

# Statistical Methods for Test and Evaluation, Volume 4: Experimental Design using R

Jeffrey Strickland, Ph.D.

United States Space Force, Colorado Springs, CO, USA



**Statistical Methods for Test and Evaluation, Volume 4:  
Experimental Design using R**

© 2024 by Jeffrey Strickland, Ph.D.

Licensed under CC BY-NC-SA 4.0. To view a copy of this license,  
visit <https://creativecommons.org/licenses/by-nc-sa/4.0/>

ISBN: 978-1-304-26225-7



# Table of Contents

---

<b>1. Introduction.....</b>	<b>1</b>
1.1. Motivation.....	1
1.2. Survey, Observation, and Experimentation .....	1
1.3. Designs and Methodologies .....	3
1.4. Beginnings of Statistically Planned Experiments.....	4
1.5. Definitions and Preliminaries .....	5
1.6. Purposes of Experimental Design .....	8
1.7. Types of Experimental Designs.....	10
1.8. Planning Experiments.....	12
1.9. Performing the Experiments .....	15
1.10. Use of R Software .....	18
1.11. A Brief History .....	19
1.12. Review of Important Concepts .....	22
<b>2. Completely Randomized Designs - One Factor .....</b>	<b>25</b>
2.1. Introduction.....	25
2.2. Example – Aerogels for Space Operations .....	25
2.2.1. Replication.....	25
2.2.2. Randomization .....	26
2.3. A Historical Example.....	28
2.4. Linear Model for CRD .....	29
2.4.1. Matrix Representation .....	31
2.4.2. LS Calculations with R Function lm.....	32
2.4.3. Estimation of $\sigma^2$ and Distribution of Quadratic Forms .....	34
2.4.4. Estimable Functions .....	34
2.4.5. Hypothesis Test of No Treatment Effects.....	35
2.4.6. A Word of Caution.....	38
2.5. Verifying Assumptions of the Linear Model .....	39

2.6.	Analysis Strategies When Assumptions Are Violated .....	42
2.6.1.	Box-Cox Power Transformations .....	43
2.6.2.	Distribution-Based Transformations .....	45
2.6.3.	Alternatives to Least Squares Analysis .....	47
2.7.	Determining the Number of Replicates.....	50
2.8.	Comparison of Treatments after the F-test.....	53
2.8.1.	Preplanned Comparisons – Rad-Hard Protection.....	54
2.8.2.	Unplanned Comparisons – Rad-Hard Protection.....	57
2.8.3.	Comparison of All Means to a Control – Rad-Hard Protection .....	61
2.9.	Example: Unmanned Aerial Vehicle Design.....	64
2.10.	Review of Important Concepts .....	65
<b>3.</b>	<b>Factorial Designs .....</b>	<b>69</b>
3.1.	Introduction.....	69
3.2.	Classical One at a Time versus Factorial Plans.....	69
3.3.	Interpreting Interactions .....	71
3.4.	Creating a Two-Factor Factorial Plan in R.....	75
3.5.	Analysis of a Two-Factor Factorial in R .....	76
3.5.1.	Matrix Representation of Model and Analysis .....	77
3.5.2.	Determining the Number of Replicates.....	85
3.5.3.	Analysis with an Unequal Number of Replicates per Cell.....	88
3.5.4.	Testing for Interaction with One Replicate per Cell .....	93
3.6.	Factorial Designs with Multiple Factors—CRFD .....	98
3.7.	Example: Screening Variables Affecting Zinc Battery Electrodes	101
3.7.1.	Experiment Variables.....	102
3.7.2.	The Experimental Design .....	102
3.8.	Two-Level Factorials .....	108
3.8.1.	Main Effects and Regression Slopes .....	109
3.8.2.	Interactions.....	111

3.9.	Example - Silica Aerogel 2k Factorial .....	113
3.9.1.	Introduction.....	113
3.9.2.	Data Dictionary.....	117
3.9.3.	Factorial design 2 <sup>2</sup> : Input values .....	118
3.9.4.	Define and Enter Input Values .....	118
3.9.5.	Effects.....	119
3.9.6.	Regression Model.....	120
3.9.7.	Interaction Plots .....	120
3.9.8.	ANOVA Model .....	121
3.9.9.	ANOVA Model (Alternate).....	122
3.9.10.	Display a Raincloud Plot .....	122
3.10.	Example of a 2 <sup>3</sup> Factorial.....	123
3.10.1.	Shortcut Formula for Determining the Number of Replicates ....	129
3.10.2.	Analysis with One Replicate per Cell .....	130
3.10.3.	Verifying Assumptions of the Model.....	139
3.11.	Experiment Example – Antireflection Coating (ARC).....	143
3.11.1.	Factors and Responses .....	144
3.11.2.	Data Prep.....	144
3.11.3.	<b>Part 1:</b> Model fitting .....	145
3.11.4.	Part 2. Make a Half-Normal Plot .....	149
3.11.5.	Problem 3. Drop insignificant terms and Remodel .....	153
3.12.	Review of Important Concepts.....	160
<b>4.</b>	<b>Randomized Block Designs .....</b>	<b>163</b>
4.1.	Introduction.....	163
4.2.	Creating an RCB in R .....	164
4.3.	Model for RCB .....	166
4.4.	Example of an RCB – Advanced Combat Helmet.....	168
4.4.1.	ACH RTP Performance Requirements .....	169
4.4.2.	FAT RTP Requirements .....	169

4.5.	Example of an RCB – ADHD Research.....	174
4.6.	Determining the Number of Blocks .....	178
4.7.	Factorial Designs in Blocks.....	179
4.8.	Example of an RCBF – Cancer Research.....	180
4.9.	Generalized Complete Block Design.....	183
4.10.	Two Block Factors LSD .....	187
	4.10.1.Creating and Randomizing Latin-Square Designs .....	189
	4.10.2.Analysis of a Latin-Square Design .....	190
	4.10.3.Determining the Number of Replicates.....	192
4.11.	Review of Important Concepts .....	193
<b>5.</b>	<b>Designs to Study Variances.....</b>	<b>197</b>
5.1.	Introduction.....	197
5.2.	Reason for Studying Variability.....	197
5.3.	Random Factors and Random Sampling Experiments.....	199
5.4.	One-Factor Sampling Designs .....	201
5.5.	Estimating Variance Components.....	203
	5.5.1. Method of Moments Estimators .....	204
	5.5.2. Interval Estimates .....	207
	5.5.3. Maximum Likelihood and REML Estimators .....	209
	5.5.4. Determining the Sample Size For One-Factor Sampling Studies	213
5.6.	Two-Factor Sampling Designs .....	216
	5.6.1. Estimating Variance Components.....	218
	5.6.2. Confidence Intervals on Variance Components in Two-Factor Designs .....	221
	5.6.3. Determining Sample Sizes for Two-Factor Sampling Experiments	223
	5.6.4. Two-Factor Studies with Unequal Replication.....	223
5.7.	Nested Sampling Experiments (NSE) .....	227
5.8.	Staggered Nested Designs .....	230

