regression model in these variables, and include products of these variables to represent interactions. We usually also include the intercept term. For Example 4.1:

```
desilylation.df <- dplyr::mutate(desilylation,
                                 across(.cols = temp:reagent,
                                        ~ as.numeric(as.character(.x))))
desilylation.df <- dplyr::select(desilylation.df, -c(trt))
desilylation.df <- dplyr::mutate(desilylation.df, across(.cols = temp:reagent,
                                                         ~ scales::rescale(.x, to = c(-1, 1))))
desilylation_reg.lm <- lm(yield ~ (.) ^ 4, data = desilylation.df)
knitr::kable(2 * coef(desilylation_reg.lm)[-1], caption = "Desilylation example: factor
```

A regression modelling approach is usually more straightforward to apply than defining contrasts in the unit-treatment model, and makes clearer the connection between interaction contrasts and products of main effect contrasts (automatically defined in a regression model). It also enables us to make use of the `effects` package in `R` to quickly produce main effect and interaction plots.

```
temp_x_time <- effects::Effect(c("temp", "time"), desilylation_reg.lm, xlevels = list(t
plot(temp_x_time, main = "", rug = F, x.var = "temp", ylim = c(80, 100))
```

### 4.4.1 ANOVA for factorial experiments

The basic ANOVA table has the following form.

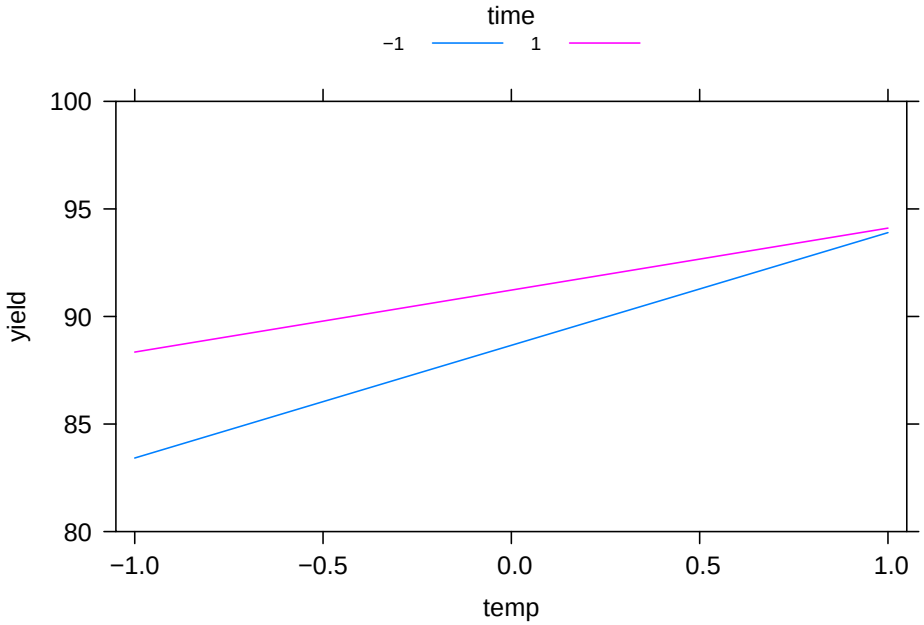



Figure 4.5: Desilylation experiment: interaction plot generated using the `effects` package.

Table 4.7: The ANOVA table for a full factorial experiment

| Source | Degress of Freedom | (Sequential) Sum of Squares | Mean Square |
|---|---|---|---|
| Regression | $2^f - 1$ | $\sum_{j=1}^{2^f-1} n\hat{\beta}_j^2 - n\bar{y}^2$ | Reg SS$/(2^f - 1)$ |
| Residual | $2^f(r-1)$ | $(\boldsymbol{Y} - X\hat{\boldsymbol{\beta}})^{\mathrm{T}}(\boldsymbol{Y} - X\hat{\boldsymbol{\beta}})$ | RSS$/(2^f(r-1))$ |
| Total | $2^f r - 1$ | $\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y} - n\bar{Y}^2$ | |

The regression sum of squares for a factorial experiment has a very simple form. If we include an intercept column in $X$, from Section 1.5.1,

$$\begin{aligned} \text{Regression SS} &= \text{RSS(null)} - \text{RSS} \\ &= \hat{\boldsymbol{\beta}}^{\mathrm{T}} X^{\mathrm{T}} X \hat{\boldsymbol{\beta}} - n\bar{y}^2 \\ &= \sum_{j=1}^{2^f-1} n\hat{\beta}_j^2 - n\bar{y}^2 \,, \end{aligned}$$

Table 4.8: Desilylation experiment: regression sums of squares for each factorial effect calculated directly.

| | Sum Sq. |
|---|---|
| Regression | 427.2837 |
| temp | 263.7376 |
| time | 26.3682 |
| solvent | 19.6692 |
| reagent | 38.1306 |
| temp:time | 22.2312 |
| temp:solvent | 22.2312 |
| temp:reagent | 30.7470 |
| time:solvent | 0.7744 |
| time:reagent | 1.6641 |
| solvent:reagent | 0.9604 |
| temp:time:solvent | 0.2401 |
| temp:time:reagent | 0.1521 |
| temp:solvent:reagent | 0.0036 |
| time:solvent:reagent | 0.2256 |
| temp:time:solvent:reagent | 0.1482 |

as $X^{\mathrm{T}}X = nI_{2^f}$. Hence, the $j$th factorial effect contributes $n\hat{\beta}_j^2$ to the regression sum if squares, and this quantity can be used to construct a test statistic **if** $r > 1$ and hence an estimate of $\sigma^2$ is available. For Example 4.1, the regression sum of squares and ANOVA table are given in Tables 4.8 and 4.9.

```
ss <- nrow(desilylation) * coef(desilylation_reg.lm)^2
regss <- sum(ss) - nrow(desilylation) * mean(desilylation$yield)^2
names(regss) <- "Regression"
knitr::kable(c(regss, ss[-1]), col.names = "Sum Sq.", caption = "Desilylation experimer
```

```
knitr::kable(anova(desilylation_reg.lm)[, 1:2], caption = "Desilylation experiment: ANO
```

```
## Warning in anova.lm(desilylation_reg.lm): ANOVA F-tests on an essentially
## perfect fit are unreliable
```

## 4.5 Exercises

1. A reactor experiment that was presented by Box, Hunter and Hunter (2005, pp259-261) that used a full factorial design for $m = 5$ factors, each at two levels, to investigate the effect of *feed rate* (litres/min), *catalyst* (%), *agitation rate* (rpm), *temperature* (C) and *concentration* (%) on the *percentage reacted*. The levels of the experimental factors will be coded as $-1$ for low level, and 1 for high level. Table 4.10 presents the true factor

Table 4.9: Desilylation experiment: ANOVA table from 'anova' function.

| | Df | Sum Sq |
|---|---|---|
| temp | 1 | 263.7376 |
| time | 1 | 26.3682 |
| solvent | 1 | 19.6692 |
| reagent | 1 | 38.1306 |
| temp:time | 1 | 22.2312 |
| temp:solvent | 1 | 22.2312 |
| temp:reagent | 1 | 30.7470 |
| time:solvent | 1 | 0.7744 |
| time:reagent | 1 | 1.6641 |
| solvent:reagent | 1 | 0.9604 |
| temp:time:solvent | 1 | 0.2401 |
| temp:time:reagent | 1 | 0.1521 |
| temp:solvent:reagent | 1 | 0.0036 |
| time:solvent:reagent | 1 | 0.2256 |
| temp:time:solvent:reagent | 1 | 0.1482 |
| Residuals | 0 | 0.0000 |

settings corresponding to these coded values.

Table 4.10: Factor levels for the full factorial reactor experiment

| Factor | Low level (−1) | High level (1) |
|---|---|---|
| Feed Rate (litres/min) | 10 | 15 |
| Catalyst (%) | 1 | 2 |
| Agitation Rate (rpm) | 100 | 120 |
| Temperature (C) | 140 | 180 |
| Concentration (%) | 3 | 6 |

The data from this experiment is given in Table 4.11.

```
reactor.frf2 <- FrF2::FrF2(nruns = 32, nfactors = 5, randomize = F,
                           factor.names = c("FR", "Cat", "AR", "Temp", "Conc"))
y <- c(61, 53, 63, 61, 53, 56, 54, 61, 69, 61, 94, 93, 66, 60, 95, 98, 56, 63,
       70, 65, 59, 55, 67, 65, 44, 45, 78, 77, 49, 42, 81, 82)
reactor <- data.frame(reactor.frf2, pre.react = y)
knitr::kable(reactor, caption = "Reactor experiment.")
```

a. Estimate all the factorial effects from this experiment, and use a half-normal plot and Lenth's method to decide which are significantly different from zero.

b. Use the `effects` package to produce main effect and/or interaction plots for each significant factorial effect from part a.

c. Now fit a regression model that only includes terms corresponding to main effects and two-factor interactions. How many degrees of freedom does this model use? What does this mean for the estimation of $\sigma^2$? How does the estimate of $\sigma^2$ from this model relate to your analysis in part a?

Solution

a. We will estimate the factorial effects as twice the corresponding regression

Table 4.11: Reactor experiment.

| FR | Cat | AR | Temp | Conc | pre.react |
|---|---|---|---|---|---|
| -1 | -1 | -1 | -1 | -1 | 61 |
| 1 | -1 | -1 | -1 | -1 | 53 |
| -1 | 1 | -1 | -1 | -1 | 63 |
| 1 | 1 | -1 | -1 | -1 | 61 |
| -1 | -1 | 1 | -1 | -1 | 53 |
| 1 | -1 | 1 | -1 | -1 | 56 |
| -1 | 1 | 1 | -1 | -1 | 54 |
| 1 | 1 | 1 | -1 | -1 | 61 |
| -1 | -1 | -1 | 1 | -1 | 69 |
| 1 | -1 | -1 | 1 | -1 | 61 |
| -1 | 1 | -1 | 1 | -1 | 94 |
| 1 | 1 | -1 | 1 | -1 | 93 |
| -1 | -1 | 1 | 1 | -1 | 66 |
| 1 | -1 | 1 | 1 | -1 | 60 |
| -1 | 1 | 1 | 1 | -1 | 95 |
| 1 | 1 | 1 | 1 | -1 | 98 |
| -1 | -1 | -1 | -1 | 1 | 56 |
| 1 | -1 | -1 | -1 | 1 | 63 |
| -1 | 1 | -1 | -1 | 1 | 70 |
| 1 | 1 | -1 | -1 | 1 | 65 |
| -1 | -1 | 1 | -1 | 1 | 59 |
| 1 | -1 | 1 | -1 | 1 | 55 |
| -1 | 1 | 1 | -1 | 1 | 67 |
| 1 | 1 | 1 | -1 | 1 | 65 |
| -1 | -1 | -1 | 1 | 1 | 44 |
| 1 | -1 | -1 | 1 | 1 | 45 |
| -1 | 1 | -1 | 1 | 1 | 78 |
| 1 | 1 | -1 | 1 | 1 | 77 |
| -1 | -1 | 1 | 1 | 1 | 49 |
| 1 | -1 | 1 | 1 | 1 | 42 |
| -1 | 1 | 1 | 1 | 1 | 81 |
| 1 | 1 | 1 | 1 | 1 | 82 |

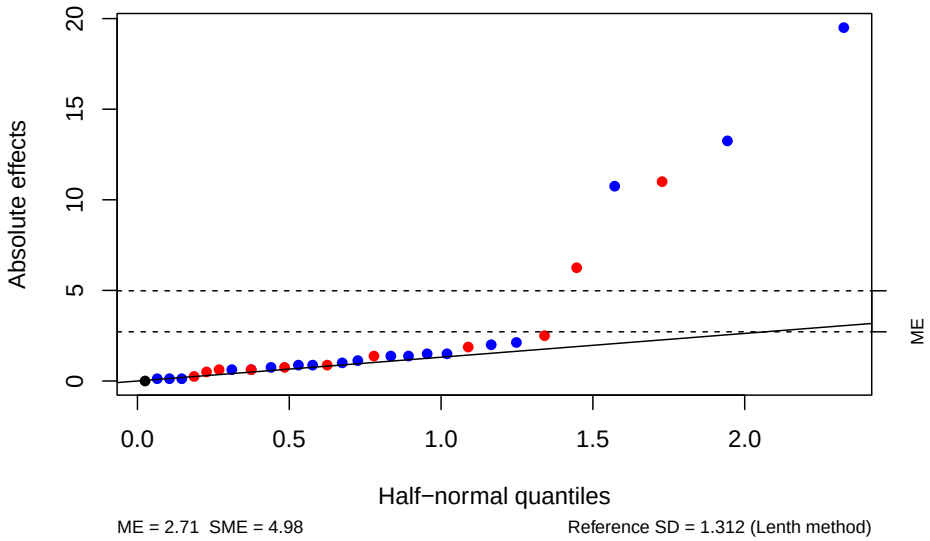
Table 4.12: Reactor experiment: estimated factorial effects.

| | x |
|---|---|
| FR | -1.375 |
| Cat | 19.500 |
| AR | -0.625 |
| Temp | 10.750 |
| Conc | -6.250 |
| FR:Cat | 1.375 |
| FR:AR | 0.750 |
| FR:Temp | -0.875 |
| FR:Conc | 0.125 |
| Cat:AR | 0.875 |
| Cat:Temp | 13.250 |
| Cat:Conc | 2.000 |
| AR:Temp | 2.125 |
| AR:Conc | 0.875 |
| Temp:Conc | -11.000 |
| FR:Cat:AR | 1.500 |
| FR:Cat:Temp | 1.375 |
| FR:Cat:Conc | -1.875 |
| FR:AR:Temp | -0.750 |
| FR:AR:Conc | -2.500 |
| FR:Temp:Conc | 0.625 |
| Cat:AR:Temp | 1.125 |
| Cat:AR:Conc | 0.125 |
| Cat:Temp:Conc | -0.250 |
| AR:Temp:Conc | 0.125 |
| FR:Cat:AR:Temp | 0.000 |
| FR:Cat:AR:Conc | 1.500 |
| FR:Cat:Temp:Conc | 0.625 |
| FR:AR:Temp:Conc | 1.000 |
| Cat:AR:Temp:Conc | -0.625 |
| FR:Cat:AR:Temp:Conc | -0.500 |

Table 4.13: Reactor experiment: factorial effects significantly different from zero via Lenth's method.

| | x |
|---|---|
| Cat | 19.50 |
| Temp | 10.75 |
| Conc | -6.25 |
| Cat:Temp | 13.25 |
| Temp:Conc | -11.00 |

```
unrepx::hnplot(fac.effects, horiz = F, method = "Lenth", alpha = 0.05)
```

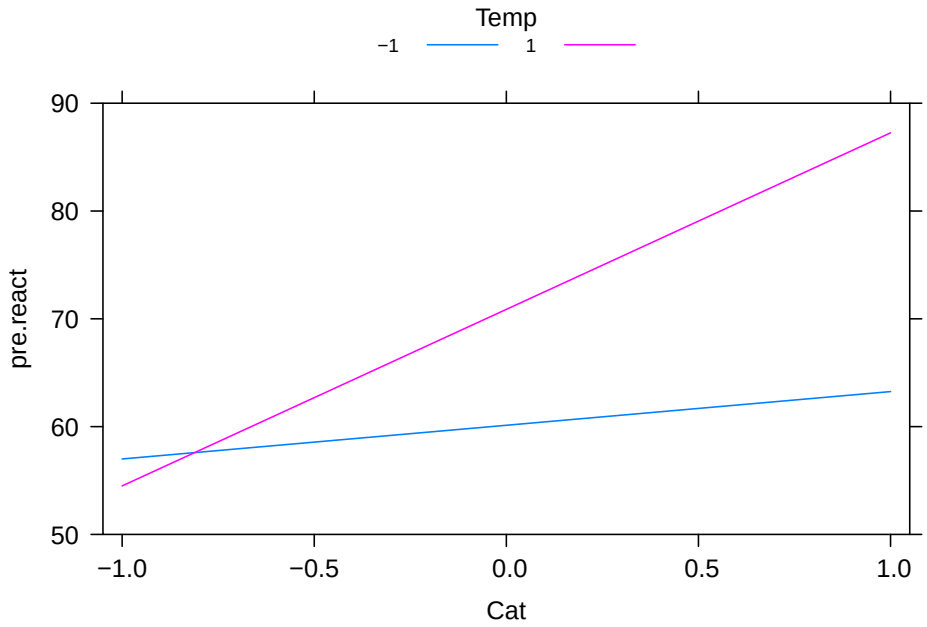


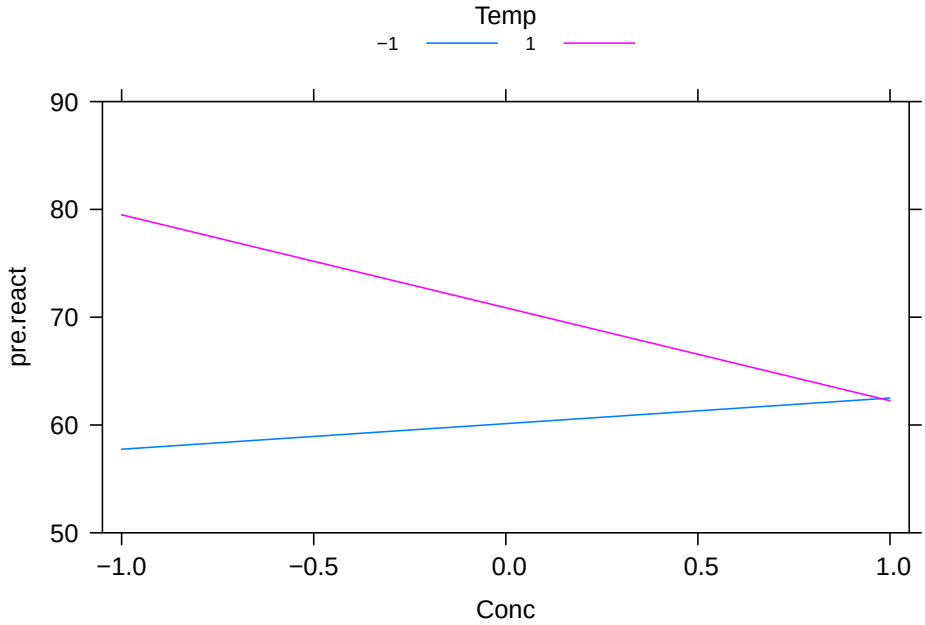

We see that PSE = 1.3125, giving individual and simultaneous margins of error of 2.6993 and 5.1858, respectively (where the latter is adjusted for multiple testing). There is a very clear distinction between the five effects which are largest in absolute value and the other factorial effects, which form a very clear line. The five of the largest effects are given in Table 4.13, are all greater than both margins of error and can be declared as significant.

```
knitr::kable(fac.effects[abs(fac.effects) > unrepx::ME(fac.effects,
                                                       method = "Lenth")[2]],
             caption = "Reactor experiment: factorial effects significantly different
```

b. We will produce plots for the interactions between Catalyst and Temperature and Temperature and Concentration. We will not produce main effect plots for Catalyst and Temperature, as these are involved in the large interactions.

```
Cat_x_Temp <- effects::Effect(c("Cat", "Temp"), reactor.lm,
                              xlevels = list(Cat = c(-1, 1), Temp = c(-1, 1)),
                              se = F)
Temp_x_Conc <- effects::Effect(c("Temp", "Conc"), reactor.lm,
                               xlevels = list(Conc = c(-1, 1), Temp = c(-1, 1)),
                               se = F)
plot(Cat_x_Temp, style = "stacked", main = "", rug = F, x.var = "Cat",
     ylim = c(50, 90))
plot(Temp_x_Conc, style = "stacked", main = "", rug = F, x.var = "Conc",
     ylim = c(50, 90))
```






Notice that changing the level of Temperature changes substantial the effect of both Catalyst and Concentration on the response; in particular, the effect of Concentration changes sign depending on the level of Temperature.

c. We start by fiting the reduced regression model.

```
reactor2.lm <- lm(pre.react ~ (.) ^ 2, data = reactor)
summary(reactor2.lm)
```

```
##
## Call:
## lm(formula = pre.react ~ (.)^2, data = reactor)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -5.00  -1.62  -0.25   1.75   4.50
##
## Coefficients:
##             Estimate Std. Error t value        Pr(>|t|)
## (Intercept)  65.5000     0.5660  115.73         < 2e-16 ***
## FR           -0.6875     0.5660   -1.21           0.242
## Cat           9.7500     0.5660   17.23 0.0000000000094 ***
## AR           -0.3125     0.5660   -0.55           0.588
## Temp          5.3750     0.5660    9.50 0.0000000560392 ***
## Conc         -3.1250     0.5660   -5.52 0.0000464538196 ***
## FR:Cat        0.6875     0.5660    1.21           0.242
## FR:AR         0.3750     0.5660    0.66           0.517
```

```
## FR:Temp      -0.4375     0.5660   -0.77           0.451
## FR:Conc       0.0625     0.5660    0.11           0.913
## Cat:AR        0.4375     0.5660    0.77           0.451
## Cat:Temp      6.6250     0.5660   11.71 0.0000000029456 ***
## Cat:Conc      1.0000     0.5660    1.77           0.096 .
## AR:Temp       1.0625     0.5660    1.88           0.079 .
## AR:Conc       0.4375     0.5660    0.77           0.451
## Temp:Conc    -5.5000     0.5660   -9.72 0.0000000408373 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.2 on 16 degrees of freedom
## Multiple R-squared:  0.976,  Adjusted R-squared:  0.954
## F-statistic: 44.1 on 15 and 16 DF,  p-value: 0.000000000376
```

This model includes regression parameters corresponding to $5 + \binom{5}{2} = 15$ factorial effects, plus the intercept, and hence uses 16 degrees of freedom. The remaining 16 degrees of freedom, which were previously used to estimate three-factor and higher interactions, is now used to estimate $\sigma^2$, the background variation.

The residual mean square in the reduced model, used to estimate $\sigma^2$, is the sum of the sums of squares for the higher-order interactions in the original model, divided by 16 (the remaining degrees of freedom).

```
sum(anova(reactor.lm)[16:31, 3]) / 16
```

```
## [1] 10.25
```

```
summary(reactor2.lm)$sigma^2
```

```
## [1] 10.25
```

This "pooling" of higher-order effects to estimate $\sigma^2$ maybe a reasonable strategy here, given that the high-order interactions are all small, but could be biased if one or more interactions were large.

2. (Adapted from Morris, 2011) Consider an unreplicated ($r = 1$) $2^6$ factorial experiment. The total sums of squares,

$$\text{Total SS} = \sum_{i=1}^{n} (y_i - \bar{y})^2 \,,$$

has value 2856. Using Lenth's method, an informal analysis of the data suggests that there are only three important factorial effects, with least squares estimates

- main effect of factor $A = 3$
- interaction between factors $A$ and $B = 4$
- interaction between factors $A$, $B$ and $C = 2$.

If a linear model including only an intercept and these three effects is fitted to the data, what is the value of the residual sum of squares?

Solution

The residual sum of squares has the form

$$\text{RSS} = (\boldsymbol{y} - X\hat{\boldsymbol{\beta}})^{\mathrm{T}}(\boldsymbol{y} - X\hat{\boldsymbol{\beta}})\,,$$

where in this case $X$ is a $2^6 \times 4$ model matrix, with columsn corresponding to the intercept, main effect of factor $A$, the interaction between factors $A$ and $B$, the interaction between factors $A$, $B$ and $C$. We can rewrite the RSS as

$$\begin{aligned}
\text{RSS} &= (\boldsymbol{y} - X\hat{\boldsymbol{\beta}})^{\mathrm{T}}(\boldsymbol{y} - X\hat{\boldsymbol{\beta}}) \\
&= \boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} - 2\boldsymbol{y}^{\mathrm{T}}X\hat{\boldsymbol{\beta}} + \hat{\boldsymbol{\beta}}^{\mathrm{T}}X^{\mathrm{T}}X\hat{\boldsymbol{\beta}} \\
&= \boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} - 2\hat{\boldsymbol{\beta}}^{\mathrm{T}}X^{\mathrm{T}}X\hat{\boldsymbol{\beta}} + \hat{\boldsymbol{\beta}}^{\mathrm{T}}X^{\mathrm{T}}X\hat{\boldsymbol{\beta}} \\
&= \boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} - \hat{\boldsymbol{\beta}}^{\mathrm{T}}X^{\mathrm{T}}X\hat{\boldsymbol{\beta}}\,,
\end{aligned}$$

as $\boldsymbol{y}^{\mathrm{T}}X = \hat{\boldsymbol{\beta}}^{\mathrm{T}}X^{\mathrm{T}}X$.

Due the matrix $X$ having orthogonal columns, $X^{\mathrm{T}}X = 2^f I_{p+1}$, for a model containing coefficients corresponding to $p$ factorial effects; here, $p = 3$. Hence,

$$\text{RSS} = \boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} - 2^f \sum_{i=0}^{p} \hat{\beta}_i^2\,.$$

Finally, the estimate of the intercept takes the form $\hat{\beta}_0 = \bar{Y}$, and so

$$\begin{aligned}
\text{RSS} &= \boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} - 2^f\bar{y} - 2^f \sum_{i=1}^{p} \hat{\beta}_i^2 \\
&= \sum_{i=1}^{2^f} (y_i - \bar{y})^2 - 2^f \sum_{i=1}^{p} \hat{\beta}_i^2 \\
&= \text{Total SS} - 2^f \sum_{i=1}^{p} \hat{\beta}_i^2
\end{aligned}$$

Recalling that each regression coefficient is one-half of the corresponding factorial effect, for this example we have:

$$\text{RSS} = 2856 - 2^6(1.5^2 + 2^2 + 1^2) = 2392\,.$$

3. (Adapted from Morris, 2011) Consider a $2^7$ experiment with each treatment applied to two units ($r = 2$). Assume a linear regression model will be fitted containing terms corresponding to all factorial effects.

   a. What is the variance of the estimator of each factorial effect, up to a constant factor $\sigma^2$?

   b. What is the variance of the least squares estimator of $E(y_{11})$, the expected value of an observation with the first treatment applied? You can assume the treatments are given in standard order, so the first treatment is defined by setting all factors to their low level. [The answer is, obviously, the same for $E(y_{12})$]. In a practical experimental setting, why is this not a useful quantity to estimate?

   c. What is the variance of the least squares estimator of $E(y_{11}) - E(y_{21})$? You may assume that the second treatment has all factors set to their low levels except for the seventh factor.

Solutions

a. Each factorial contrast is scaled so the variance for the estimator is equal to $4\sigma^2/n = \sigma^2/64$.

b. $E(y_1 1) = \boldsymbol{x}_1^{\mathrm{T}}\boldsymbol{\beta}$, where $\boldsymbol{x}_1^{\mathrm{T}}$ is the row of the $X$ matrix corresponding to the first treatment and $\boldsymbol{\beta}$ are the regression coefficients. The estimator is given by

$$\hat{E}(y_{11}) = \boldsymbol{x}_1^{\mathrm{T}}\hat{\boldsymbol{\beta}}\,,$$

with variance

$$\begin{aligned}
\mathrm{var}\left\{\hat{E}(y_{11})\right\} &= \mathrm{var}\left\{\boldsymbol{x}_1^{\mathrm{T}}\hat{\boldsymbol{\beta}}\right\} \\
&= \boldsymbol{x}_1^{\mathrm{T}}\mathrm{var}(\hat{\boldsymbol{\beta}})\boldsymbol{x}_1 \\
&= \boldsymbol{x}_1^{\mathrm{T}}\left(X^{\mathrm{T}}X\right)^{-1}\boldsymbol{x}_1\sigma^2 \\
&= \frac{\boldsymbol{x}_1^{\mathrm{T}}\boldsymbol{x}_1\sigma^2}{2^8} \\
&= \frac{2^7\sigma^2}{2^8} \\
&= \sigma^2/2\,.
\end{aligned}$$

This holds for the expected response from any treatment, as $\boldsymbol{x}_j^{\mathrm{T}}\boldsymbol{x}_j = 2^7$ for all treatments, as each entry of $\boldsymbol{x}_j$ is equal to $\pm 1$.

This would not be a useful quantity to estimate in a practical experiment, as it is not a contrast in the treatments. In particular, it depends on the

estimate of the overall mean, $\mu$ or $\beta_0$ (in the unit-treatment or regression model) that will vary from experiment to experiment.

c. The expected values of $y_{11}$ and $y_{21}$ will only differ in terms involving the seventh factor, which is equal to its low level (-1) for the first treatment and its high level (+1) for the second treatment; all the other terms will cancel. Hence

$$E(y_{11}) - E(y_{21}) = -2\left(\beta_7 + \sum_{j=1}^{6}\beta_{j7} + \sum_{j=1}^{6}\sum_{k=j+1}^{6}\beta_{jk7} + \ldots + \beta_{1234567}\right).$$

The variance of the estimator has the form

$$\begin{aligned}\operatorname{var}\left\{\widehat{E(y_{11}) - E(y_{21})}\right\} &= 4 \times \operatorname{var}\Bigg(\hat{\beta}_7 + \sum_{j=1}^{6}\hat{\beta}_{j7} + \sum_{j=1}^{6}\sum_{k=j+1}^{6}\hat{\beta}_{jk7} + \\ &\qquad \ldots + \hat{\beta}_{1234567}\Bigg) \\ &= \frac{4\sigma^2}{2 \times 2^7}\sum_{j=0}^{6}\binom{6}{j} \\ &= \frac{\sigma^2}{2^6} \times 64 \\ &= \sigma^2 .\end{aligned}$$

Or, as this is a treatment comparison in a CRD, we have

$$\hat{E}(y_{11}) - \hat{E}(y_{21}) = \widehat{\boldsymbol{c}^{\mathrm{T}}\boldsymbol{\tau}},$$

where $\boldsymbol{c}$ corresponds to a pairwise treatment comparison, and hence has one entry equal to +1 and one entry equal to -1. From Section 2.5,

$$\begin{aligned}\operatorname{var}\left(\widehat{\boldsymbol{c}^{\mathrm{T}}\boldsymbol{\tau}}\right) &= \sum_{i=1}^{t} c_i^2 \operatorname{var}(\bar{y}_{i\cdot}) \\ &= \sigma^2 \sum_{i=1}^{t} c_i^2/n_i ,\end{aligned}$$

where in this example $n_i = 2$ for all $i$ and $\sum_{i=1}^{t} c_i^2 = 2$. Hence, the variance is again equal to $\sigma^2$.