5.9.	Designs with Fixed and Random Factors.....	237
5.10.	Graphical Methods to Check Model Assumptions .....	247
	5.10.1.Simple Plots .....	247
	5.10.2.Gamma and Half-Normal Plots to Check Constant Variance Assumption.....	249
	5.10.3.Probability Plots of Empirical Best Linear Unbiased Estimates of Random Effects.....	254
5.11.	Review of Important Concepts.....	256
<b>6.</b>	<b>Fractional Factorial Designs.....</b>	<b>259</b>
6.1.	Introduction.....	259
6.2.	Half-Fractions of $2^k$ Designs.....	260
6.3.	Quarter and Higher Fractions of $2^k$ Designs.....	274
6.4.	Criteria for Choosing Generators for $2^{k-p}$ Designs.....	276
6.5.	Augmenting Fractional Factorials.....	291
	6.5.1. Optional: rand seed settings .....	292
	6.5.2. Augmenting by Foldover or Mirror Image Designs .....	292
	6.5.3. Augmenting by Optimal Design.....	301
6.6.	Plackett-Burman (PB) Model Robust Screening Designs.....	304
6.7.	Mixed Level Factorials and Orthogonal Arrays (OAs).....	321
6.8.	Example – Force-Level Encounter Assessment .....	327
	6.8.1. Choice of Experimental Designs and Data Generation .....	330
	6.8.2. Full Factorial Design .....	330
	6.8.3. Load FrF2 package & Kill Ratio Data.....	331
	6.8.4. Load the Yield Factors .....	332
	6.8.5. Fractional Factorial Design 1 .....	332
	6.8.6. Fractional Factorial Design 2 .....	334
	6.8.7. Randomization, Blocking and Replication .....	334
	6.8.8. Adding Response Variable to the Design .....	334
	6.8.9. Randomization for First Design .....	335

6.8.10. Randomization for Second Design.....	335
6.8.11. Model Comparison .....	336
6.8.12. Estimation.....	338
6.8.13. Final Modeling & Analysis.....	338
6.8.14. Calculate Main Effects .....	340
6.8.15. Print Main Effects .....	341
6.8.16. Interaction Plots .....	341
<b>6.9. Example Experiment: Crosslinking Amine-Modified Silica Aerogels with Epoxies.....</b>	<b>346</b>
6.9.1. The Laboratory Experiment .....	347
6.9.2. Experimental Design.....	347
6.9.3. Exploratory Data Analysis .....	350
6.9.4. Experimental Results .....	352
<b>6.10. doer R ShinyApp .....</b>	<b>353</b>
6.10.1.Load skpr Package .....	353
6.10.2.The doer GUI.....	353
6.10.3.The Advanced Tab .....	354
6.10.4.doer Model Tab .....	355
6.10.5.Generating candidate set: .....	357
6.10.6.Generating design:.....	357
6.10.7.Evaluating (Monte Carlo) Design: .....	357
6.10.8.Analysis .....	358
6.10.9.Linear model (lm) with main effects only.....	358
6.10.10. lm with main effects and 2nd order effects .....	359
6.10.11. ANOVA Model with Main Effect only .....	361
6.10.12. ANOVA Model with main effects and 2nd order effects ...	361
<b>6.11. Review of Important Concepts .....</b>	<b>362</b>
<b>7. Incomplete and Confounded Block Designs .....</b>	<b>365</b>
7.1. Introduction.....	365

7.2.	Balanced Incomplete Block (BIB) Designs .....	366
7.3.	Analysis of Incomplete Block Designs .....	369
7.3.1.	Example – Taste Test.....	370
7.3.2.	Determining the Number of Replicates.....	372
7.4.	BTIB and PBIB Designs.....	373
7.5.	Row Column Designs.....	378
7.6.	Confounded $2^k$ and $2^{k-p}$ Designs.....	379
7.6.1.	Confounding 2k Designs .....	380
7.6.2.	Confounding in $2^{k-p}$ Designs .....	386
7.6.3.	Objective .....	389
7.6.4.	Design.....	389
7.6.5.	Participants.....	389
7.6.6.	Interventions .....	390
7.6.7.	Main Outcome Measures.....	390
7.6.8.	Criteria for Choosing Block Defining Contrasts .....	391
7.7.	Confounding 3-Level and p-Level Factorial Designs .....	395
7.8.	Blocking Mixed Level Factorials and OAs .....	398
7.8.1.	Blocking Mixed Level Factorials.....	399
7.8.2.	Blocking Orthogonal Arrays, Orthogonal Main Effect Plans, and Nearly Orthogonal Main Effect Plans .....	407
7.9.	Partially Confounded Blocked Factorial (PCBF) .....	408
7.10.	Review of Important Concepts.....	415
<b>8.</b>	<b>Response Surface Designs .....</b>	<b>419</b>
8.1.	Introduction.....	419
8.2.	Fundamentals of Response Surface Methodology.....	420
8.2.1.	Empirical Quadratic Model.....	420
8.2.2.	Design Considerations .....	421
8.3.	Standard Designs for Second Order Models .....	424
8.3.1.	Central Composite Design (CCD) .....	424

8.3.2.	Box-Behnken Design .....	427
8.3.3.	Small Composite Design .....	431
8.3.4.	Hybrid Design.....	434
8.4.	Creating Standard Response Surface Designs in R .....	435
8.5.	Non-Standard Response Surface Designs.....	439
8.6.	Fitting the Response Surface Model with R.....	448
8.6.1.	Fitting a Linear Model and Checking for Curvature.....	448
8.6.2.	Fitting the General Quadratic Model.....	450
8.6.3.	Fitting a Nonlinear Mechanistic Model .....	454
8.7.	Determining Optimum Operating Conditions .....	456
8.7.1.	Contour Plots.....	456
8.7.2.	Canonical Analysis .....	458
8.7.3.	Ridge Analysis .....	462
8.7.4.	Nonlinear Optimization .....	464

## Preface

---

*“Experimental design is a technique that enables scientists and engineers to efficiently assess the effect of multiple inputs, or factors, on measures of performance, or responses. Compared to one-factor-at-a-time, trial-and-error approaches, a well-designed experiment can provide clear results while dramatically reducing the required amount of testing.”*

—Bradley Jones, JMP

A methodology for designing experiments was proposed by Ronald Fisher, in his innovative books: *The Arrangement of Field Experiments* (1926) and *The Design of Experiments* (1935). Much of his pioneering work dealt with agricultural applications of statistical methods. As a mundane example, he described how to test the lady tasting tea hypothesis, that a certain lady could distinguish by flavor alone whether the milk or the tea was first placed in the cup. These methods have been broadly adapted in biological, psychological, and agricultural research. However, this book was written for scientists and engineers engaged in the formal process of test and evaluation (T&E) for system acquisition in the Department of Defense.

Test engineers and analyst study experimental design to:

1. Choose an experimental design that is appropriate for the chosen test design (e.g. system characterization, error identification, etc. (see Table 1-1. Linkage between Test Design and Designs of Experiment (DOE)).
2. Construct the design (including performing proper randomization and determining the required number of replicates).
3. Execute the plan to collect the data (or advise a team members on how to do it).
4. Determine the model appropriate for the data
5. Fit the model to the data.
6. Interpret the data and present the results in a meaningful way to answer the research question.

The purpose of this book is to focus on connecting the objectives of testing to the type of experimental design required, describing the actual process of creating the design and collecting the data, showing how to perform the proper analysis of the data, and illustrating the interpretation of results. Explanation on the mechanics of computation is minimized by relying on the statistical software package, R.

With the availability of modern statistical computing packages, the analysis of data has become much easier and is well covered in statistical methods books. In a book on the design and analysis of experiments, there is no longer a need to show all the computational formulas that were necessary before the advent of modern computing. However, there is a need for careful explanation of how to get the proper analysis from a computer package. The default analysis performed by most statistical software assumes the data have come from a completely randomized design. In practice, this is often a false assumption. This book emphasizes the connection between the experimental units, and the way treatments are randomized to experimental units, and the proper error term for an analysis of the data.

We use R is used throughout this book to illustrate both construction of experimental designs and analysis of data. We chose R because it is an open-source software that can be downloaded free of charge for Windows, Linux, and Apple operating systems from [www.r-project.org](http://www.r-project.org). Additionally, user developed packages for R have given it extensive capabilities in both creating experimental designs and analyzing data. Information about many of these user written packages is available on at: <http://cran.r-project.org/web/views/ExperimentalDesign.html>.

We illustrate user written packages along with base R functionality in numerous examples in the text. The user generated packages simplify things that could require extensive R coding without their use. The code examples in the book are available for download on my GitHub site. It is possible that some of the R packages illustrated in this book may be removed from CRAN, if the developer drops support, or the developer may make changes so that the examples in the book no longer work. If problems such as this arise, I will make revisions to the code online so that a working copy of the examples is available.