# Chapter 5

# Blocking in factorial designs

We now consider splitting the treatments in a factorial design into blocks. As in Chapter 3, the simplest factorial blocked design is a **randomised complete block design**, where the blocks are large enough for a complete replicate of the factorial treatments to occur in each block. Analysis then proceeds as in Chapter 3, with the contrasts of interest being those corresponding to the factorial effects (main effects and interactions).

However, the number of treatments grows rapidly in a factorial design, and it is unusual for the block sizes to be sufficiently large to accomodate a complete replication within each block. Hence, **incomplete block designs** must be employed. While balanced incomplete block designs (Section 3.6) can be used, they do not tend to have good statistical properties and their construction is complicated. In this chapter, we will focus on a class of methods specific to splitting a two-level factorial design in to blocks who common size $k$ is a power of two.

## 5.1 Two examples

We will use two simple examples to illustrate this approach, based on a $2^3$ experiment with factors labelled $A$, $B$ and $C$ (Table 5.1).

```
example.design <- FrF2::FrF2(nruns = 8, nfactors = 3, randomize = F)
knitr::kable(example.design, caption = "Treatments from a $2^3$ factorial design", align = rep("r
```

**Example 5.1.** Consider splitting the treatments between two blocks of size $2^{3-1} = 4$. One choice is given in Table 5.2.

```
block1 <- c(1, 2, 2, 1, 2, 1, 1, 2)
example.design.a <- cbind(example.design, Block = block1)
knitr::kable(example.design.a, caption = "Treatments from a $2^3$ factorial design split into two
```

Table 5.1: Treatments from a $2^3$ factorial design

| A | B | C |
|---|---|---|
| -1 | -1 | -1 |
| 1 | -1 | -1 |
| -1 | 1 | -1 |
| 1 | 1 | -1 |
| -1 | -1 | 1 |
| 1 | -1 | 1 |
| -1 | 1 | 1 |
| 1 | 1 | 1 |

Table 5.2: Treatments from a $2^3$ factorial design split into two blocks of size four.

| A | B | C | Block |
|---|---|---|---|
| -1 | -1 | -1 | 1 |
| 1 | -1 | -1 | 2 |
| -1 | 1 | -1 | 2 |
| 1 | 1 | -1 | 1 |
| -1 | -1 | 1 | 2 |
| 1 | -1 | 1 | 1 |
| -1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 2 |

Table 5.3: Unscaled factorial effect contrasts for a $2^3$ design with one possible assignment of treatments to blocks

| Treatment | Block | A | B | C | A:B | A:C | B:C | A:B:C |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | -1 | -1 | -1 | 1 | 1 | 1 | -1 |
| 2 | 2 | 1 | -1 | -1 | -1 | -1 | 1 | 1 |
| 3 | 2 | -1 | 1 | -1 | -1 | 1 | -1 | 1 |
| 4 | 1 | 1 | 1 | -1 | 1 | -1 | -1 | -1 |
| 5 | 2 | -1 | -1 | 1 | 1 | -1 | -1 | 1 |
| 6 | 1 | 1 | -1 | 1 | -1 | 1 | -1 | -1 |
| 7 | 1 | -1 | 1 | 1 | -1 | -1 | 1 | -1 |
| 8 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

To assess the impact of this choice of blocking scheme on the analysis of the experiment, we need to consider the (unscaled) contrasts corresponding to all the factorial effects, see Table 5.3.

```
X <- model.matrix( ~ Block + (A + B + C)^3, data = example.design.a)
Xdf <- data.frame(X[, -1])
colnames(Xdf) <- c("Block", "A", "B", "C", "A:B", "A:C", "B:C", "A:B:C")
Xdf <- dplyr::mutate(Xdf, Treatment = 1:8, .before = Block)
knitr::kable(Xdf, caption = "Unscaled factorial effect contrasts for a $2^3$ design with one poss
```

Each contrast vector in Table 5.3 is orthogonal, in the sense of -1 and +1 occuring equally often (twice) in each block, except for the contrast vector for the three-factor interaction. This vector has all -1 entries occuring in block 1, and all +1 entries occuring in block 2.

The difference in average response between blocks 1 and 2 in this design is estimated by

$$\widehat{\beta_1 - \beta_2} = \frac{1}{4}\left\{(y_{11} + y_{14} + y_{16} + y_{17}) - (y_{22} + y_{23} + y_{25} + y_{28})\right\} ,$$

where $\beta_i$ is the effect of the $i$th block in unit-block-treatment model (3.1) and $y_{ij}$ is the response from applying treatment $j$ to a unit in block $i$ ($i = 1, 2$; $j = 1, \ldots, 8$).

This contrast is **exactly** the same as the contrast for estimating the three-factor interaction.

$$\text{Int}(A, B, C) = \frac{1}{4}\left\{(y_{11} + y_{14} + y_{16} + y_{17}) - (y_{22} + y_{23} + y_{25} + y_{28})\right\} .$$

Hence this choice of blocking makes it impossible for us to estimate this interaction. If the contrast is large, we would anticipate it was because there is a large difference in average response between blocks, not because of the three-factor interaction.

So why choose this particular blocking? Well, it is impossible to split this set of treatments into incomplete blocks (with $k < 8$) and not lose some information about the factorial effects.

1. From effect hierarchy, the three-factor interaction is the least likely factorial effect to be important, and hence this is the interaction we care least about losing information about. Choosing any of the other factorial effects to determine the blocking would be a worst choice[1]

2. What about if we don't use a column of Table 5.3 to assign treatments to blocks? We now longer lose all information about a particular factorial effect, but instead we lose some information about many, or even all, factorial effects.

We can study this information lose by assessing all $\binom{8}{4}! = 70$ possible assignments of treatments to blocks. For each, we will calculate the average variance of the main effect and two-factor interaction contrasts (up to a constant $\sigma^2$).

```
no.assign <- choose(8, 4)
assignments <- combinat::combn(8, 4)
yfake <- rnorm(8)
Xadf <- cbind(Xdf[, c(-1, -9)], y = yfake)
avgvar <- NULL
for(i in 1:no.assign) {
  B <- rep(1, 8)
  B[assignments[, i]] <- -1
  Xadf$Block <- B
  temp.lm <- lm(y ~ Block + (A + B + C)^2, data = Xadf)
  temp.lm$residuals <- yfake
  temp.lm$df.residual <- 8
  vmat <- vcov(temp.lm) / (summary(temp.lm)$sigma^2)
  vars <- (diag(vmat[-c(1:2), -c(1:2)]))
  tidyr::replace_na(vars, Inf)
  avgvar[i] <- sum(tidyr::replace_na(vars, Inf)) / 6
}
knitr::kable(table(avgvar), col.names = c("Avg. variance", "Freq."))
```

| Avg. variance | Freq. |
|---|---|
| 0.125 | 2 |
| 0.1875 | 32 |
| Inf | 36 |

From our study, we see that there are two allocations of treatments to blocks that give us the smallest average variance of $0.125\sigma^2$. These two allocations are those that use the three-factor interaction column to assign treatments to blocks. For 32 other allocations, the average variance $0.188\sigma^2$, and hence an efficiency of

[1] In the absence of any prior information telling us that certain effects are of particular interest, we always favour losing information on higher order effects.

Table 5.4: Scaled inner-products between contrast vectors for $2^3$ with treatments assigned to blocks so Blocks = $ABC$.

| | Block | A | B | C | A:B | A:C | B:C | A:B:C |
|---|---|---|---|---|---|---|---|---|
| Block | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| A | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| B | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| C | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| A:B | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| A:C | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| B:C | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| A:B:C | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

Table 5.5: Scaled inner-products between contrast vectors for $2^3$ with treatments assigned to blocks arbitrarily.

| | Block | A | B | C | A:B | A:C | B:C | A:B:C |
|---|---|---|---|---|---|---|---|---|
| Block | 1.0 | 0.5 | 0.5 | 0.5 | 0 | 0 | 0 | -0.5 |
| A | 0.5 | 1.0 | 0.0 | 0.0 | 0 | 0 | 0 | 0.0 |
| B | 0.5 | 0.0 | 1.0 | 0.0 | 0 | 0 | 0 | 0.0 |
| C | 0.5 | 0.0 | 0.0 | 1.0 | 0 | 0 | 0 | 0.0 |
| A:B | 0.0 | 0.0 | 0.0 | 0.0 | 1 | 0 | 0 | 0.0 |
| A:C | 0.0 | 0.0 | 0.0 | 0.0 | 0 | 1 | 0 | 0.0 |
| B:C | 0.0 | 0.0 | 0.0 | 0.0 | 0 | 0 | 1 | 0.0 |
| A:B:C | -0.5 | 0.0 | 0.0 | 0.0 | 0 | 0 | 0 | 1.0 |

0.6649 compared to the first two allocations. There are also 36 allocations that have infinite average variance; these allocations use one of the six main effect or two-factor interaction columns to assign treatments to blocks. For any of these choices, the corresponding factorial effect cannot be estimated, equivalent to the estimator having infinite variance.

We now compare our original design to one of the 32 allocations with average variance $0.188\sigma^2$ (chosen arbitrarily).

```
Xa <- as.matrix(Xdf[, -c(1)])
Xb <- Xa
B <- rep(1, 8)
B[assignments[, 2]] <- -1
Xb[,1] <- B
```

```
knitr::kable(cor(Xa), caption = "Scaled inner-products between contrast vectors for $2^3$ with tr
```

Table 5.6: Treatments from a $2^3$ factorial design split into four blocks of size two.

| A | B | C | Block |
|---|---|---|---|
| -1 | -1 | -1 | 4 |
| 1 | -1 | -1 | 3 |
| -1 | 1 | -1 | 2 |
| 1 | 1 | -1 | 1 |
| -1 | -1 | 1 | 1 |
| 1 | -1 | 1 | 2 |
| -1 | 1 | 1 | 3 |
| 1 | 1 | 1 | 4 |

```
knitr::kable(cor(Xb), caption = "Scaled inner-products between contrast vectors for $2
```

From Table 5.4, we can see that the block contrast is orthogonal to (has zero inner product with) all the main effect and two-factor interaction contrasts. In comparison, the design with arbitrary treatment assignment has non-zero inner products between blocks and the main effect contrasts (Table 5.5). However, this design does allow estimation of the three-factor interaction (although it too has non-zero inner-product with the block contrast).[2]

Clearly, if interest is in estimation of the main effects and two-factor interactions, it is best to use the design which assigns treatments to blocks via the three-factor interaction contrast coefficients.

**Definition 5.1.** A factorial effect is said to be **confounded** with blocks if the same contrast in the observations estimates both the factorial effect and a difference between blocks.

In Example 5.1, the three-factor interaction $ABC$ is confounded with blocks. We write Blocks $= ABC$ as a shorthand to represent this confounding.

**Example 5.2.** Now consider splitting the treatments between four blocks of size $2^{3-2} = 2$. One choice is given in Table 5.6.

```
block2 <- c(4, 3, 2, 1, 1, 2, 3, 4)
example.design.b <- cbind(example.design, Block = block2)
knitr::kable(example.design.b, caption = "Treatments from a $2^3$ factorial design spli
```

Clearly, we cannot use a single factorial contrast (taking only two values) to divide treatments between four blocks. The obvious extension to the approach from Example 5.1 is to use the combination of two columns. Here, we have used the contrasts for the $AB$ and $AC$ interactions.

[2]Also, simultaneous estimation of the intercept, blocking effect and all 8 factorial effects is not possible, as there are insufficient degrees of freedom. In fact, the rank of the combined treatment and block model matrices for the unit-block-treatment model ((??)) is t+b-3.

Table 5.7: Unscaled factorial effect contrasts for a $2^3$ design with one possible assignment of treatments to four blocks of size two.

| Treatment | Block | A | B | C | A:B | A:C | B:C | A:B:C |
|---|---|---|---|---|---|---|---|---|
| 1 | 4 | -1 | -1 | -1 | 1 | 1 | 1 | -1 |
| 2 | 3 | 1 | -1 | -1 | -1 | -1 | 1 | 1 |
| 3 | 2 | -1 | 1 | -1 | -1 | 1 | -1 | 1 |
| 4 | 1 | 1 | 1 | -1 | 1 | -1 | -1 | -1 |
| 5 | 1 | -1 | -1 | 1 | 1 | -1 | -1 | 1 |
| 6 | 2 | 1 | -1 | 1 | -1 | 1 | -1 | -1 |
| 7 | 3 | -1 | 1 | 1 | -1 | -1 | 1 | -1 |
| 8 | 4 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

```
X <- model.matrix( ~ Block + (A + B + C)^3, data = example.design.b)
Xdf <- data.frame(X[, -1])
colnames(Xdf) <- c("Block", "A", "B", "C", "A:B", "A:C", "B:C", "A:B:C")
Xdf <- dplyr::mutate(Xdf, Treatment = 1:8, .before = Block)
knitr::kable(Xdf, caption = "Unscaled factorial effect contrasts for a $2^3$ design with one poss
```

In Table 5.7 we can see that the contrasts for $AB$ and $AC$ are confounded with blocks (the contrast coefficients are constant within each blocks).

- Block 1 contains treatments 4 and 5, which have $AB = +1$ and $AC = -1$.
- Block 2 contains treatments 3 and 6, which have $AB = -1$ and $AC = +1$.
- Block 3 contains treatments 2 and 7, whicn have $AB = AC = -1$.
- Block 4 contains treatments 1 and 8, which have $AB = AC = +1$.

However, by confounding interactions $AB$ and $AC$ we have also confounded the elementwise (Hadamard) product of these two interactions:

$$AB \odot AC = A \odot B \odot A \odot C = B \odot C = BC\,.$$

If the contrast vectors for $AB$ and $AC$ are constant, the contrast vector for interaction $BC$ must also be constant. Hence, interaction $BC$ is also confounded with blocks.

We write this confounding as

$$\text{Block}_1 = AB\,, \quad \text{Block}_2 = AC\,, \quad \text{Block}_3 = \text{Block}_1 \times \text{Block}_2 = BC\,.$$

Clearly, alternative blocking schemes are possible. However, we must be careful not to inadvertently confound low-order factorial effects. For example, if we chose to confound the three-factor interaction $ABC$ with blocks, along with one two-factor interaction, say $BC$, then we also confound

$$ABC \odot BC = A \odot B \odot C \odot B \odot C = A\,.$$

Hence, the main effect of factor $A$ is also confounded with blocks. This is clearly undesirable, e.g. by effect hierarchy.

We shouldn't be surprised that a third factorial effect was confounded with blocks. In the unit-block-treatment model (3.2), the rank of the block model matrix $X_1$ is equal to 3, and hence there will be three degrees of freedom required to estimate the blocking effects.

As in Example 5.1, we could also explore alternative blocking schemes that do not completely confound factorial effects with blocks. However, as before, these alternatives would lead to higher average variance for the estimation of main effects compared to the design that confounds the three two-factor interactions.[3]

**Definition 5.2.** In a blocked factorial design, those effects which are not confounded with blocks are called **clear**.

In Example 5.1, the clear effects are $A$, $B$, $C$, $AB$, $AC$ and $BC$. In Example 5.2, the clear effects are $A$, $B$, $C$ and $ABC$.

## 5.2 General method of constructing a confounded block design

To arrange a $2^f$ design in $b = 2^q$ blocks of size $k = 2^{f-q}$:

- choose $q$ independent factorial contrasts for the **defining blocks**. Typically, we choose higher-order interactions (due to effect hierarchy):

$$\text{Block}_1 = \boldsymbol{c}_1, \ldots, \text{Block}_q = \boldsymbol{c}_q\,.$$

- all the hadamard products of $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_q$ are also confounded with blocks:

$$\begin{array}{rcl} \text{Block}_1\text{Block}_2 & = & \boldsymbol{c}_1 \odot \boldsymbol{c}_2 \\ \text{Block}_1\text{Block}_3 & = & \boldsymbol{c}_1 \odot \boldsymbol{c}_3 \\ \vdots & = & \vdots \\ \text{Block}_1\text{Block}_2 \ldots \text{Block}_q & = & \boldsymbol{c}_1 \odot \boldsymbol{c}_2 \odot \cdots \odot \boldsymbol{c}_q \end{array}$$

- in total, $2^q - 1$ factorial effects will be confounded with blocks.

For example, a $2^8$ design in $b = 2^3 = 8$ blocks of size $k = 2^{8-3} = 2^5 = 32$. We choose the following $q = 3$ defining blocks:

$$\text{Block}_1 = ACEGH\,, \quad \text{Block}_2 = BCFGH\,, \quad \text{Block}_3 = BDEGH\,.$$

[3] This design is too small to split into four blocks and still be able to estimate all two-factor interactions clear of blocks.

Table 5.8: $2^3$ factorial design in two blocks of size four

| Treatment | Block | A | B | C | A:B | A:C | B:C | A:B:C |
|---|---|---|---|---|---|---|---|---|
| 1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 | -1 |
| 2 | -1 | -1 | 1 | 1 | -1 | -1 | 1 | -1 |
| 3 | -1 | 1 | -1 | 1 | -1 | 1 | -1 | -1 |
| 4 | -1 | 1 | 1 | -1 | 1 | -1 | -1 | -1 |
| 5 | 1 | -1 | -1 | 1 | 1 | -1 | -1 | 1 |
| 6 | 1 | -1 | 1 | -1 | -1 | 1 | -1 | 1 |
| 7 | 1 | 1 | -1 | -1 | -1 | -1 | 1 | 1 |
| 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

We obtain the other confounded effects by hadamard multiplication:

$$\begin{aligned} \text{Block}_1 \odot \text{Block}_2 &= ABEF \\ \text{Block}_1 \odot \text{Block}_3 &= ABCD \\ \text{Block}_2 \odot \text{Block}_3 &= CDEF \\ \text{Block}_1 \odot \text{Block}_2 \odot \text{Block}_3 &= ADFGH\,. \end{aligned}$$

It is also straightforward to find blocked fractional factorial designs using `FrF2` in `R`. For example, to find the two $2^3$ designs at the start of this chapter, with blocks of size $k = 4$ and $k = 2$, we simply set the `blocks` argument equal to the number of blocks $b$, see Tables 5.8 and 5.9[4].

```
block1.frf2 <- FrF2::FrF2(nruns = 8, nfactors = 3, blocks = 2,
                          alias.info = 3, randomize = F)
block1 <- data.frame(model.matrix(~ Blocks + (A + B + C)^3, block1.frf2))
block1 <- dplyr::mutate(block1, Treatment = 1:8, .before = Blocks1)
knitr::kable(block1[, -1], col.names = c("Treatment", "Block", "A", "B", "C",
                                         "A:B", "A:C", "B:C", "A:B:C"),
             caption = "$2^3$ factorial design in two blocks of size four")
```

```
block2.frf2 <- FrF2::FrF2(nruns = 8, nfactors = 3, blocks = 4,
                          alias.info = 3, randomize = F, alias.block.2fis = T)
block2 <- data.frame(model.matrix(~ Blocks + (A + B + C)^3, block2.frf2))
block2 <- dplyr::mutate(block2, Treatment = 1:8, .before = Blocks1)
knitr::kable(block2[, -1],
             col.names = c("Treatment", "Block1", "Block2", "Block3",
                           "A", "B", "C",
                           "A:B", "A:C", "B:C", "A:B:C"),
             caption = "$2^3$ factorial design in four blocks of size two")
```

[4]The additional `R` code is to tidy up the output.

Table 5.9: $2^3$ factorial design in four blocks of size two

| Treatment | Block1 | Block2 | Block3 | A | B | C | A:B | A:C | B:C | A:B:C |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | -1 | -1 | 1 | -1 | 1 | 1 | -1 | -1 | 1 | -1 |
| 2 | -1 | -1 | 1 | 1 | -1 | -1 | -1 | -1 | 1 | 1 |
| 3 | 1 | -1 | -1 | -1 | 1 | -1 | -1 | 1 | -1 | 1 |
| 4 | 1 | -1 | -1 | 1 | -1 | 1 | -1 | 1 | -1 | -1 |
| 5 | -1 | 1 | -1 | -1 | -1 | 1 | 1 | -1 | -1 | 1 |
| 6 | -1 | 1 | -1 | 1 | 1 | -1 | 1 | -1 | -1 | -1 |
| 7 | 1 | 1 | 1 | -1 | -1 | -1 | 1 | 1 | 1 | -1 |
| 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

In each case, `FrF2` returns for us the design, in coded $\pm 1$ units (or the unscaled factorial constrast coefficients), including columns giving the contrast coefficients for estimating the block effects. The function automatically tries to find the best allocation of treatments to blocks, in terms of maximising the number of lower-order factorial effects which are clear of blocks. In the second example, we set `alias.block.2fis = T` to allow `FrF2` to confound two-factor interactions with blocks, otherwise a solution could not be found.

Setting `alias.info = 3` ensures `FrF2` returns information about confounding between blocks and three-factor interactions[5] We extract this information using `design.info`[6].

```
library(FrF2)
design.info(block1.frf2)$aliased.with.blocks
```

```
## [1] "ABC"
```

```
design.info(block2.frf2)$aliased.with.blocks
```

```
## [1] "AB" "AC" "BC"
```

We can also specify which factorial effects we wish to confound with blocks, rather than letting `FrF2` choose. For the $2^8$ example above, we can specify the three defining blocks

$$\text{Block}_1 = ACEGH\,, \quad \text{Block}_2 = BCFGH\,, \quad \text{Block}_3 = BDEGH\,,$$

the `blocks` argument

[5]By default, `FrF2` only returns information about confounding involving two-factor interactions. Unfortunately, it is not possible to get `FrF2` to return information about confounding involving four-factor or higher interactions.

[6]We previously have not explicitly loaded packages before. However, `design.info` is actually extracting *attributes* of the `FrF2` object, and I couldn't find a way to make this work without loading the package!

```
block3.frf2 <- FrF2::FrF2(nruns = 2^8, nfactors = 8,
                    alias.info = 3, randomize = F, blocks = c("ACEGH", "BCFGH", "BDEGH"))
```

## 5.3 Analysis of a confounded factorial design

We can analyse a confounded design by by combining ideas from Chapters 3 and 4. The most straightforward approach is to add a block effect to the regression model introduced in Section 4.4. Let $X$ be the $n \times d$ model matrix with columns corresponding to the $d = 2^f - b$ unscaled contrast coefficients for the estimable factorial effects (i.e. those **not** confounded with blocks). As no factorial effects that have been confounded with blocks are estimable, the $X$ matrix cannot include columns corresponding to these effects. We write a block-regression model as

$$\boldsymbol{y} = \mathbf{1}_n \beta_0 + X\boldsymbol{\beta} + + Z\boldsymbol{\gamma} + \boldsymbol{\varepsilon}\,,$$

where in a (necessary) change of notation, $Z$ is the $n \times b$ model matrix for blocks (previously referred to as $X_1$) and $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_b)^{\mathrm{T}}$ is the vector of block effects (previously called $\boldsymbol{\beta}$). The errors remain independent and identically normally distributed with constant variance. Recall that for equal block sizes

$$Z = \bigoplus_{i=1}^{b} \mathbf{1}_k = \begin{bmatrix} \mathbf{1}_k & \mathbf{0}_k & \cdots & \mathbf{0}_k \\ \mathbf{0}_k & \mathbf{1}_k & \cdots & \mathbf{0}_k \\ \vdots & & \ddots & \vdots \\ \mathbf{0}_k & \mathbf{0}_k & \cdots & \mathbf{1}_k \end{bmatrix}.$$

The normal equations take the form

$$n\hat{\beta}_0 + \mathbf{1}_n^{\mathrm{T}} X\hat{\boldsymbol{\beta}} + \mathbf{1}_n^{\mathrm{T}} Z\hat{\boldsymbol{\gamma}} = \mathbf{1}_n^{\mathrm{T}}\boldsymbol{y}\,, \tag{5.1}$$

$$X^{\mathrm{T}}\mathbf{1}_n\hat{\beta}_0 + X^{\mathrm{T}}X\hat{\boldsymbol{\beta}} + X^{\mathrm{T}}Z\hat{\boldsymbol{\gamma}} = X^{\mathrm{T}}\boldsymbol{y}\,, \tag{5.2}$$

$$Z^{\mathrm{T}}\mathbf{1}_n\hat{\beta}_0 + Z^{\mathrm{T}}X\hat{\boldsymbol{\beta}} + Z^{\mathrm{T}}Z\hat{\boldsymbol{\tau}} = Z^{\mathrm{T}}\boldsymbol{y}\,. \tag{5.3}$$

$$\tag{5.4}$$

However, we have that $X^{\mathrm{T}}\mathbf{1}_n = \mathbf{0}_d$, as all factorial contrasts are orthogonal to the constant vector. Also, $X^{\mathrm{T}}Z = \mathbf{0}_{d\times b}$, as each factorial contrasts takes values -1 and +1 an equal number of times in each block. Also, $X^{\mathrm{T}}X = nI_d$, as all the vectors of factorial contrast coefficients are orthogonal. Hence, the regression coefficients $\boldsymbol{\beta}$ can be estimated independently of the intercept and block effects, with reduced normal equations

$$n\hat{\boldsymbol{\beta}} = X^{\mathrm{T}}\boldsymbol{y}\,.$$

Obviously, there is a unique solution for $\hat{\boldsymbol{\beta}}$,

$$\hat{\boldsymbol{\beta}} = \frac{1}{n} X^{\mathrm{T}} \boldsymbol{y}\,,$$

with estimators of the factorial effects having the form $2\hat{\boldsymbol{\beta}}$ (see Section 4.4).

For a simulated response in Example 5.2, we can perform this analysis in `R` using `lm`.

```
ex2.df <- dplyr::mutate_at(Xdf,c("Block"), factor)
X <- model.matrix(~ Block + (A + B + C)^3, ex2.df)
coef.values <- c(0, 2, 6, 12, 4, 0, 3, 0, 2, 0, 0)
y <- X %*% coef.values + rnorm(8, 0, 1)
betahat <- t(X[, c(5, 6, 7, 11)]) %*% y / nrow(X)
betahat # coef estimates, obtained directly
```

```
##           [,1]
## A      4.46086
## B      0.09589
## C      3.16320
## A:B:C -0.33697
```

```
ex2.df <- data.frame(ex2.df, y = y)
ex2.lm <- lm(y ~  Block + A + B + C + A:B:C, data = ex2.df)
anova(ex2.lm)
```

```
## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq F value Pr(>F)
## Block      3  240.9    80.3     NaN    NaN
## A          1  159.2   159.2     NaN    NaN
## B          1    0.1     0.1     NaN    NaN
## C          1   80.0    80.0     NaN    NaN
## A:B:C      1    0.9     0.9     NaN    NaN
## Residuals  0    0.0     NaN
```

```
coef(ex2.lm)[5:8] # coef estimates
```

```
##        A        B        C    A:B:C
##  4.46086  0.09589  3.16320 -0.33697
```

```
2 * coef(ex2.lm)[5:8] # factorial effects
```

```
##       A       B       C   A:B:C
##  8.9217  0.1918  6.3264 -0.6739
```

As the two-factor interactions are confounded with blocks, we can get an equivalent analysis by including these factorial effects but excluding blocks.

```
ex2.lm2 <- lm(y ~  (A + B + C)^3, data = ex2.df)
anova(ex2.lm2)
```

```
## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq F value Pr(>F)
## A          1  159.2   159.2     NaN    NaN
## B          1    0.1     0.1     NaN    NaN
## C          1   80.0    80.0     NaN    NaN
## A:B        1    4.4     4.4     NaN    NaN
## A:C        1  103.1   103.1     NaN    NaN
## B:C        1  133.4   133.4     NaN    NaN
## A:B:C      1    0.9     0.9     NaN    NaN
## Residuals  0    0.0     NaN
```

```
coef(ex2.lm2)
```

```
## (Intercept)           A           B           C         A:B         A:C
##     4.45268     4.46086     0.09589     3.16320     0.73940     3.58982
##         B:C       A:B:C
##     4.08390    -0.33697
```

But we must keep in mind that the degrees of freedom and coefficient estimates for the two-factor interactions are actually associated with blocks.

## 5.4 Exercises

1. Suppose that a single replicate $2^5$ factorial experiment is to be arranged in $b = 4$ blocks, each containing $k = 8$ treatments.

   How many interactions need to be confounded with blocks in this experiment? Can these be chosen independently? Select suitable interactions which could be confounded with blocks and write down the treatments in one block corresponding to your choice.

Solution

To run a $2^5$ experiment in $b = 4$ blocks requires $b - 1 = 3$ interactions to be confounded and these cannot be chosen independently; the choice of two determines the third. Confounding the interactions $ABCD$, $CDE$ and their product $ABE$ may be a sensible choice, as no main effects or two-factor interactions are then confounded with blocks. (Other choices are acceptable).

To find a block for this choice, we can solve the equations

$ABCD = 1$, $CDE =$ -1

which gives the treatments

| A | B | C | D | E |
|---|---|---|---|---|
| -1 | -1 | -1 | -1 | -1 |
| 1 | 1 | -1 | -1 | -1 |
| -1 | -1 | 1 | 1 | -1 |
| 1 | 1 | 1 | 1 | -1 |
| 1 | -1 | 1 | -1 | 1 |
| -1 | 1 | 1 | -1 | 1 |
| 1 | -1 | -1 | 1 | 1 |
| -1 | 1 | -1 | 1 | 1 |

This same block can be found using `FrF2`.

```
block.2.5 <- FrF2::FrF2(nruns = 2^5, nfactors = 5, blocks = c("ABCD", "CDE"))
subset(block.2.5, Blocks == 3)
```

```
##    Blocks  A  B  C  D  E
## 17      3  1 -1 -1  1  1
## 18      3 -1  1 -1  1  1
## 19      3 -1  1  1 -1  1
## 20      3  1  1 -1 -1 -1
## 21      3  1  1  1  1 -1
## 22      3 -1 -1  1  1 -1
## 23      3  1 -1  1 -1  1
## 24      3 -1 -1 -1 -1 -1
```

2. Suppose a $2^7$ design is arranged in $b = 8$ blocks with the defining blocks $\text{Block}_1 = ABC$, $\text{Block}_2 = DEF$ and $\text{Block}_3 = AFG$.

    a. Find all the interactions that are confounded with blocks.

    b. An alternative blocking scheme has $\text{Block}_1 = ABCD$, $\text{Block}_2 = ABEF$ and $\text{Block}_3 = ACEG$. Find all the confounded interactions in this design, and compare with the original blocking scheme.