With fewer pages devoted to computational formulas, I have attempted to spend more time discussing the following: (1) how the objectives of a system test leads to the choice of an appropriate design, (2) practical aspects of creating a design or list of experiments to be performed, (3) practical aspects of performing experiments, and (4) interpretation of the results of a computer analysis of the data. We can best learn items (1)–(3) by studying many examples of experiments that require readers to perform their own experiments.

This book attempts to give uniform coverage to experimental designs and design concepts that are most commonly used in system testing, rather than focusing on specialized areas. Although tests for software systems are covered in this volume, we wrote a separate volume specific software testing. I base the selection of topics is based on my own experience working in the space, defense, banking, insurance, and research and development (R&D). At the end of each chapter a diagram is presented to help identify where the various designs should be used. The examples I chose for this book come from a variety of application areas, but focuses primarily on space and defense, when suitable cases exist. We place emphasis on how the sample size, the assignment of experimental units to combinations of treatment factor levels (error control), and the selection of treatment factor combinations (treatment design) will affect the resulting variance and bias of estimates and the validity of conclusions.

### **Example Defense and Space Applications**

<b>Example Application</b>	<b>Section(s)</b>
Thermosphere Cooling	1.2
Ethanol Fuel Formulation	3.5,
Aerogels for Space Operations	2.2, 2.5-2.7, 3.6, 3.9, 5.3, 6.9
Unmanned Aerial Vehicle (UAV) Design	2.9, 3.2-3.5, 4.3, 5.4
Rad-Hard Protection	2.8, 4.4, 7.6
Screening Variable Affecting Zinc Battery Electrodes	3.7
Antireflection Coatings (ARC)	3.11
Advanced Combat Helmet (ACH)	4.4, 7.6
Force Level Encounter Assessment	6.8
Eagle Tactical Athlete Program (ETAP)	7.6
Government Electronics Lab	3.10
Golf	4.9

*Intended audience:* I wrote this book primarily for Department of Defense test organizations, specifically for testers with college credit in undergraduate statistics, mathematics, operations research, or engineering. To be fully comprehended, a student using this book should have had previous courses in calculus, introductory statistics, basic statistical theory, and applied linear models such as Kutner *et al.* (2004) and Faraway (2015). We use matrix notation for analysis of linear models throughout the book, and students should be familiar with matrix operations at least to the degree illustrated in chapter 5 of Kutner *et al.* (2004). Also we assume the reader has some experience with R or command driven statistical software (e.g., SAS, Python).

However, for tester from applied sciences or engineering who do not have all these prerequisites, there is still much to be gained from this book. There are many examples of diagnosing the experimental environment to choose the correct design, creating the design, analyzing data, and interpreting and presenting results of analysis. There is R code to create and analyze all the example tests and experiments in the book. One with a basic understanding of R, and aid of the documentation for the R functions illustrated, should be able to follow these examples and modify them to solve problems in their own research without needing to understand the detailed theoretical justification for each procedure.

*For instructors:* This book can be used for a one-semester or two-quarter course in experimental design. There is too much material for a one-semester course, unless the students have had all the prerequisites mentioned above. The first four chapters in the book cover the classical ideas in experimental design and should be covered in any course for students without a prior background in designed experiments. Later chapters start with basics but proceed to the latest research published on particular topics, and they include code to implement all of these ideas. An instructor can pick and choose from these remaining topics, although if there is time to cover the whole book, I would recommend presenting the topics in order.

For general testers whose primary responsibilities do not include statistical analysis, covering factorial experiments in Chapter 3, fractional factorials in Chapter 6, and response surface methods in Chapter 8,

following the pattern established by the DuPont Strategies of Experimentation Short Courses that were developed in the 1970s. I did not include more advance topics in experimental designs, including split -plot designs, crossover and repeated measures design, mixture experiments, and robust parameter designs. Several of these advanced topics are covered by Montgomery (1997).

*Acknowledgments:* This book is the culmination of many years of thought and application through consulting and teaching at the National Aeronautics and Space Administration (NASA), Missile Defense Agency (MDA), Army Space and Missile Defense Command (SMDC), The Amry Logistics Management University (ALMU), The United States Military Academy (USMA), the United Services Automobile Association (USAA), The Vellore Institute of Technology (VIT) and more.

*“DOE was born in agriculture. We can’t grow corn in Space.”*

- Jeffrey Strickland, Ph.D.

## *Other Technical Books by the Author*

1. Statistical Methods for Test and Evaluation, Volume 1: The Combat Test Framework, Jeffrey Strickland, 2024, CC BY-NC-SA. Lulu, Inc. ISBN 978-1-304-26228-8
2. Statistical Methods for Test and Evaluation, Volume 3: Combinatorial Design using R, Jeffrey Strickland, 2024, CC BY-NC-SA. Lulu, Inc. ISBN 978-1-304-42779-3
3. Statistical Methods for Test and Evaluation, Volume 4: Experimental Design using R, Jeffrey Strickland, 2024, CC BY-NC-SA. Lulu, Inc. ISBN 978-1-304-26228-8
4. Statistical Methods for Test and Evaluation, Volume 4: Experimental Designs using Excel, Jeffrey Strickland, 2022, CC BY-NC-SA. Lulu, Inc. ISBN 978-1-387-90547-8
5. Practitioner's Guide to Military Modeling and Simulation, 2024 by Jeffrey Strickland, CC BY-NC-SA. Lulu, Inc. ISBN 978-1-304-42780-9
6. *Building Scenarios in Freeciv: The Templars*, 2024 by Jeffrey Strickland, CC BY-NC-SA. Lulu, Inc. ISBN: 9-781304633087
7. *Discrete Event Simulation Using ExtendSim 10*, Copyright 2023 by Jeffrey S. Strickland. Lulu, Inc., ISBN: 978-1-716-20282-7
8. *Regression totum modum*. Copyright © 2023 by Jeffrey S. Strickland. Lulu, Inc., ISBN: 978-1-329-33732-9
9. *Orbital Mechanics using Python and R*, Copyright © 2022 by Jeffrey S. Strickland. Lulu, Inc., ISBN: 978-1-387-50683-5
10. *Systems Engineering Processes and Practice*, Second Edition, Copyright © 2022, Lulu, Inc., ISBN 978-1-387-81105-2
11. *The Space Handbook*, Jeffrey Strickland, Copyright © 2022, Lulu, Inc., ISBN 978-1-387-97656-3
12. *The Python Guide for New Data Scientists*, Jeffrey Strickland, Copyright © 2022, Lulu, Inc. ISBN 978-1-4583-2161-9
13. *The R Guide for New Data Scientists*, Copyright © 2022, Lulu, Inc. ISBN 978-1-6780-0244-2
14. *Time Series Analysis and Forecasting using Python & R*. Jeffrey Strickland, Copyright © 2020, Lulu, Inc. ISBN 978-1-716-45113-3
15. *Data Science Applications using Python and R*. Jeffrey Strickland, Copyright © 2020, Lulu, Inc. ISBN 978-1-716-89644-6