Solution

a. $\text{Block}_1 = ABC$, $\text{Block}_2 = DEF$, $\text{Block}_1\text{Block}_2 = ABCDEF$, $\text{Block}_3 = AFG$, $\text{Block}_1\text{Block}_3 = BCFG$, $\text{Block}_2\text{Block}_3 = ADEG$, $\text{Block}_1\text{Block}_2\text{Block}_3 = BCDEG$.

b. For the alternative blocking scheme, $\text{Block}_1 = ABCD$, $\text{Block}_2 = ABEF$, $\text{Block}_1\text{Block}_2 = CDEF$, $\text{Block}_3 = ACEG$, $\text{Block}_1\text{Block}_3 = BDEG$, $\text{Block}_2\text{Block}_3 = BCFG$, $\text{Block}_1\text{Block}_2\text{Block}_3 = ADFG$.

The second scheme does not confound any three-factor interactions with blocks, unlike the original scheme.

3. Suppose that the $2^3$ design

| Run | A | B | C | AB | AC | BC | ABC |
|---|---|---|---|---|---|---|---|
| 1 | -1 | -1 | -1 | +1 | +1 | +1 | -1 |
| 2 | -1 | -1 | +1 | +1 | -1 | -1 | +1 |
| 3 | -1 | +1 | -1 | -1 | +1 | -1 | +1 |
| 4 | -1 | +1 | +1 | -1 | -1 | +1 | -1 |
| 5 | +1 | -1 | -1 | -1 | -1 | +1 | +1 |
| 6 | +1 | -1 | +1 | -1 | +1 | -1 | -1 |
| 7 | +1 | +1 | -1 | +1 | -1 | -1 | -1 |
| 8 | +1 | +1 | +1 | +1 | +1 | +1 | +1 |

is arranged in two blocks with runs 1, 2, 5 and 7 in block 1 and runs 3, 4, 6 and 8 in block 2.

a. Show that a contrast vector defining the difference between the two blocks is not identical to any of the seven contrast vectors in the table and, therefore, it is not confounded with any of the factorial effects.

b. Show that the block effect is not orthogonal to some of the factorial effects. Identify these effects. Discuss why it is undesirable to use this blocking scheme.

Solution

a. A unscaled contrast vector (with coefficients -1/+1) has been added to the table below. It is clearly not identical to any of the other columns, and hence no factorial effects have been completely confounded with blocks.

| Run | A | B | C | AB | AC | BC | ABC | Block |
|---|---|---|---|---|---|---|---|---|
| 1 | -1 | -1 | -1 | +1 | +1 | +1 | -1 | -1 |
| 2 | -1 | -1 | +1 | +1 | -1 | -1 | +1 | -1 |
| 3 | -1 | +1 | -1 | -1 | +1 | -1 | +1 | +1 |
| 4 | -1 | +1 | +1 | -1 | -1 | +1 | -1 | +1 |
| 5 | +1 | -1 | -1 | -1 | -1 | +1 | +1 | -1 |
| 6 | +1 | -1 | +1 | -1 | +1 | -1 | -1 | +1 |
| 7 | +1 | +1 | -1 | +1 | -1 | -1 | -1 | -1 |
| 8 | +1 | +1 | +1 | +1 | +1 | +1 | +1 | +1 |

b. The inner products of each effect column with the blocking column are given by

| A | B | C | AB | AC | BC | ABC |
|---|---|---|---|---|---|---|
| 0 | +4 | +4 | -4 | +4 | 0 | 0 |

The blocking effect is not orthogonal to the main effects of $B$ or $C$ or the interactions $AB$ and $AC$. Two of the main effects are *partially* confounded

with the blocking effect, and the variances of the estimators of these effects will inflated. Therefore, this scheme is not desirable. In general, we may prefer to completely confound a smaller number of higher-order factorial effects with blocks, in order to have greater clarity in our design and to not lose information about main effects.

4. In the Chapter 4 exercises we studied the reactor experiment, a $2^5$ full factorial experiment with the following data:

```
reactor.frf2 <- FrF2::FrF2(nruns = 32, nfactors = 5, randomize = F,
                           factor.names = c("FR", "Cat", "AR", "Temp", "Conc"))
y <- c(61, 53, 63, 61, 53, 56, 54, 61, 69, 61, 94, 93, 66, 60, 95, 98, 56, 63,
       70, 65, 59, 55, 67, 65, 44, 45, 78, 77, 49, 42, 81, 82)
reactor <- data.frame(reactor.frf2, pre.react = y)
knitr::kable(reactor, caption = "Reactor experiment.")
```

   Assume now that this experiment was performed in $b = 4$ blocks, confounding the interactions between `FR`, `Cat` and `AR` and between `FR`, `Temp` and `Conc`. Obtain the anova table for this new experiment, and estimate the interactions that are not confounded with blocks. Use Lenth's method to decide which effects are important.

Solution

For ease of notation, we will label the factors

| Factor | Label |
|---|---|
| FR | A |
| Cat | B |
| AR | C |
| Temp | D |
| Conc | E |

The design has defining blocks $\text{Block}_1 = ABC$ and $\text{Block}_2 = ADE$, and hence also confounds $\text{Block}_1\text{Block}_2 = BCDE$. We can assign treatments to blocks according to the four combinations of any pair of these interactions. In the code chunk below, we use the pipe operator (`|>`) to combine our data processing steps.

```
fac_to_numeric <- function(x) as.numeric(as.character(x))
reactor <- reactor |> dplyr::mutate(block1 = fac_to_numeric(FR) * fac_to_numeric(Cat) *
                                    fac_to_numeric(AR),
                                  block2 = fac_to_numeric(FR) * fac_to_numeric(Temp) *
                                    fac_to_numeric(Conc)) |>
  dplyr::arrange(block1, block2) |>
  dplyr::mutate(block = factor(rep(1:4, rep(8, 4)))) |>
  dplyr::relocate(block) |>
```

Table 5.14: Reactor experiment.

| FR | Cat | AR | Temp | Conc | pre.react |
|---|---|---|---|---|---|
| -1 | -1 | -1 | -1 | -1 | 61 |
| 1 | -1 | -1 | -1 | -1 | 53 |
| -1 | 1 | -1 | -1 | -1 | 63 |
| 1 | 1 | -1 | -1 | -1 | 61 |
| -1 | -1 | 1 | -1 | -1 | 53 |
| 1 | -1 | 1 | -1 | -1 | 56 |
| -1 | 1 | 1 | -1 | -1 | 54 |
| 1 | 1 | 1 | -1 | -1 | 61 |
| -1 | -1 | -1 | 1 | -1 | 69 |
| 1 | -1 | -1 | 1 | -1 | 61 |
| -1 | 1 | -1 | 1 | -1 | 94 |
| 1 | 1 | -1 | 1 | -1 | 93 |
| -1 | -1 | 1 | 1 | -1 | 66 |
| 1 | -1 | 1 | 1 | -1 | 60 |
| -1 | 1 | 1 | 1 | -1 | 95 |
| 1 | 1 | 1 | 1 | -1 | 98 |
| -1 | -1 | -1 | -1 | 1 | 56 |
| 1 | -1 | -1 | -1 | 1 | 63 |
| -1 | 1 | -1 | -1 | 1 | 70 |
| 1 | 1 | -1 | -1 | 1 | 65 |
| -1 | -1 | 1 | -1 | 1 | 59 |
| 1 | -1 | 1 | -1 | 1 | 55 |
| -1 | 1 | 1 | -1 | 1 | 67 |
| 1 | 1 | 1 | -1 | 1 | 65 |
| -1 | -1 | -1 | 1 | 1 | 44 |
| 1 | -1 | -1 | 1 | 1 | 45 |
| -1 | 1 | -1 | 1 | 1 | 78 |
| 1 | 1 | -1 | 1 | 1 | 77 |
| -1 | -1 | 1 | 1 | 1 | 49 |
| 1 | -1 | 1 | 1 | 1 | 42 |
| -1 | 1 | 1 | 1 | 1 | 81 |
| 1 | 1 | 1 | 1 | 1 | 82 |

```
  dplyr::select(!c(block1, block2))
knitr::kable(reactor, caption = "Reactor experiment in blocks.")
```

We can produce an ANOVA table with the following code; `lm` and `anova` should automatically get the degrees of freedom correct (thr functions will recognise that three interactions are confounded with blocks and not estimable). The three confounded interactions do not appear in the `anova` output.

```
reactor.lm <- lm(pre.react ~ block + (FR + Cat + AR + Temp + Conc) ^ 5, data = reactor)
anova(reactor.lm)
```

```
## Analysis of Variance Table
##
## Response: pre.react
##                      Df Sum Sq Mean Sq F value Pr(>F)
## block                 3     24       8     NaN    NaN
## FR                    1     15      15     NaN    NaN
## Cat                   1   3042    3042     NaN    NaN
## AR                    1      3       3     NaN    NaN
## Temp                  1    925     925     NaN    NaN
## Conc                  1    312     312     NaN    NaN
## FR:Cat                1     15      15     NaN    NaN
## FR:AR                 1      4       4     NaN    NaN
## FR:Temp               1      6       6     NaN    NaN
## FR:Conc               1      0       0     NaN    NaN
## Cat:AR                1      6       6     NaN    NaN
## Cat:Temp              1   1404    1404     NaN    NaN
## Cat:Conc              1     32      32     NaN    NaN
## AR:Temp               1     36      36     NaN    NaN
## AR:Conc               1      6       6     NaN    NaN
## Temp:Conc             1    968     968     NaN    NaN
## FR:Cat:Temp           1     15      15     NaN    NaN
## FR:Cat:Conc           1     28      28     NaN    NaN
## FR:AR:Temp            1      4       4     NaN    NaN
## FR:AR:Conc            1     50      50     NaN    NaN
## Cat:AR:Temp           1     10      10     NaN    NaN
## Cat:AR:Conc           1      0       0     NaN    NaN
## Cat:Temp:Conc         1      0       0     NaN    NaN
## AR:Temp:Conc          1      0       0     NaN    NaN
## FR:Cat:AR:Temp        1      0       0     NaN    NaN
## FR:Cat:AR:Conc        1     18      18     NaN    NaN
## FR:Cat:Temp:Conc      1      3       3     NaN    NaN
## FR:AR:Temp:Conc       1      8       8     NaN    NaN
## FR:Cat:AR:Temp:Conc   1      2       2     NaN    NaN
## Residuals             0      0     NaN
```

Estimates of the clear factorial effects can be obtained as twice the corresponding

Table 5.16: Reactor experiment in blocks.

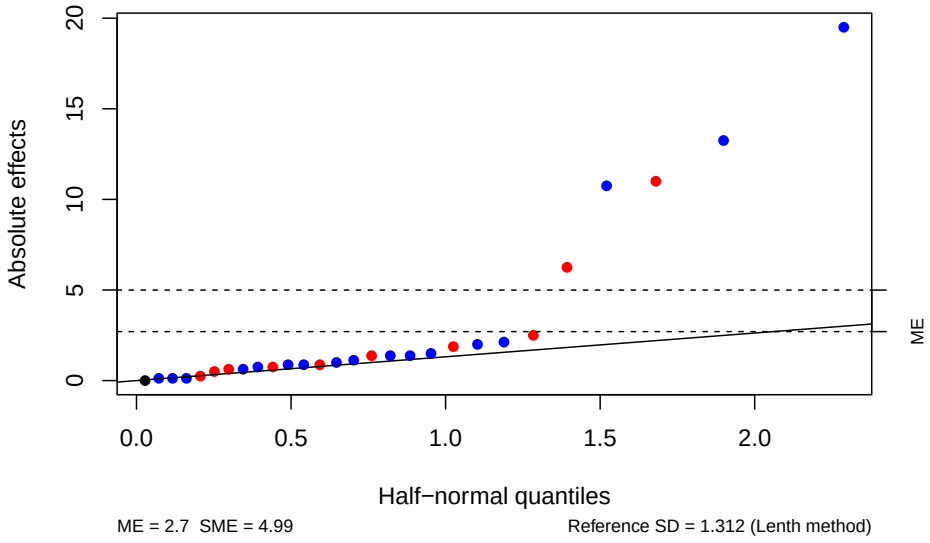
| block | FR | Cat | AR | Temp | Conc | pre.react |
|---|---|---|---|---|---|---|
| 1 | -1 | -1 | -1 | -1 | -1 | 61 |
| 1 | -1 | 1 | 1 | -1 | -1 | 54 |
| 1 | 1 | 1 | -1 | 1 | -1 | 93 |
| 1 | 1 | -1 | 1 | 1 | -1 | 60 |
| 1 | 1 | 1 | -1 | -1 | 1 | 65 |
| 1 | 1 | -1 | 1 | -1 | 1 | 55 |
| 1 | -1 | -1 | -1 | 1 | 1 | 44 |
| 1 | -1 | 1 | 1 | 1 | 1 | 81 |
| 2 | 1 | 1 | -1 | -1 | -1 | 61 |
| 2 | 1 | -1 | 1 | -1 | -1 | 56 |
| 2 | -1 | -1 | -1 | 1 | -1 | 69 |
| 2 | -1 | 1 | 1 | 1 | -1 | 95 |
| 2 | -1 | -1 | -1 | -1 | 1 | 56 |
| 2 | -1 | 1 | 1 | -1 | 1 | 67 |
| 2 | 1 | 1 | -1 | 1 | 1 | 77 |
| 2 | 1 | -1 | 1 | 1 | 1 | 42 |
| 3 | -1 | 1 | -1 | -1 | -1 | 63 |
| 3 | -1 | -1 | 1 | -1 | -1 | 53 |
| 3 | 1 | -1 | -1 | 1 | -1 | 61 |
| 3 | 1 | 1 | 1 | 1 | -1 | 98 |
| 3 | 1 | -1 | -1 | -1 | 1 | 63 |
| 3 | 1 | 1 | 1 | -1 | 1 | 65 |
| 3 | -1 | 1 | -1 | 1 | 1 | 78 |
| 3 | -1 | -1 | 1 | 1 | 1 | 49 |
| 4 | 1 | -1 | -1 | -1 | -1 | 53 |
| 4 | 1 | 1 | 1 | -1 | -1 | 61 |
| 4 | -1 | 1 | -1 | 1 | -1 | 94 |
| 4 | -1 | -1 | 1 | 1 | -1 | 66 |
| 4 | -1 | 1 | -1 | -1 | 1 | 70 |
| 4 | -1 | -1 | 1 | -1 | 1 | 59 |
| 4 | 1 | -1 | -1 | 1 | 1 | 45 |
| 4 | 1 | 1 | 1 | 1 | 1 | 82 |

regression coefficients. They are unchanged from the analysis of the unblocked design, as they are estimated independently of the blocks.

```
fac.effs <- data.frame(fac.effs = 2 * coef(reactor.lm)[-(1:4)])
fac.effs <- tidyr::drop_na(fac.effs)
rownames(fac.effs) <- rownames(anova(reactor.lm))[-c(1, 30)]
knitr::kable(fac.effs, caption = "Reactor experiment: estimates of clear factorial effe
```

Using Lenth's method produces very similar results to before, albeit using three fewer factorial effects (those confounded with blocks), as the pseudo-standard error hasn't changed[7]

```
unrepx::hnplot(fac.effs$fac.effs, horiz = F, method = "Lenth", alpha = 0.05)
```




[7] As the median of the estimated factorial effects is still 0.375.

Table 5.17: Reactor experiment: estimates of clear factorial effects

| | Estimate |
|---|---|
| FR | -1.375 |
| Cat | 19.500 |
| AR | -0.625 |
| Temp | 10.750 |
| Conc | -6.250 |
| FR:Cat | 1.375 |
| FR:AR | 0.750 |
| FR:Temp | -0.875 |
| FR:Conc | 0.125 |
| Cat:AR | 0.875 |
| Cat:Temp | 13.250 |
| Cat:Conc | 2.000 |
| AR:Temp | 2.125 |
| AR:Conc | 0.875 |
| Temp:Conc | -11.000 |
| FR:Cat:Temp | 1.375 |
| FR:Cat:Conc | -1.875 |
| FR:AR:Temp | -0.750 |
| FR:AR:Conc | -2.500 |
| Cat:AR:Temp | 1.125 |
| Cat:AR:Conc | 0.125 |
| Cat:Temp:Conc | -0.250 |
| AR:Temp:Conc | 0.125 |
| FR:Cat:AR:Temp | 0.000 |
| FR:Cat:AR:Conc | 1.500 |
| FR:Cat:Temp:Conc | 0.625 |
| FR:AR:Temp:Conc | 1.000 |
| FR:Cat:AR:Temp:Conc | -0.500 |

# Chapter 6

# Fractional factorial designs

The factorial designs we studied in Chapters 4 and 4 can involve a large number of treatments, for even a moderate number of factors (Table 6.1).

```
size <- data.frame(1:15, 2^(1:15))
knitr::kable(size, col.names = c("No. factors", "No. of trts"), caption = "Number of treatments i
```

For larger numbers of factors, resource constraints may mean it is not possible to run an experiment using all $2^f$ treatments. Also, many degrees of freedom in these experiments are used to estimate high-order interactions. For example, in a $2^5$ experiment, 16 degrees of freedom are used to estimate three-factor and higher interactions, half the size of the experiment. The principles of effect hierarchy and sparsity (Section 4.2) suggest this is probably wasteful.

We can select smaller experiments by using a subset, or **fraction** of the treatments of size $2^{f-q}$:

a. divide the treatments in subsets by confounding $q$ factorial effects (and their products), as in blocking;

b. only use **one** of the subsets in the experiment.

**Example 6.1.** Spring experiment (Wu and Hamada, 2009, ch. 5)

Consider an industrial experiment to investigate the effect of $f = 5$ factors on the unloaded height of a spring produced using a heat treatment. The five factors are described in Table 6.2.

```
factor.name <- c("Quench temperature (F)", "Heat temperature (F)", "Heating time (s)",
                 "Transfer time (s)", "Hold-down time (s)")
low.level <- c("130-150", 1840, 23, 10, 2)
high.level <- c("150-170", 1880, 25, 12, 3)
spring.factors <- data.frame(factor = factor.name, low = low.level, high = high.level)
row.names(spring.factors) <- LETTERS[1:5]
```

Table 6.1: Number of treatments in a $2^f$ factorial designs for different numbers, $f$, of factors.

| No. factors | No. of trts |
|---|---|
| 1 | 2 |
| 2 | 4 |
| 3 | 8 |
| 4 | 16 |
| 5 | 32 |
| 6 | 64 |
| 7 | 128 |
| 8 | 256 |
| 9 | 512 |
| 10 | 1024 |
| 11 | 2048 |
| 12 | 4096 |
| 13 | 8192 |
| 14 | 16384 |
| 15 | 32768 |

Table 6.2: Spring experiment: factors and levels

| | Factor | Low level | High level |
|---|---|---|---|
| A | Quench temperature (F) | 130-150 | 150-170 |
| B | Heat temperature (F) | 1840 | 1880 |
| C | Heating time (s) | 23 | 25 |
| D | Transfer time (s) | 10 | 12 |
| E | Hold-down time (s) | 2 | 3 |

```
knitr::kable(spring.factors, col.names = c("Factor", "Low level", "High level"),
             align = c("l", "r", "r"), caption = "Spring experiment: factors and levels
```

Enough experimental units were available to perform $n = 16$ runs, which is one-half of the total number of treatments. We refer to this type of design as a **one-half fractional replicate** of the full factorial design, or a $2^{5-1}$ **fractional factorial design**[1].

The design was constructed by confounding $q = 1$ factorial effects with blocks, the interaction $BCDE$ was chosen, and running just one of the two resulting subsets, see Table 6.3 where `FrF2` is used to generate the design.

[1] If we only run one-half of the treatments from a $2^5$ design, the design contains $\frac{2^5}{2} = 2^{5-1}$ treatments

Table 6.3: Spring experiment: 16 run design.

| A | B | C | D | E | height |
|---|---|---|---|---|---|
| -1 | -1 | -1 | -1 | -1 | 7.54 |
| 1 | -1 | -1 | -1 | -1 | 7.20 |
| -1 | 1 | -1 | -1 | 1 | 7.69 |
| 1 | 1 | -1 | -1 | 1 | 7.63 |
| -1 | -1 | 1 | -1 | 1 | 7.94 |
| 1 | -1 | 1 | -1 | 1 | 7.40 |
| -1 | 1 | 1 | -1 | -1 | 7.95 |
| 1 | 1 | 1 | -1 | -1 | 7.62 |
| -1 | -1 | -1 | 1 | 1 | 7.52 |
| 1 | -1 | -1 | 1 | 1 | 7.52 |
| -1 | 1 | -1 | 1 | -1 | 7.63 |
| 1 | 1 | -1 | 1 | -1 | 7.65 |
| -1 | -1 | 1 | 1 | -1 | 7.79 |
| 1 | -1 | 1 | 1 | -1 | 7.29 |
| -1 | 1 | 1 | 1 | 1 | 8.07 |
| 1 | 1 | 1 | 1 | 1 | 7.73 |

```
spring <- FrF2::FrF2(nruns = 16, nfactors = 5, generators = "BCD", randomize = F)
spring$height <- c(7.54, 7.20, 7.69, 7.63, 7.94, 7.40, 7.95, 7.62, 7.52, 7.52,
                   7.63, 7.65, 7.79, 7.29, 8.07, 7.73)
knitr::kable(spring, caption = "Spring experiment: 16 run design.", align = rep("r", 6))
```

Clearly, using a subset of the treatments, we will no longer be able to estimate all the factorial effects (we have insufficient degrees of freedom). We have confounded the interaction $BCDE$, and hence clearly the contrast coefficients for this effect will be constant in our design. We say the interaction $BCDE$ is **aliased** with the mean, and we write this as $I = BCDE$. This expression is called the **defining relation**, as knowledge of which factorial effects are aliased with the mean completely define the fractional factorial.

```
fac_to_numeric <- function(x) as.numeric(as.character(x))
BCDE <- fac_to_numeric(spring$B) * fac_to_numeric(spring$C) *
  fac_to_numeric(spring$D) * fac_to_numeric(spring$E)
BCDE
```

```
##  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

This removes one factorial effect from consideration, but we are still short on degrees of freedom. What are the other consequences of using a fractional factorial design?

As the contrast coefficients for the interaction $BCDE$ are constant, the contrast coefficients for any pairs of factorial effects whose (hadamard) product form

$BCDE$ will be equal. For example, the contrast coefficient vectors for interactions $BC$ and $DE$ will be equal, as will the vectors for the main effect $B$ and the interaction $CDE$, and so on.

```
BC <- fac_to_numeric(spring$B) * fac_to_numeric(spring$C)
DE <- fac_to_numeric(spring$D) * fac_to_numeric(spring$E)
BC
DE
all.equal(BC, DE)
```

```
##  [1]  1  1 -1 -1 -1 -1  1  1  1  1 -1 -1 -1 -1  1  1
##  [1]  1  1 -1 -1 -1 -1  1  1  1  1 -1 -1 -1 -1  1  1
## [1] TRUE
```

```
B <- fac_to_numeric(spring$B)
CDE <- fac_to_numeric(spring$C) * fac_to_numeric(spring$D) * fac_to_numeric(spring$E)
B
CDE
all.equal(B, CDE)
```

```
##  [1] -1 -1  1  1 -1 -1  1  1 -1 -1  1  1 -1 -1  1  1
##  [1] -1 -1  1  1 -1 -1  1  1 -1 -1  1  1 -1 -1  1  1
## [1] TRUE
```

We say these factorial effects are **aliased**. From the defining relation, we can derive the complete **aliasing scheme** for a fractional factorial design. For the example,

$$I = BCDE \tag{6.1}$$
$$A = ABCDE \tag{6.2}$$
$$B = CDE \tag{6.3}$$
$$C = BDE \tag{6.4}$$
$$D = BCE \tag{6.5}$$
$$E = BCD \tag{6.6}$$
$$AB = ACDE \tag{6.7}$$
$$AC = ABDE \tag{6.8}$$
$$AD = ABCE \tag{6.9}$$
$$AE = ABCD \tag{6.10}$$
$$BC = DE \tag{6.11}$$
$$BD = CE \tag{6.12}$$
$$BE = CD \tag{6.13}$$
$$ABC = ADE \tag{6.14}$$
$$ABD = ACE \tag{6.15}$$
$$ABE = ACD \tag{6.16}$$
$$\tag{6.17}$$

The aliasing scheme contains $2^{f-q} = 2^{5-1} = 16$ "strings", each one containing $2^q = 2^1 = 2$ "words". The design is not capable to distinguishing between factorial effects in the same alias string.

We can also generate this information using the `aliases` function from `FrF2`.

```
spring.lm <- lm(height ~ (.)^5, data = spring)
FrF2::aliases(spring.lm)
```

```
##
##  A = A:B:C:D:E
##  B = C:D:E
##  C = B:D:E
##  D = B:C:E
##  E = B:C:D
##  A:B = A:C:D:E
##  A:C = A:B:D:E
##  A:D = A:B:C:E
##  A:E = A:B:C:D
##  B:C = D:E
##  B:D = C:E
##  B:E = C:D
##  A:B:C = A:D:E
```

```
##  A:B:D = A:C:E
##  A:B:E = A:C:D
```

**Definition 6.1.** A **regular $2^{f-q}$fractional factorial design** is constructed by aliasing $2^q - 1$ factorial effects with the mean; $q$ of these effects can be chosen independently, the others are formed via the hadamard product of the contrast coefficients for the $q$ effects,

How do we chose the factorial effects to alias with the mean? As with blocking, we tend to choose higher-order effects, taking care when $q > 1$ not to inadvertently alias together lower-order effects (see later examples).

For Example 6.1, a slightly unusual defining relation was chosen. It would be more common to use $I = ABCDE$, leading to the aliasing scheme:

$$I = ABCDE \tag{6.18}$$
$$A = BCDE \tag{6.19}$$
$$B = ACDE \tag{6.20}$$
$$C = ABDE \tag{6.21}$$
$$D = ABCE \tag{6.22}$$
$$E = ABCD \tag{6.23}$$
$$AB = CDE \tag{6.24}$$
$$AC = BDE \tag{6.25}$$
$$AD = BCE \tag{6.26}$$
$$AE = BCD \tag{6.27}$$
$$BC = ADE \tag{6.28}$$
$$BD = ACE \tag{6.29}$$
$$BE = ACD \tag{6.30}$$
$$CD = ABE \tag{6.31}$$
$$CE = ABD \tag{6.32}$$
$$DE = ABC \tag{6.33}$$
$$\tag{6.34}$$

This defining relation results in main effects being aliased with four-factor interactions and, perhaps more importantly, no pairs of two-factor interactions aliased together. The original design from Example **??** might be used if factor $A$ and its interactions were a priori thought likely to be important (two-factor interactions involving factor $A$ are aliased with four-factor interactions).

## 6.1 Estimability and aliasing

Any factorial effect in an alias string is only estimable **if all other effects in that string are assumed zero**[2]. Wd can study this further by introducing the **alias matrix**.

**Definition 6.2.** Assume a linear data generating model

$$\boldsymbol{y} = X_1\boldsymbol{\beta}_1 + X_2\boldsymbol{\beta}_2 + \boldsymbol{\varepsilon}\,,$$

where $\boldsymbol{y}$ is an $n$-vector of responses, $X_1$ and $X_2$ are $n \times p_1$ and $n \times p_2$ model matrices, respectively, with $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ corresponding $p_1$- and $p_2$-vectors of parameters and random errors $\varepsilon\ N(\mathbf{0}, I_n\sigma^2)$.

If the submodel

$$\boldsymbol{y} = X_1\boldsymbol{\beta}_1 + \boldsymbol{\varepsilon}\,,$$

is fitted to the response data, then $\hat{\boldsymbol{\beta}}_1 = (X_1^{\mathrm{T}}X_1)^{-1}X_1^{\mathrm{T}}\boldsymbol{y}$, and

$$\begin{aligned} E(\hat{\boldsymbol{\beta}}_1) &= \boldsymbol{\beta}_1 + (X_1^{\mathrm{T}}X_1)^{-1}X_1^{\mathrm{T}}X_2\boldsymbol{\beta}_2 \\ &= \boldsymbol{\beta}_1 + A\boldsymbol{\beta}_2\,, \end{aligned}$$

where $A = (X_1^{\mathrm{T}}X_1)^{-1}X_1^{\mathrm{T}}X_2$ is the **alias** matrix.

We also introduce an alternative definition of estimability.

**Definition 6.3.** A linear combination of parameters $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{\theta}$ is estimable if and only if there exists a linear combination of the responses $\boldsymbol{a}^{\mathrm{T}}\boldsymbol{y}$ such that

$$E(\boldsymbol{a}^{\mathrm{T}}\boldsymbol{y}) = c^{\mathrm{T}}\boldsymbol{\theta}\,.$$

Now assume that using a two-level fractional factorial design, we will estimate one factorial effect (equivalently, the corresponding regression coefficient) from each alias string. Then the $A$ matrix will have entries 0, -1 or +1, depending on the defining relation of the fraction. Each regression parameter will be biased by the parameters corresponding to other factorial effects in the alias string. Hence, by Definition 6.3, each factorial effect is only estimable under the assumption that all other factorial effects in the alias string are zero.

For Example 6.1 we can generate the alias matrix using the `alias` function.