16. *Data Science Applications using R*. Jeffrey Strickland, Copyright© 2020, Lulu, Inc. ISBN 978-0-359-81042-0
17. *Predictive Crime Analysis using R*. Jeffrey Strickland, Copyright© 2018, Jeffrey Strickland. Lulu, Inc. ISBN 978-0-359-43159-5
18. *Logistic Regression Inside-Out*. Copyright Jeffrey Strickland, © 2017 by Jeffrey S. Strickland. Lulu, Inc. ISBN 978-1-365-81915-5
19. *Time Series Analysis using Open-Source Tools*. Jeffrey Strickland, Copyright© 2016, Jeffrey Strickland. Glasstree, Inc. ISBN 978-1-5342-0100-2
20. *Data Analytics Using Open-Source Tools*. Copyright Jeffrey Strickland, © 2015 by Jeffrey Strickland. Lulu Inc. ISBN 978-1-365-21384-7
21. *Missile Flight Simulation - Surface-to-Air Missiles*, 2<sup>nd</sup> Edition. Copyright © 2015 by Jeffrey S. Strickland. Lulu.com. ISBN 978-1-329-64495-3
22. *Predictive Analytics using R*. Copyright © 2015 by Jeffrey S. Strickland. Lulu.com. ISBN 978-1-312-84101-7
23. *Operations Research using Open-Source Tools*. Copyright © 2015 by Jeffrey Strickland. Lulu Inc. ISBN 978-1-329-00404-7.
24. *I Rode With Wallace*. Copyright © 2015 by Jeffrey S. Strickland. Lulu, Inc., ISBN: 978-1329565661
25. *Predictive Modeling and Analytics*. © 2014 by Jeffrey S. Strickland. Lulu.com. ISBN 978-1-312-37544-4.
26. *Verification and Validation for Modeling and Simulation*. Copyright © 2014 by Jeffrey S. Strickland. Lulu.com. ISBN 978-1-312-74061-7.
27. *Using Math to Defeat the Enemy: Combat Modeling for Simulation*. © 2011 by Jeffrey S. Strickland. Lulu.com. ISBN 978-1-257-83225-5.
28. *Mathematical Modeling of Warfare and Combat Phenomenon*. Copyright © 2011 by Jeffrey S. Strickland. Lulu.com. ISBN 978-1-45839255-8.
29. *Simulation Conceptual Modeling*. Copyright © 2011 by Jeffrey S. Strickland. Lulu.com. ISBN 978-1-105-18162-7.
30. *Systems Engineering Processes and Practice*, Copyright © 2011 by Jeffrey S. Strickland. Lulu.com. ISBN 978-1-257-09273-4.

31. *Discrete Event Simulation using ExtendSim 8*. Copyright © 2010 by Jeffrey S. Strickland. Lulu.com. ISBN 978-0-557-72821-3
32. *Fundamentals of Combat Modeling*, Copyright © 2011 by Jeffrey S. Strickland. Lulu.com, ISBN 978-1-257-00583-3.

### ***Non-Technical Books by the Author***

33. *Knights of the Cross*, Copyright © 2011 by Jeffrey S. Strickland. Lulu.com, ISBN 978-1-105-35162-4.
34. A Mason's Mason, by Jeffrey S. Strickland. Lulu.com 2024 by Jeffrey S. Strickland. Lulu, Inc., ISBN 978-1-312-03214-9
35. *Weird Scientists – the Creators of Quantum Physics*, Copyright © 2011 by Jeffrey S. Strickland. Lulu.com, ISBN 978-1-257-97624-9
36. *Albert Einstein: "Nobody expected me to lay golden eggs"*, Copyright © 2011 by Jeffrey S. Strickland. ISBN 978-1-257-86014-2.
37. *The Men of Manhattan: Creators of the Nuclear Era*, Copyright © 2011 by Jeffrey S. Strickland. Lulu.com, ISBN 978-1-257-76188-3.
38. *The Devil Did Not Make Me Do It*, Copyright © 2018 by Jeffrey S. Strickland, Printed by Lulu, Inc., ISBN 9781365982149.
39. *To Watch Over Them Day and Night*, Copyright © by Jeffrey Strickland, Printed by Lulu, Inc. ISBN 9781365880759.
40. *Quantum Phaith*, Copyright © 2011 by Jeffrey Strickland, Printed by Lulu, Inc., ISBN 9781257645619
41. *Quantum Hope*, Copyright © 2015 by Jeffrey Strickland, Printed by Lulu, Inc., ISBN 9781329160781.
42. *Quantum Love*, Copyright © 2017 by Jeffrey Strickland, Printed by Lulu, Inc., ISBN 9781365835551
43. *I Rode with Wallace*, Copyright © 2015 by Jeffrey Strickland, Printed by Lulu, Inc., ISBN 9781329565661.
44. *Handbook of Handguns*, Cop ht © 2013 by Jeffrey Strickland, Printed by Lulu, Inc., ISBN 9781300973294.
45. *LinkedIn Memoirs*, Copyright © 2014 by Jeffrey Strickland, Printed by Lulu, Inc., ISBN 9781312717329.
46. *Dear Mr. President*, Copyright © 2012 by Jeffrey Strickland, Printed by Lulu, Inc., ISBN 9781105555145.

# 1. Introduction

---

## 1.1. Motivation

*"You can't fix by analysis what you bungled by design"*

—Light, Singer and Willett (1990)

In testing systems produced in the Defense Acquisition Framework for **operational acceptance**, it is necessary to use quality statistics to ensure that suppliers meet our quality standards. We define statistics as the science of collecting, analyzing, and drawing conclusions from data. Data is usually collected through sampling surveys, observational studies, or operational experiments. In this volume, we cover the bulk of experimental design options that testers using in the course of operational test and evaluation.

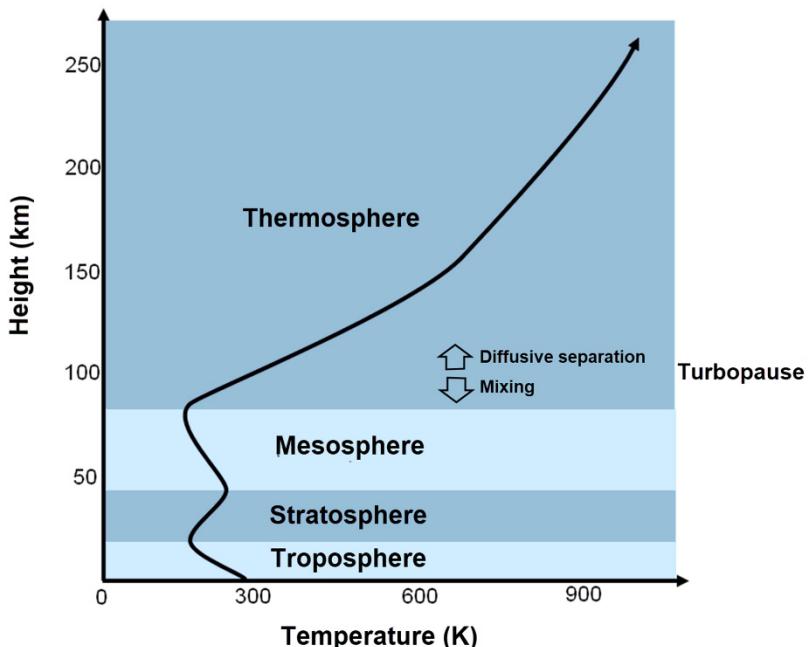
Although this book is similar to other books about experimental designs or design of experiments (DOE), we introduce **defense acquisition** examples, along with the more common examples from agriculture and medicine.

## 1.2. Survey, Observation, and Experimentation

Sampling surveys are normally used when the purpose of data collection is to estimate some property of a finite population without conducting a complete census of every item in the population. For example, if we were interested in finding the proportion of operational users of a space domain awareness software system that favors the graphical user interface (GUI), this proportion could be estimated by polling a random sample of operational users rather than questioning every registered user of the software system.

Observational studies and experiments, on the other hand, are normally used to determine the relationship between two or more measured quantities in a conceptual **battlespace**. For example, through observation, we believe there is a relationship between warming in the *troposphere* and cooling in the *ionosphere*, particularly the

*thermosphere* (the upper layer in **Figure 1-1**). Observational data supports this conclusion. The long-term changes in the upper atmosphere have important practical implications. For example, the decline in density due to cooling of the thermosphere reduces atmospheric drag on objects passing through the thermosphere, increasing the lifetime and accumulation rate of space debris.



**Figure 1-1.** The regions of the Earth's upper atmosphere. The ionosphere lies within the Thermosphere.

To fully understand the dynamics of atmospheric changes, experimentation is required. Herein lies the main difference between **observation** and **experimentation**: to experiment we must control the factors that are related to the response we are studying. For instance, if we hypothesize that the increase in CO<sub>2</sub> concentration is the main driver of the global mean cooling and contraction of the upper atmosphere, we have to control other factors, like other trace gases, the Earth's magnetic field, and solar radiation (we currently do this through simulation) to isolate the true relationship between CO<sub>2</sub> concentration and thermospheric cooling. To absolutely project that future thermospheric cooling is due to greenhouse warming, i.e., CO<sub>2</sub> concentration in the

troposphere, with observational data alone is irresponsible. Rather, controlled experimentation is required.

### **1.3. Designs and Methodologies**

When testing system performance and system effectiveness, the test design depends on the objectives of the test. It may surprise some readers that test design of experiments (DOE) or experimental design is not synonymous with design of a test or test design. In operational testing there are generally three primary test designs: system characterization, system error identification, and system comparison. To execute these test designs, we require methodologies that support the design objectives (see **Table 1-1**). The body of these methodologies is called experimental design. Whether we are simply making observations or identifying errors we require measures and the data that is necessary for evaluating them.

**Table 1-1. Linkage between Test Design and Designs of Experiment (DOE)**

<b>Test Design (Objective)</b>	<b>Potential Experimental Design (Methodology)</b>
<b>System Characterization</b>	Factorial designs, fractional factorial designs, response surface designs, optimal designs
<b>Problem Identification</b>	Combinatorial designs, Orthogonal Arrays, Space filling designs
<b>Compare two or more systems</b>	Factorial or fractional factorial designs, matched pairs optimal designs
<b>Screen for important factors</b>	Factorial or fractional factorial designs
<b>Optimize system performance</b>	Response surface designs, optimal designs
<b>Predict performance</b>	Response Surface Designs, Optimal Designs, Accelerated life tests
<b>Improve system reliability</b>	Response surface designs, Taguchi designs, Orthogonal Arrays

When testing system performance and effectiveness, the methodologies we use depend on the outcomes we expect. In some instances, we merely describe system behavior, while in others we look for statistical correlation or cause-and-effect. With respect to the thermosphere problem, our objective is most likely to understand the effect on the thermosphere caused by changes in the troposphere. That is, to make accurate and actionable predictions of what will happen when factors of the environment are controlled, cause and effect relationships must be

assumed. For example, to predict the future average temperature of the thermosphere and that greenhouse gas emissions will be controlled at a certain level, we must assume that the relationship between greenhouse gas emissions and global temperature is cause and effect.

Herein lies the first difference in observational studies and experiments. In an **observational study**, data is observed in its natural environment, but in an **experiment** the environment is controlled. In observational studies we cannot prove that the relationships detected are **cause and effect**. We may find **correlations** between two observed variables because they are both affected by changes in a third variable that was not observed or recorded. Any future predictions we make based on the relationships we find in an observational study must assume the same interrelationships among variables that existed in the past will exist in the future. In an experiment, on the other hand, some variables or **factors** are purposely changed while others are held constant. In that way the effect that is caused by the change in the purposely varied factor can be directly observed, and we can make predictions about the result of future changes to the purposely varied factor. Both situations require different methodologies for collecting and analyzing data.

## **1.4. Beginnings of Statistically Planned Experiments**

*"Scientific research is a process of guided learning. The object of statistical methods is to make that process as efficient as possible."*

—George E. P. Box

There are many purposes for experimentation. Some examples include determining the cause for variation in measured responses observed in the past; finding conditions that give rise to the maximum or minimum response; comparing the response between different settings of controllable variables; and obtaining a mathematical model to predict future response values.

Presently, planned experiments are used in many different fields of application such as: engineering design, quality improvement, industrial research and manufacturing, basic research in physical and biological science, research in social sciences, psychology, business management

and marketing research, and many more. However, the roots of modern experimental design methods stem from R. A. Fisher's (1958) work in agricultural experimentation at the Rothamsted Experimental Station near Harpenden, England.

Fisher was a gifted mathematician whose first paper as an undergraduate at Cambridge University introduced the theory of likelihood. He was later offered a position at University College but turned it down to join the staff at Rothamsted in 1919. There, inspired by daily contact with agricultural research, he not only contributed to experimental studies in areas such as crop yields, field trials, and genetics, but also developed theoretical statistics at an astonishing rate. He also developed with the ideas for planning and analysis of experiments that we use as the basis for valid inference and prediction in various fields of application to this day. Fisher (1926) first published his ideas on planning experiments in his paper "The arrangement of field experiments"; nine years later he published the first edition of his book *The Design of Experiments*, Fisher (1966).

The challenges that Fisher faced were the large amount of variation in agricultural and biological experiments that often confused the results, and the fact that experiments were time consuming and costly to carry out. This motivated him to find experimental techniques that could:

- eliminate as much of the natural variation as possible
- prevent unremoved variation from confusing or biasing the effects being tested
- detect cause and effect with the minimal amount of experimental effort necessary.

## **1.5. Definitions and Preliminaries**

Before initiating an extended discussion of experimental designs and the planning of experiments, I will begin by defining the terms that we use frequently.

**Experiment (also called a Run)** is an action where the experimenter changes at least one of the variables being studied and then observes the effect of his or her actions(s). Note the passive collection of

observational data is not experimentation.

**Experimental Unit** is the item under study upon which something is changed. This could be raw materials, human subjects, or just a point in time.

**Sub-Sample, Sub-Unit, or Observational Unit.** When the experimental unit is split, after the action has been taken upon it, this is called a sub-sample or sub-unit. Sometimes it is only possible to measure a characteristic separately for each sub-unit; for that reason they are often called observational units. Measurements on sub-samples, or sub-units of the same experimental unit, are usually correlated and should be averaged before analysis of data rather than being treated as independent outcomes. When sub-units can be considered independent and there is interest in determining the variance in sub-sample measurements, while not confusing the *F*-tests on the treatment factors, the mixed model described in **Section 5.5.3** should be used instead of simply averaging the sub-samples.

**Independent Variable (Factor or Treatment Factor)** is one of the variables under study that is being controlled at or near some **target** value, or **level**, during any given experiment. The level is being changed in some systematic way from run to run in order to determine what effect it has on the response(s).

**Background Variable** (also called a **Lurking Variable**) is a variable that the experimenter is unaware of or cannot control, and which could have an effect on the outcome of the experiment. In a well-planned experimental design, the effect of these **lurking variables** should balance out so as to not alter the conclusion of a study.

**Dependent Variable** (or the **Response** denoted by  $Y$ ) is the characteristic of the experimental unit that is measured after each experiment or run. The magnitude of the response depends upon the settings of the independent variables or factors and lurking variables.

**Effect** is the change in the response that is caused by a change in a factor or independent variable. After the runs in an experimental design are conducted, the effect can be estimated by calculating it from the observed response data. This estimate is called the **calculated effect**.

Before the experiments are ever conducted, the researcher may know how large the effect should be to have practical importance. This is called a **practical effect** or the **size of a practical effect**.

**Replicate** runs are two or more experiments conducted with the same settings of the factors or independent variables but using different experimental units. The measured dependent variable may differ among replicate runs due to changes in lurking variables and inherent differences in experimental units.

**Duplicates** refer to duplicate measurements of the same experimental unit from one run or experiment. The measured dependent variable may vary among duplicates due to measurement error, but in the analysis of data these duplicate measurements should be averaged and not treated as separate responses.

**Experimental Design** is a collection of experiments or runs that is planned in advance of the actual execution. The particular runs selected in an experimental design will depend upon the purpose of the design.

**Confounded Factors** arise when each change an experimenter makes for one factor, between runs, is coupled with an identical change to another factor. In this situation it is impossible to determine which factor causes any observed changes in the response or dependent variable.