```
t(alias(spring.lm)$Complete)
```

```
##             A1:C1:D1 A1:C1:E1 A1:D1:E1 B1:C1:D1 B1:C1:E1 B1:D1:E1 C1:D1:E1
## (Intercept) 0        0        0        0        0        0        0
## A1          0        0        0        0        0        0        0
```

[2]Except for the defining relation, where no effects are estimable

```
## B1          0        0        0        0        0        0        1
## C1          0        0        0        0        0        1        0
## D1          0        0        0        0        1        0        0
## E1          0        0        0        1        0        0        0
## A1:B1       0        0        0        0        0        0        0
## A1:C1       0        0        0        0        0        0        0
## A1:D1       0        0        0        0        0        0        0
## A1:E1       0        0        0        0        0        0        0
## B1:C1       0        0        0        0        0        0        0
## B1:D1       0        0        0        0        0        0        0
## B1:E1       0        0        0        0        0        0        0
## A1:B1:C1    0        0        1        0        0        0        0
## A1:B1:D1    0        1        0        0        0        0        0
## A1:B1:E1    1        0        0        0        0        0        0
##             A1:B1:C1:D1 A1:B1:C1:E1 A1:B1:D1:E1 A1:C1:D1:E1 B1:C1:D1:E1
## (Intercept) 0           0           0           0           1
## A1          0           0           0           0           0
## B1          0           0           0           0           0
## C1          0           0           0           0           0
## D1          0           0           0           0           0
## E1          0           0           0           0           0
## A1:B1       0           0           0           1           0
## A1:C1       0           0           1           0           0
## A1:D1       0           1           0           0           0
## A1:E1       1           0           0           0           0
## B1:C1       0           0           0           0           0
## B1:D1       0           0           0           0           0
## B1:E1       0           0           0           0           0
## A1:B1:C1    0           0           0           0           0
## A1:B1:D1    0           0           0           0           0
## A1:B1:E1    0           0           0           0           0
##             A1:B1:C1:D1:E1 C1:D1 C1:E1 D1:E1
## (Intercept) 0              0     0     0
## A1          1              0     0     0
## B1          0              0     0     0
## C1          0              0     0     0
## D1          0              0     0     0
## E1          0              0     0     0
## A1:B1       0              0     0     0
## A1:C1       0              0     0     0
## A1:D1       0              0     0     0
## A1:E1       0              0     0     0
## B1:C1       0              0     0     1
## B1:D1       0              0     1     0
## B1:E1       0              1     0     0
## A1:B1:C1    0              0     0     0
```

```
## A1:B1:D1    0             0     0     0
## A1:B1:E1    0             0     0     0
```

## 6.2 General method for choosing a fractional factorial design

To select a $2^{f-q}$ fractional factorial design:

- choose $q$ independent factorial contrasts for the **generators**, or **defining words**, $\nu_1, \ldots, n_q$. Typically, we choose higher-order interactions (due to effect hierarchy):

$$\nu_1 = \boldsymbol{c}_1, \ldots, \nu_q = \boldsymbol{c}_q \,.$$

- all the hadamard products of $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_q$ are also aliased with the mean, and together give the **defining relation**:

$$I = \nu_1 = \ldots = \nu_q = \nu_1\nu_q = \ldots = \nu_1 \cdots \nu_q \,.$$

- the **aliasing scheme** is a list of the $2^{f-q}$ **alias strings**. Every effect in one string is estimated by the same contrast in the observations:

$$\begin{array}{rcccccccccc}
I & = & \nu_1 & = & \ldots & = & \nu_q & = & \nu_1\nu_q & = & \ldots & = & \nu_1 \cdots \nu_q \\
A & = & A\nu_1 & = & \ldots & = & A\nu_q & = & A\nu_1\nu_q & = & \ldots & = & A\nu_1 \cdots \nu_q \\
B & = & B\nu_1 & = & B\ldots & = & B\nu_q & = & B\nu_1\nu_q & = & \ldots & = & B\nu_1 \cdots \nu_q \\
\vdots & & & & & & & & & & & & \\
CD & = & CD\nu_1 & = & CD\ldots & = & CD\nu_q & = & CD\nu_1\nu_q & = & CD\ldots & = & CD\nu_1 \cdots \nu_q \\
\vdots & & & & & & & & & & & &
\end{array}$$

All the effects within a single alias string are aliased, and cannot be simultaneously aliased. In fact, the estimation of any effect within an alias string can only proceed if all other effects in the string are assumed zero; otherwise, we can estimate the sum of the effects (see Section 6.1).

To find the treatments in a particular fraction, we simply need to solve a set of equations of the form

$$\nu_1 = \pm 1, \quad \nu_2 + \pm 1, \quad \ldots \quad \nu_q = \pm 1 \,,$$

with $2^{f-q}$ treatments satisfying the $2^q$ equations corresponding to each choice of $\pm 1$ for each generator.

**Example 6.2.** Consider a $2^{6-2}$ design, with $q = 2$ generators $ABCE$ and $BCDF$. The defining relation is given by

$$I = ABCE = BCDF = ADEF\,.$$

The aliasing scheme has $2^{6-2} = 16$ strings, each of which contains $2^2 = 4$ effects or words.

$$\begin{array}{rcrcrcr}
I & = & ABCE & = & BCDF & = & ADEF \\
A & = & BCE & = & ABCDF & = & DEF \\
B & = & ACE & = & CDF & = & ABDEF \\
C & = & ABE & = & BDF & = & ACDEF \\
D & = & ABCDE & = & BCF & = & AEF \\
E & = & ABC & = & BCDEF & = & ADF \\
F & = & ABCEF & = & BCD & = & ADE \\
AB & = & CE & = & ACDF & = & BDEF \\
AC & = & BE & = & ABDF & = & CDEF \\
AD & = & BCDE & = & ABCF & = & EF \\
AE & = & BC & = & ABCDEF & = & DF \\
AF & = & BCEF & = & ABCD & = & DE \\
BD & = & ACDE & = & CF & = & ABEF \\
BF & = & ACEF & = & CD & = & ABDE \\
ABD & = & CDE & = & ACF & = & BEF \\
ABF & = & CEF & = & ACD & = & BDE
\end{array}$$

This aliasing scheme can result from any one of the four possible fractions defined by

$$ABCE = \pm 1\,, \quad BCDF = \pm 1\,.$$

Each fraction has (essentially) the same statistical properties, except the sign of the biasing coefficients from the alias matrix may be reversed (-1, rather than +1).

For example, the treatments in the fraction may be those $2^{f-q} = 2^{6-2} = 16$ treatments that all have $ABCE = +1$ and $BCDF = +1$.

We can also use `FrF2` to find two-level fractional factorial designs, as we already saw in Example 6.1. The `generators` argument can be used to specify which factorial effects to use as generators. These are specified using what `FrF2` refers to as the $f - q$ **base factors**. The $q$ generators must specify with which factorial effects in the base factors the $q$ additional factors are aliased.
For Example 6.2, a $2^{6-2}$ design, the base factors are $A, B, C, D$ and from our defining relation we have $E = ABC$ and $F = BCD$.

```
ff.2.6.2 <- FrF2::FrF2(nruns = 16, nfactors = 6, generators = c("ABC", "BCD"),
                       randomize = F, alias.info = 3)
```

Once `FrF2` has generated the design, we can interrogate the aliasing (up to three-factor interactions) using `design.info` (and using the `aliases` function, see Example 6.1).

Table 6.4: Treatments from a $2^{6-2}$ fractional factorial design.

| A | B | C | D | E | F |
|---|---|---|---|---|---|
| -1 | -1 | -1 | -1 | -1 | -1 |
| 1 | -1 | -1 | -1 | 1 | -1 |
| -1 | 1 | -1 | -1 | 1 | 1 |
| 1 | 1 | -1 | -1 | -1 | 1 |
| -1 | -1 | 1 | -1 | 1 | 1 |
| 1 | -1 | 1 | -1 | -1 | 1 |
| -1 | 1 | 1 | -1 | -1 | -1 |
| 1 | 1 | 1 | -1 | 1 | -1 |
| -1 | -1 | -1 | 1 | -1 | 1 |
| 1 | -1 | -1 | 1 | 1 | 1 |
| -1 | 1 | -1 | 1 | 1 | -1 |
| 1 | 1 | -1 | 1 | -1 | -1 |
| -1 | -1 | 1 | 1 | 1 | -1 |
| 1 | -1 | 1 | 1 | -1 | -1 |
| -1 | 1 | 1 | 1 | -1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 |

```
library(FrF2)
design.info(ff.2.6.2)$aliased
```

```
## $legend
## [1] "A=A" "B=B" "C=C" "D=D" "E=E" "F=F"
##
## $main
## [1] "A=BCE=DEF" "B=ACE=CDF" "C=ABE=BDF" "D=AEF=BCF" "E=ABC=ADF" "F=ADE=BCD"
##
## $fi2
## [1] "AB=CE"    "AC=BE"    "AD=EF"    "AE=BC=DF" "AF=DE"    "BD=CF"    "BF=CD"
##
## $fi3
## [1] "ABD=ACF=BEF=CDE" "ABF=ACD=BDE=CEF"
```

By default, `FrF2` chooses the fraction defined by each generator being equal to $+1$.

```
knitr::kable(ff.2.6.2, align = "r",
             caption = "Treatments from a $2^{6-2}$ fractional factorial design.")
```

## 6.3 Resolution and aberration

Clearly, the numbers of factors involved in the effects in the defining relation play an important role in the properties of the design.

**Definition 6.4.** Each factorial effect is commonly referred to as a **word**, and the number of factors involved in a factorial effect is its **length**.

For example, word $ABCD$ has length 4; word $BDE$ has length 3.

**Definition 6.5.** The **resolution** of a $2^{f-q}$ design is the length of the shortest word in the defining relation.

A design having resolution $R$ implies that no effect involving $x$ factors is aliased with effects involving less than $R - x$ factors.

Designs of the following resolution are particularly common.

- *Resolution III*: shortest word of length 3. No main effect is aliased with any other main effect; at least one main effect of aliased with a two-factor interaction.
- *Resolution IV*: shortest word of length 4. No main effect is aliased with any other main effect or any two-factor interaction; at least one pair of two-factor interactions are aliased together.
- *Resolution V*: shortest word of length 5. No main effect or two-factor interaction is aliased with any other main effect or two-factor interaction.

For example,

- $2^{6-2}$, $I = ABCD = CDEF = ABEF$: resolution IV.
- $2^{3-1}$, $I = ABC$: resolution III.

**Definition 6.6.** The *word length pattern* of a $2^{f-q}$ design is given by

$$W = (w_3, w_4, \ldots, w_f),$$

where $w_i$ is the number of words of length $i$ in the defining relation.

For example,

- $d_1$: $2^{7-2}$, $I = ABCF = ADEG = BCDEFG$
- $d_2$: $2^{7-2}$, $I = DEFG = ABCDF = ABCEG$

are both resolution IV designs, but have different word length patterns

- $W(d_1) = (0, 2, 0, 1, 0)$
- $W(d_2) = (0, 1, 2, 0, 0)$.

**Definition 6.7.** For two $2^{f-q}$ designs, say $d^\star$ and $d^\dagger$, let $r$ be the smallest integer such that $w_r(d^\star) \neq w_r(d^\dagger)$. Then design $d^\star$ is said to have less **aberration** than design $d^\dagger$ if

$$w_r(d^\star) < w_r(d^\dagger)\,.$$

If no design has less aberration than $d^\star$, then $d^\star$ has **minimum aberration**.

Consider again designs $d_1$ and $d_2$ from above ($2^{7-2}$ fractions). Here,

$$\begin{array}{rcccrcl} w_3(d_1) & = & 0 & = & w_3(d_2) & = & 0 \\ w_4(d_1) & = & 2 & > & w_4(d_2) & = & 1\,, \end{array}$$

and hence $d_2$ has less aberration than $d_1$. In fact, $d_2$ has minimum aberration.

We can use `FrF2` to find designs of a specific resolution using the `resolution` argument (and leaving `nruns` and `generators` unspecified). The resulting design will have the minimum number of runs required to obtain the requested resolution.

```
ff.2.6.2.r <- FrF2::FrF2(nfactors = 6, resolution = 4,
                         randomize = F, alias.info = 3)
design.info(ff.2.6.2.r)$aliased
```

```
## $legend
## [1] "A=A" "B=B" "C=C" "D=D" "E=E" "F=F"
##
## $main
## [1] "A=BCE=BDF" "B=ACE=ADF" "C=ABE=DEF" "D=ABF=CEF" "E=ABC=CDF" "F=ABD=CDE"
##
## $fi2
## [1] "AB=CE=DF" "AC=BE"    "AD=BF"    "AE=BC"    "AF=BD"    "CD=EF"    "CF=DE"
##
## $fi3
## [1] "ACD=AEF=BCF=BDE" "ACF=ADE=BCD=BEF"
```

When `generators` and `resolution` are not specified, `FrF2` function selects designs from catalogues of good designs, most of which have minimum aberration.

```
ff.2.6.2.a <- FrF2::FrF2(nfactors = 6, nruns = 16,
                         randomize = F, alias.info = 3)
design.info(ff.2.6.2.a)$aliased
```

```
## $legend
## [1] "A=A" "B=B" "C=C" "D=D" "E=E" "F=F"
##
## $main
## [1] "A=BCE=BDF" "B=ACE=ADF" "C=ABE=DEF" "D=ABF=CEF" "E=ABC=CDF" "F=ABD=CDE"
##
## $fi2
## [1] "AB=CE=DF" "AC=BE"    "AD=BF"    "AE=BC"    "AF=BD"    "CD=EF"    "CF=DE"
##
## $fi3
```

```
## [1] "ACD=AEF=BCF=BDE" "ACF=ADE=BCD=BEF"
```

## 6.4 Analysis of fractional factorial designs

The analysis can proceed as for full factorial designs (Chapter 4). Assuming only one factorial effect in each alias string is non-zero, we can estimate $2^{f-q}-1$ factorial effects (one from each string) either by fitting the unit-treatment model or the corresponding regression model.

For Example 6.1, we can use the unit-treatment model and contrasts to estimate all main effects and selected two-factor interactions, assuming all other factorial effects are zero.

```
spring$treatment <- factor(1:16)
spring.ut <- lm(height ~ treatment, data = spring)
fac.contrasts.emmc <- function(nlevs) {
  spring.num <- apply(spring[, c("A", "B", "C", "D", "E")], 2, fac_to_numeric)
  data.frame(model.matrix(~ . + A:B + A:C + A:D + A:E + B:C
                          + B:D + B:E, data.frame(spring.num))[, -1] / 8)
}
spring.emm <- emmeans::emmeans(spring.ut, ~ treatment)
emmeans::contrast(spring.emm, 'fac.contrasts')
```

```
##  contrast estimate  SE df t.ratio p.value
##  A         -0.2612 NaN  0     NaN     NaN
##  B          0.2213 NaN  0     NaN     NaN
##  C          0.1762 NaN  0     NaN     NaN
##  D          0.0288 NaN  0     NaN     NaN
##  E          0.1037 NaN  0     NaN     NaN
##  A.B        0.0838 NaN  0     NaN     NaN
##  A.C       -0.1663 NaN  0     NaN     NaN
##  A.D        0.0563 NaN  0     NaN     NaN
##  A.E        0.0262 NaN  0     NaN     NaN
##  B.C        0.0163 NaN  0     NaN     NaN
##  B.D        0.0187 NaN  0     NaN     NaN
##  B.E       -0.0362 NaN  0     NaN     NaN
```

When looking at this output, we must remember the aliasing scheme and recognise that we can only estimate these effects if all of their alaises are zero. Otherwise, each of these factorial effects is biased and not estimable (we are actually estimating the linear combination of all the aliased effects).

Alternatively, we can fit the regression model directly in the contrasts; `lm` automatically recognises which pairs of effects are aliased, and chooses the lexicographically first effect to include in the model.