**Biased Factor** results when an experimenter makes changes to an independent variable at the precise time when changes in background or lurking variables occur. When a factor is biased it is impossible to determine if the resulting changes to the response were caused by changes in the factor or by changes in other background or lurking variables.

**Experimental Error** is the difference between the observed response for a particular experiment and the long run average of all experiments conducted at the same settings of the independent variables or factors. The fact that it is called “error” should not lead one to assume that it is a mistake or blunder. Experimental errors are not all equal to zero because background or lurking variables cause them to change from run to run. Experimental errors can be broadly classified into two types: bias error and random error. Bias error tends to remain constant or change in a

consistent pattern over the runs in an experimental design, while random error changes from one experiment to another in an unpredictable manner and average to be zero. The variance of random experimental errors can be obtained by including replicate runs in an experimental design.

With these definitions in mind, the difference between observational studies and experiments can be explained more clearly. In an observational study, variables (both independent and dependent) are observed without any attempt to change or control the value of the independent factors. Therefore any observed changes in the response, or dependent variable, cannot necessarily be attributed to observed changes in the independent variables because background or lurking variables might be the cause. In an experiment, however, the independent variables are purposely varied, and the runs are conducted in a way to balance out the effect of any background variables that change. In this way the average change in the response can be attributed to the changes made in the independent variables.

## ***1.6. Purposes of Experimental Design***

The use of experimental designs is a prescription for successful application of the scientific method. The scientific method consists of iterative application of the following steps:

- a. observing of the state of nature,
- b. conjecturing or hypothesizing the mechanism for what has been observed, then
- c. collecting data, and
- d. analyzing the data to confirm or reject the conjecture.

Statistical experimental designs provide a plan for collecting data in a way that can be analyzed statistically to corroborate the conjecture in question. When an experimental design is used, the conjecture must be stated clearly, and a list of experiments proposed in advance to provide the data to test the ***hypothesis***. This is an organized approach which helps to avoid false starts and incomplete answers to research questions.

Another advantage to using the experimental design approach is the

ability to avoid confounding factor effects. When the research hypothesis is not clearly stated and a plan is not constructed to investigate it, researchers tend toward a trial-and-error approach wherein many variables are simultaneously changed in an attempt to achieve some goal. When this is the approach, the goal may sometimes be achieved, but it cannot be repeated because it is not known what changes actually caused the improvement.

One of Fisher's early contributions to the planning of experiments was popularizing a technique called randomization, which helps to avoid confusion or biases due to changes in background or lurking variables. As an example of what we mean by bias is "The Biggest Health Experiment Ever," (Meier, 1972), wherein a trial of a polio vaccine was tested on over 1.8 million children. An initial plan was proposed to offer vaccinations to all children in the second grade in participating schools, and to follow the polio experience of first through third graders. The first and third grade group would serve as a "control" group. This plan was rejected, however, because doctors would have been aware that the vaccine was only offered to second graders. There are vagaries in the diagnosis of the majority of polio cases, and the polio symptoms of fever and weakness are common to many other illnesses. A doctor's diagnosis could be unduly influenced by his knowledge of whether or not a patient had been vaccinated. In this plan the factor purposely varied, vaccinated or not, was biased by the lurking variable of doctors' knowledge of the treatment.

When conducting physical experiments, the response will normally vary over replicate runs due solely to the fact that the experimental units are different. This is what we defined to be experimental error in the last section. One of the main purposes for experimental designs is to minimize the effect of experimental error. Aspects of designs that do this, such as *randomization*, *replication*, and *blocking*, are called methods of **error control**. Statistical methods are used to judge the average effect of varying experimental factors against the possibility that they may be due totally to experimental error. Another purpose for experimental designs is to accentuate the factor effects (or signal). Aspects of designs that do this, such as choice of the number and spacing of factor levels

and factorial plans, are called methods of **treatment design**. How this is done will be explained in the following chapters.

## 1.7. Types of Experimental Designs

There are many types of experimental designs (see **Table 1-1** for example). The appropriate one to use depends upon the objectives of the experimentation. We can classify objectives into two main categories. The first category is to study the **sources of variability**, and the second is to establish **cause and effect relationships**. When variability is observed in a measured variable, one objective of experimentation might be to determine the cause of that variation. But before we can study cause-and-effect relationships, we must determine a list of independent variables. By understanding the source of variability, researchers are often led to hypothesize what independent variables or factors to study. Thus, experiments designed to study the source of variability are often a starting point for many research programs. The type of experimental design used to classify sources of variation will depend on the number of sources under study. We present these alternatives in Chapter 5.

The appropriate experimental design that we use to study cause and effect relationships will depend on a number of items. Throughout the book we describe the various designs in relation to the purpose for experimentation, the type and number of *treatment factors*, the degree of homogeneity of experimental units, the ease of randomization, and the ability to **block** experimental units into more homogeneous groups. After we present all the designs (if that's even possible), the final chapter describes how they can be used in sequential experimentation strategies where knowledge is increased through different stages of experimentation. Initial stages involve discovering what the important treatment factors are. Later, we quantify the effects of changing treatment factors, and in final stages, we can determine optimal operating conditions. Different types of experimental designs are appropriate for each of these phases. We use screening experiments when the researcher has little knowledge of the cause-and-effect relationships, and many potential independent variables are under study. We usually conduct this type of experimentation early in a research program to

identify the important factors. This is a critical step, and if it is skipped, the later stages of many research programs run amuck because we fail to record or control the important variables.

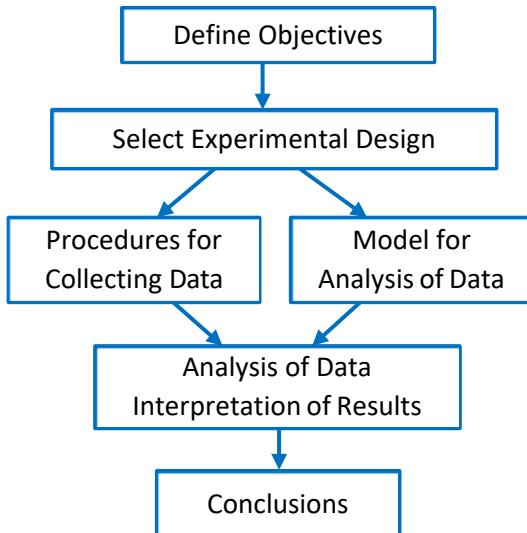
After we identify the most important factors in a screening stage, our next objective would be to choose between **constrained optimization** or **unconstrained optimization** (see Lawson, (2003)). In constrained optimization there are usually six or fewer factors under study and the purpose is to quantify the effects of the factors, their interaction, or joint effects, and to identify optimum conditions among the tested factor combinations.

If our goal is to study only a few quantitative factors and curvilinear relationships with the response variable are possible, it may be possible to identify improved operating conditions by interpolating within the factor levels. If this is the case, the objective of experimentation is called *unconstrained optimization*. With an unconstrained optimization objective, the researcher is normally trying to map the relationship between one or more responses and five or fewer quantitative factors.

We will present specific experimental design plans for each of the stages of experimentation as we progress through the book.

**Figure 1-2** shows the relationship between the objectives of experimentation, the design of the experiment, and the conclusions that we can draw. The objective of a research program dictates which type of experimental design we should use. In turn, the experimental design plan specifies how we should collect the data and what mathematical model we should use to fit to analyze and interpret the data.

Finally, the type of data and selection of the mathematical model will determine the possible conclusions we can draw from the experiment. These steps are inseparable and dependent upon each other. Mistakes are often made in research by trying to alter these steps. We cannot complete an appropriate analysis of data without knowledge of the choice of experimental design and means by which the data was collected. Moreover, conclusions are not reliable if we cannot justify the proper modeling and analysis of the data.