```
spring.lm <- lm(height ~ (A + B + C + D + E) ^ 5, data = spring)
c(na.omit(2 * coef(spring.lm)[-1]))
```

```
##       A1       B1       C1       D1       E1    A1:B1    A1:C1    A1:D1
## -0.26125  0.22125  0.17625  0.02875  0.10375  0.08375 -0.16625  0.05625
##    A1:E1    B1:C1    B1:D1    B1:E1 A1:B1:C1 A1:B1:D1 A1:B1:E1
##  0.02625  0.01625  0.01875 -0.03625  0.00875 -0.03875 -0.04875
```

In this experiment, there are three alias string that only include three-factor interactions. We may wish to use these three degrees of freedom to estimate $\sigma^2$, under the assumption that all six interactions involved in these three strings are zero.

```
spring.lm <- lm(height ~ (A + B + C + D + E) ^ 2, data = spring)
anova(spring.lm)
```

```
## Analysis of Variance Table
##
## Response: height
##           Df Sum Sq Mean Sq F value Pr(>F)
## A          1 0.2730  0.2730   51.78 0.0055 **
## B          1 0.1958  0.1958   37.13 0.0089 **
## C          1 0.1243  0.1243   23.56 0.0167 *
## D          1 0.0033  0.0033    0.63 0.4863
## E          1 0.0431  0.0431    8.17 0.0647 .
## A:B        1 0.0281  0.0281    5.32 0.1043
## A:C        1 0.1106  0.1106   20.97 0.0196 *
## A:D        1 0.0127  0.0127    2.40 0.2191
## A:E        1 0.0028  0.0028    0.52 0.5220
## B:C        1 0.0011  0.0011    0.20 0.6848
## B:D        1 0.0014  0.0014    0.27 0.6412
## B:E        1 0.0053  0.0053    1.00 0.3917
## Residuals  3 0.0158  0.0053
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

## 6.5 Blocking fractional factorial designs

We block fractional factorial designs using the same approach as Chapter 5. To block a $2^{f-q}$ design into $b = 2^m$ blocks, we choose $m$ factorial effects to confound with blocks, also confounding all products of these $m$ effects. However, we of course also confound the $2^q - 1$ aliases of each of these $2^m - 1$ effects. Hence, we actually choose to confound $2^m - 1$ *strings of factorial effects* with blocks.

We must pay particular attention to make sure we do not accidentically confound a lower-order effect through the aliasing scheme. We can also use `FrF2` to find blocked factorial designs.

**Example 6.3.** Consider a $2^{6-2}$ design with defining relation

$$I = ABCE = ABDF = CDEF\,.$$

To split this design into $b = 2^2 = 4$ blocks of size $k = 4$, we choose $m = 2$ defining blocks, and also confound their product.

$$\begin{aligned} \text{Block}_1 &= ACD \\ \text{Block}_2 &= BCD \\ \text{Block}_1\text{Block}_2 &= AB\,. \end{aligned}$$

We also confound all aliases of these effects:

$$\begin{aligned} \text{Block}_1 &= ACD = BDE = BCF = ADEF \\ \text{Block}_2 &= BCD = ADE = ACF = BEF \\ \text{Block}_1\text{Block}_2 &= AB = CE = DF = ABCDEF\,. \end{aligned}$$

We can also find this design using `FrF2`, combining the arguments `generators` and `blocks`.

```
ff.2.6.2.b.4 <- FrF2::FrF2(nruns = 16, nfactors = 6, generators = c("ABC", "ABD"), bloc
                           randomize = F, alias.block.2fis = T, alias.info = 3)
design.info(ff.2.6.2.b.4)$aliased
```

```
## $legend
## [1] "A=A" "B=B" "C=C" "D=D" "E=E" "F=F"
##
## $main
## [1] "A=BCE=BDF" "B=ACE=ADF" "C=ABE=DEF" "D=ABF=CEF" "E=ABC=CDF" "F=ABD=CDE"
##
## $fi2
## [1] "AC=BE" "AD=BF" "AE=BC" "AF=BD" "CD=EF" "CF=DE"
##
## $fi3
## character(0)
```

```
design.info(ff.2.6.2.b.4)$aliased.with.blocks
```

```
##  [1] "AB"  "CE"  "DF"  "ACD" "ACF" "ADE" "AEF" "BCD" "BCF" "BDE" "BEF"
```

One block of this design can be found by finding all the treatment combinations that satisfy, for example,

$$ABCE = +1, ABDF = +1, ACD = -1, BCD = -1\,.$$

Fixing values for $ABCE$ and $ABDF$, the other blocks are formed from the other 3 combinations of each of $ACD$ and $BCD$ being equal to $\pm 1$. We can also use `FrF2`.

Table 6.5: Fractional factorial $2^{6-2}$ design in $b = 4$ blocks of size $k = 4$.

| Blocks | A | B | C | D | E | F | Blocks | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | -1 | -1 | -1 | -1 | -1 | -1 | 3 | -1 | 1 | -1 | 1 | 1 | -1 |
| 1 | -1 | -1 | 1 | 1 | 1 | 1 | 3 | -1 | 1 | 1 | -1 | -1 | 1 |
| 1 | 1 | 1 | -1 | 1 | -1 | 1 | 3 | 1 | -1 | -1 | -1 | 1 | 1 |
| 1 | 1 | 1 | 1 | -1 | 1 | -1 | 3 | 1 | -1 | 1 | 1 | -1 | -1 |
| 2 | -1 | 1 | -1 | -1 | 1 | 1 | 4 | -1 | -1 | -1 | 1 | -1 | 1 |
| 2 | -1 | 1 | 1 | 1 | -1 | -1 | 4 | -1 | -1 | 1 | -1 | 1 | -1 |
| 2 | 1 | -1 | -1 | 1 | 1 | -1 | 4 | 1 | 1 | -1 | -1 | -1 | -1 |
| 2 | 1 | -1 | 1 | -1 | -1 | 1 | 4 | 1 | 1 | 1 | 1 | 1 | 1 |

```
block12 <- ff.2.6.2.b.4[1:8, ]
block34 <- ff.2.6.2.b.4[9:16, ]
knitr::kable(cbind(block12, block34),
      caption = "Fractional factorial
      $2^{6-2}$ design in $b=4$ blocks of size $k=4$.",
      align = "r")
```

Analysis proceeds as before, except no factorial effect can be estimated that is within an alias string that is confounded with blocks. For Example, split the spring experiment from Exa,ple 6.1 into $b = 2$ blocks of size $k = 8$ using confounding the alias string $ABE = ACD$ with blocks.

```
spring.blocks <- spring
spring.blocks$blocks <- with(data.frame(spring), fac_to_numeric(A) * fac_to_numeric(B) * fac_to_n
springb.lm <- lm(height ~ blocks + (A + B + C + D + E) ^ 3, data = spring.blocks)
anova(springb.lm)
```

```
## Analysis of Variance Table
##
## Response: height
##           Df Sum Sq Mean Sq F value Pr(>F)
## blocks     1 0.0003  0.0003     NaN    NaN
## A          1 0.2730  0.2730     NaN    NaN
## B          1 0.1958  0.1958     NaN    NaN
## C          1 0.1243  0.1243     NaN    NaN
## D          1 0.0033  0.0033     NaN    NaN
## E          1 0.0431  0.0431     NaN    NaN
## A:B        1 0.0281  0.0281     NaN    NaN
## A:C        1 0.1106  0.1106     NaN    NaN
## A:D        1 0.0127  0.0127     NaN    NaN
## A:E        1 0.0028  0.0028     NaN    NaN
## B:C        1 0.0011  0.0011     NaN    NaN
## B:D        1 0.0014  0.0014     NaN    NaN
```

```
## B:E         1 0.0053  0.0053     NaN    NaN
## A:B:D       1 0.0060  0.0060     NaN    NaN
## A:B:E       1 0.0095  0.0095     NaN    NaN
## Residuals   0 0.0000     NaN
```

## 6.6 Exercises

1. A $2^{5-2}$ design has generators $ACD$ and $BCE$.

   a. Write down the full defining relation. What resolution is this design?

   b. After analysis, factor $E$ turns out to be unimportant. By assuming that all effects involving factor $E$ and all three-factor and higher-order interactions are negligible, determine which two-factor interactions can be estimated together with the four main effects of factors $A$, $B$, $C$ and $D$.

Solution

a. The full defining relation is

$$I = ACD = BCE = ABDE\,.$$

The shortest word in the defining relation has length three, and therefore the design has resolution III.

b. Consider the full aliasing scheme:

$$\begin{array}{rcrcrcr} A & = & CD & = & ABCE & = & BDE \\ B & = & ABCD & = & CE & = & ADE \\ C & = & AD & = & BE & = & ABCDE \\ D & = & AC & = & BCDE & = & ABE \\ E & = & ACDE & = & BC & = & ABD \\ AB & = & BCD & = & ACE & = & DE \\ AE & = & CDE & = & ABC & = & BD \end{array}$$

If all the factorial effects involving factor $E$ and all three-factor and higher-order interactions are negligible, we can estimate the two-factor interactions $AB$, $BC$ and $BD$ in addition to the main effects of the first four factors.

2. Consider the following two fractional factorial designs.

   i. A $2^{6-2}$ design with generators $ABCDE$ and $ABDF$.
   ii. A $2^{6-2}$ design with generators $ABCE$ and $ABDF$.

   a. What is the resolution of each design?

   b. Which design would be preferred under the criterion of minimum aberration?

c. For design ii., if we know that any two-factor interaction involving factor $F$ is negligible, which other two-factor interactions can be estimated under the assumption that three-factor and higher-order interactions are also negligible?

Solution

a. Design i. has full defining relation $I = ABCDE = ABDF = CEF$, and hence is resolution III. Design ii. has full defining relation $I = ABCE = ABDF = CDEF$, and hence is resolution IV.

b. Design ii. would be preferred under the citerion of minimum aberration, as it is of higher resolution.

c. The alias strings from design ii. including two-factor interactions are as below:

$$\begin{array}{rcrcrcr} AB & = & CE & = & DF & = & ABCDEF \\ AC & = & BE & = & BCDF & = & ADEF \\ AD & = & BCDE & = & BF & = & ACEF \\ AE & = & BC & = & BDEF & = & ACDF \\ AF & = & BCEF & = & BD & = & ACDE \\ CD & = & ABDE & = & ABCF & = & EF \\ CF & = & ABEF & = & ABCD & = & DE \end{array}$$

It follows that if all two-factor interactions that involve factor $F$ are neglible, along with all three-factor and higher-order interactions, then we can estimate two-factor interactions $AD$, $BD$, $CD$ and $DE$. The pairs of interactions $AB = CE$, $AC = BE$ and $AE = BC$ are still aliased together.

3. a. Design an experiment with $n = 8$ runs to study the effect of the following five factors on yield:

- $A$: temperature (160°F or 180°F)
- $B$: concentration (30% or 40%)
- $C$: catalyst (1 or 2)
- $D$: stirring rate (60 or 100 rpm)
- $E$: pH (low or high).

It is known that the combinations temperature 180°F, concentration 40%, stirring rate 100 rpm and temperature 180°F, catalyst 2, pH high may lead to disastrous results and should be avoided. Write down the design in coded (-1, +1) units.

b. For the five factors in part (a), find a design with $n = 8$ runs such that the temperature-by-catalyst interaction ($AC$) and the concentration-by-catalyst ($BC$) interactions are neither aliased with main effects or with each other.

Solution

a. We require a $2^{5-2}$ fractional factorial design. We could choose to alias the $q = 2$ interactions $ABD$ and $ACE$ wth the mean, along with their product $BCDE$.

We want to avoid fractions with $A = +1$, $B = +1$ and $D = +1$, and $A = +1$, $C = +1$ and $E = +1$. These lead to $ABD = +1$ and $ACE = +1$, which together define one of the four possible fractions. Hence we just need top choose one of the three other choices, e.g. $ABD = -1$ and $ACE = -1$, giving

| Run | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | -1 | -1 | -1 | -1 | -1 |
| 2 | -1 | +1 | +1 | +1 | +1 |
| 3 | -1 | -1 | +1 | -1 | +1 |
| 4 | -1 | +1 | -1 | +1 | -1 |
| 5 | +1 | -1 | +1 | +1 | -1 |
| 6 | +1 | +1 | -1 | -1 | +1 |
| 7 | +1 | -1 | -1 | +1 | +1 |
| 8 | +1 | +1 | +1 | -1 | -1 |

b. We need a $2^{5-2}$ fractional factorial design such that $AC$ and $BC$ are not alised with main effects or each other. Therefore, our defining relation cannot contain any three letter words contain $AC$ or $BC$. A design with defining relation

$$I = ABDE = ABCE = CD$$

has this property, as $AC = BCDE = BE = AD$ and $BC = ACDE = AE = BD$. There are also other choices that have this property.

[To find this design, I started with the alias strings I wanted for $AC$ and $BC$ and then worked backwards.]

4. An experimenter who wishes to use a $2^{8-2}$ design can only do 16 runs per day, and would like to include "day" as a blocking factor. What design would you recommend and why? Give the defining relation of the fraction you choose, and the defining generators for the blocks. Which two-factor interactions can be clearly estimated?

Solution

The minimum aberration resolution V design has defining relation $I = ABCDG = ABEFGH = CDEFH$ (there are other similar possibilities).

There are 64 alias strings (including the defining relation) and we need to choose two to define our blocks (remembering the product will also be confounded). Two good choices are

$$ACE = BDEG = BCFGH = ADFH$$

and

$$BDF = ACFG = ADEGH = BCEH\,,$$

with product

$$ABCDEF = EFG = CDGH = ABH\,.$$

This choice leads to no two-factor interactions being confounded with blocks and hence, as the fraction is resolution V, all two-factor interactions are clear of main effects and other two-factor interactions.

# Bibliography


Box, G. E. P. and Meyer, R. D. (1986). An analysis of unreplicated fractional factorials. *Technometrics*, 28:11–18.

Cochran, W. G. and Cox, G. M. (1957). *Experimental Designs*. John Wiley and Sons, New York, 2nd edition.

Davies, O. L., editor (1954). *The Design and Analysis of Industrial Experiments*. Oliver and Boyd, London.

Fisher, R. A. (1935). *The Design of Experiments*. Oliver and Boyd, Edinburgh.

Fisher, R. A. and Yates, F. (1963). *Statistical Tables for Biological, Agricultural and Medical Research*. Oliver and Boyd, Edinburgh, 6th edition.

Kocaoz, S., Samaranayake, V. A., and Nani, A. (2005). Tensile characterization of glass FRP bars. *Composites: Part B*, 36:127–134.

Lenth, R. V. (1989). Quick and easy analysis of unreplicated factorials. *Technometrics*, 31:469–473.

Li, X., Sudarsanam, N., and Frey, D. D. (2006). Regularities in data from factorial experiments. *Complexity*, 11:32–45.

Luca, M. and Bazerman, M. H. (2020). *The Power of Experiments: Decision Making in a Data-Driven World*. MIT Press, Cambridge.

Morris, M. D. (2011). *Design of Experiments: An Introduction based on Linear Models*. Chapman and Hall/CRC Press, Boca Raton.

Owen, M. R., Luscombe, C., Lai, L., Godbert, S., Crookes, D. L., and Emiabata-Smith, D. (2001). Efficiency by design: optimisation in process research. *Organic Process Research and Development*, 5:308–323.

Wu, C. F. J. and Hamada, M. (2009). *Experiments: Planning, Analysis, and Parameter Design Optimization*. Wiley, New York, 2nd edition.