*Figure 1-2. Objectives, Design, and Conclusions from Experimentation*

## 1.8. Planning Experiments

An effective experimental design plan should include the following items: (1) a clear description of the objectives, (2) an appropriate design plan that guarantees unconfounded factor effects and factor effects that are free of bias, (3) a provision for collecting data that will allow estimation of the variance of the experimental error, and (4) a stipulation to collect enough data to satisfy the objectives. Bisgaard (1999) recommends a formal proposal to ensure that a plan includes all of these elements. The proposal should include a checklist for planning the experiments. Below is a checklist that is similar to Bisgaard's. Examples of some of the steps from this checklist will be illustrated in discussing a simple experiment in the next section.

- Define the objectives of the study.** First, this statement should answer the question of why we are performing the experiment. Second, determine if the experiment is conducted to classify sources of variability or if its purpose is to study cause and effect relationships. If it is the latter, determine if it is a screening or optimization experiment. For studies of cause-and-effect

relationships, decide how large an effect should be in order to be meaningful to detect.

- b. **Identify Experimental Units.** Declare the item upon which something will be changed. Is it an animal or human subject, raw material for some processing operation, or simply the conditions that exist at a point in time or trial? Identifying the experimental units will help in understanding the experimental error and variance of experimental error.
- c. **Define a Meaningful and Measurable Response or Dependent Variable.** Define what characteristic of the experimental units can be measured and recorded after each run. This characteristic should best represent the expected differences to be caused by changes in the factors.
- d. **List the Independent and Lurking Variables.** Declare which independent variables you wish to study. Ishikawa Cause-and-Effect Diagrams (see SAS Institute, (2004b)) are often useful at this step to help organize variables thought to affect the experimental outcome. Be sure that the independent variables chosen to study can be controlled during a single run and varied from run to run. If there is interest in a variable, but it cannot be controlled or varied, it cannot be included as a factor. Variables that are hypothesized to affect the response, but cannot be controlled, are lurking variables. The proper experimental design plan should prevent uncontrollable changes in these variables from biasing factor effects under study.
- e. **Run Pilot Tests.** Make some pilot tests to be sure you can control and vary the factors that have been selected, that the response can be measured, and that the replicate measurements of the same or similar experimental units are consistent. Inability to measure the response accurately or to control the factor levels are the main reasons that experiments fail to produce desired results. If the pilot tests fail, go back to steps 2, 3, and 4. If these tests are successful, measurements of the response for a few replicate tests with the same levels of the factors under study will produce data that can be used to get a preliminary estimate of the variance of experimental error.
- f. **Make a Flow Diagram of the Experimental Procedure for Each Run.**

This will make sure the procedure to be followed is understood and will be standardized for all runs in the design.

- g. **Choose the Experimental Design.** Choose an experimental design that is suited for the objectives of your particular experiment. This will include a description of what factor levels will be studied and will determine how the experimental units are to be assigned to the factor levels or combination of factor levels if there are more than one factor. One of the plans described in this book will almost always be appropriate. The choice of the experimental design will also determine what model should be used for analysis of the data.
- h. **Determine the Number of Replicates Required.** Based on the expected variance of the experimental error and the size of a practical difference, the researcher should determine the number of replicate runs that will give a high probability of detecting an effect of practical importance.
- i. **Randomize the Experimental Conditions to Experimental Units.** According to the particular experimental design being used, there is a proscribed method of randomly assigning experimental conditions to experimental units. In some designs, factor levels or combination of factor levels are assigned to experimental units completely at random. In other designs, randomizing factor levels is performed separately within groups of experimental units and may be done differently for different factors. The way the randomization is done affects the way the data should be analyzed, and it is important to describe and record exactly what has been done. The best way to do this is to provide a data collection worksheet arranged in the random order in which the experiments are to be collected. For more complicated experimental designs Bisgaard (1999) recommends one sheet of paper describing the conditions of each run with blanks for entering the response data and recording observations about the run. All these sheets should then be stapled together in booklet form in the order of which they are to be performed.
- j. **Describe a Method for Data Analysis.** This should be an outline of the steps of the analysis. An actual analysis of simulated data is often useful to verify that the proposed outline will work.
- k. **Timetable and Budget for Resources Needed to Complete the**

**Experiments.** Experimentation takes time and having a schedule to adhere to will improve the chances of completing the research on time. Bisgaard (1999) recommends a Gantt Chart (see SAS Institute, (2004a)), which is a simple graphical display showing the steps of the process as well as calendar times. A budget should be outlined for expenses and resources that will be required.

## **1.9. Performing the Experiments**

In experimentation, careful planning and execution of the plan are the most important steps. As we know from Murphy's Law, if anything can go wrong it will, and analysis of data can never compensate for botched experiments. To illustrate the potential problems that can occur, consider a simple experiment conducted by an amateur gardener described by Box *et al.* (1978). (In **Section 1.11**, we'll show that early DOE grew out of agriculture.) The purpose was to determine whether a change in the fertilizer mixture would result in a change in the yield of his tomato plants. Eleven tomato plants were planted in a single row, and the fertilizer type (*A* or *B*) was varied. The experimental unit in this experiment is the tomato plant plus the soil it is planted in, and the treatment factor is the type of fertilizer applied. Easterling (2004) discusses some of the nuances that we should consider when planning and carrying out an experiment. It is useful to think about these in context with the checklist presented **Section 1.9**.

When defining the objectives for this test, the experimenter needs to think ahead to the possible effects of conclusions that he or she may draw. In this case, the possible conclusions are (1) deciding that the fertilizer has no effect on the yield of tomatoes, or (2) concluding that one fertilizer produces a greater yield. If the home gardener finds no difference in yield, he or she can choose to use the less expensive fertilizer. If he or she finds a difference, they will have to decide whether the increase in yield offsets any cost increase for the better fertilizer. This can help him or her determine how large a difference in yield they should look for and the number of tomato plants he or she should include in their study. The answer to this question, which is crucial in planning the experiment, would probably be much different for a commercial grower than for a backyard enthusiast.

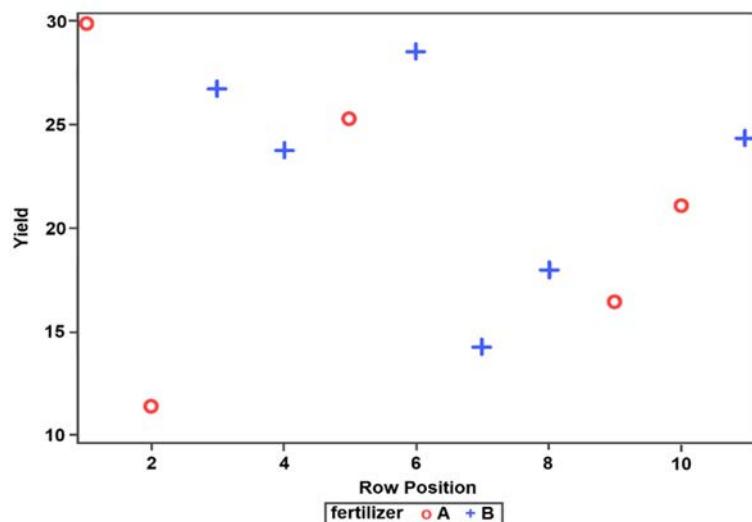
We determined that the experimental units for this experiment are the tomato plants. The experimenter should consider the similarity or homogeneity of plants and how far apart he or she is going to place the tomato plants in the soil. Will the spacing be far enough apart that the fertilizer applied to one plant does not bleed over and affect its neighbors?

Defining a meaningful response that can be measured may be problematic in this experiment. Not all the tomatoes on a single plant ripen at the same time. Thus, to measure the yield in terms of weight of tomatoes, the checklist and flow diagram describing how an experiment is conducted must be very precise. The response could be the weight of all tomatoes on the plant at a certain date, or the cumulative weight of tomatoes picked over time as they ripen. Precision in the definition of the response and consistency in adherence to the definition when making the measurements are crucial.

There are many possible lurking variables to consider in this experiment. Any differences in watering, weeding, insect treatment, the method and timing of fertilizer application, and the amount of fertilizer applied may certainly affect the yield; hence, the experimenter must pay careful attention to these variables to prevent bias. Easterling (2004) also pointed out that the row position seems to have affected the yield as well (as can be seen in **Figure 1-3**). The randomization of fertilizers to plants and row positions should equalize these differences for the two fertilizers. This was one of the things that Box *et al.* (1978) illustrated with this example. If a convenient method of applying the fertilizers (such as *A* at the beginning of the row followed by *B*) had been used in place of random assignment, the row position effect could have been mistaken for a treatment effect. Had this row position effect been known before the experiment was planned, the adjacent pairs of plots could have been grouped together in pairs, and one fertilizer assigned at random to one plant in each pair to prevent bias from the row position effect. This technique is called **blocking**, and we will discuss the details in Chapter 4.

Easterling (2004) also raised the question: why were only eleven plants used in the study (five fertilized with fertilizer A and six with fertilizer B)?

Normally flats of tomato plants purchased from a nursery come in flats of twelve. Was one plant removed from the study because it appeared unhealthy or got damaged in handling? The yield for the plant in the second-row position (see **Figure 1-3**) of the 11 plants used was considerably lower than the others planted in neighboring row positions with the same fertilizer. Was this plant unhealthy or damaged as well?



**Figure 1-3. Plot of Yield by Row Position—Tomato Experiment**

Any problems that arise during the conduct of experiments should be carefully observed, noted, and recorded as comments on the data collection form described in step 9 of the checklist. Perhaps if this had been done for the tomato experiment, the low yield at row position two could be explained.

This discussion of a very simple experiment helps to emphasize the importance of carefully considering each step of the checklist presented in **Section 1.8**, and the importance of strict adherence to a flowchart for conducting the experiments, described in step 6 of that checklist. Failing to consider each point of the checklist, and inconsistency in conducting experiments and recording results, may lead to the demise of an otherwise useful research project.

## **1.10. Use of R Software**

Fisher's original book on experimental designs laid the logical principles for experimentation, but users of experimental designs needed to have more detailed descriptions of the most useful designs along with accompanying plans. Consulting statisticians needed to have a systematic explanation of the relation between experimental designs and the statistical theory of least squares and linear hypotheses, and to have an account of designs and descriptions of experimental conditions where each design was most appropriate.

These needs were satisfied by Cochran and Cox (1950) and Kempthorne (1952) books. However, Cochran and Cox and Kempthorne's books were published before the age of computers and they both emphasize extensive tables of designs, abundant formulas, and numerical examples describing methods of manual analysis of experimental data and mathematical techniques for constructing certain types of designs. Since the publication of these books, use of experimental designs has gone far beyond agricultural research where it was initially employed, and a plethora of new books have been written on the subject. Even though computers and software (to both design and analyze data from experiments) are widely available, a high proportion of the more recent books on experimental design still follow the traditional pattern established by Cochran and Cox and Kempthorne by presenting extensive tables of designs and formulas for hand calculations and methods for constructing designs.

One of the objectives of this book is to break from the tradition and present computer code and output in place of voluminous formulas and tables. This will leave more room in the text to discuss the appropriateness of various design plans and ways to interpret and present results from experiments. The particular computer software illustrated in this book is R (R Development Core Team, 2003); (Ihaka & Gentleman, 1996). In addition to R programming statements that are useful for constructing experimental designs and base functions that are useful for the analysis of experimental data, there are many user-written packages that ease the construction of specific designs and provide analysis routines that are not available in the base R. These user-written packages

can be installed from CRAN. Packages illustrated in this book include: `agricolae`, `AlgDesign`, `BsMD`, `car`, `daewr`, `DoE.base`, `FrF2`, `GAD`, `gmodels`, `leaps`, `lme4`, `lsmeans`, `mixexp`, `multcomp`, and `Vdgraph`. Appendix provides a brief introduction to R and additional references on using R.

## 1.11. A Brief History

There have been four eras in the modern development of statistical experimental design. In the 1920s and early 1930s, Sir Ronald A. Fisher led the *agricultural era* with his pioneering work. During that time, Fisher was responsible for statistics and data analysis at the Rothamsted Agricultural Experimental Station near London, England. Fisher recognized that flaws in the way the experiment that generated the data had been performed often hampered the analysis of data from systems (in this case, agricultural systems). By interacting with scientists and researchers in many fields, he developed the insights that led to the three basic principles of experimental design that we discussed in **Section 1.7: randomization, replication, and blocking** (we will cover these in more detail in Chapter 2). Fisher systematically introduced statistical thinking and principles into designing experimental investigations, including the factorial design concept and the **analysis of variance (ANOVA)**. His two books [the most recent editions are Fisher (1958) (1966)] had profound influence on the use of statistics, particularly in agricultural and related life sciences. For an excellent biography of Fisher, see Box (1978).

Although applications of statistical design in industrial settings began in the 1930s, the second, or *industrial era* was catalyzed by the development of response surface methodology (RSM) by Box and Wilson (1951). They recognized and exploited the fact that many industrial experiments are fundamentally different from their agricultural counterparts in two ways: (1) the response variable can usually be observed (nearly) immediately, and (2) the experimenter can quickly learn crucial information from a small group of runs that can be used to plan the next experiment. Box (1999) calls these two features of industrial experiments **immediacy** and **sequentiality**. Over the next 30 years, RSM and other design techniques spread throughout the chemical and the process industries, mostly in research and development work.

George Box was the intellectual leader of this movement. However, the application of statistical design at the plant or manufacturing process level was still not extremely widespread. Some of the reasons for this include an inadequate training in basic statistical concepts and methods for engineers and other process specialists and the lack of computing resources and user-friendly statistical software to support the application of statistically designed experiments.

It was during this second or industrial era that work on optimal design of experiments began. Kiefer (1961) and Kiefer and Wolfowitz (1959) proposed a formal approach to selecting a design based on specific objective optimality criteria. Their initial approach was to select a design that would result in the model parameters being estimated with the best possible precision. This approach did not find much application because of the lack of computer tools for its implementation. However, there have been great advances in both algorithms for generating optimal designs and computing capability over the last 25 years. Optimal designs have great application and are discussed at several places in the book.

The increasing interest of Western industry in *quality improvement* that began in the late 1970s ushered in the third era of statistical design. The work of Genichi Taguchi [Taguchi and Wu (1980) and Taguchi (1987)] had a significant impact on expanding the interest in and use of designed experiments. Taguchi advocated using designed experiments for what he termed robust parameter design, or

1. Making processes insensitive to environmental factors or other factors that are difficult to control
2. Making products insensitive to variation transmitted from components
3. Finding levels of the process variables that force the mean to a desired value while simultaneously reducing variability around this value.

Taguchi suggested highly fractionated factorial designs and other orthogonal arrays along with some novel statistical methods to solve these problems. The resulting methodology generated much discussion and controversy. Part of the controversy arose because Taguchi's methodology was advocated in the West initially (and primarily) by

entrepreneurs, and the underlying statistical science had not been adequately peer reviewed. By the late 1980s, the results of peer review indicated that although Taguchi's engineering concepts and objectives were well founded, there were substantial problems with his experimental strategy and methods of data analysis. For specific details of these issues, see Box (1988), Box, Bisgaard, and Fung (1988), Hunter (1985) (1989), Myers, Montgomery and Anderson-Cook (2004), and Pignatiello and Ramberg (1992). Many of these concerns are also summarized in the extensive panel discussion in the May 1992 issue of *Technometrics* [see Nair (1992)].

There were several positive outcomes of the Taguchi controversy. First, designed experiments became more widely used in the discrete parts industries, including automotive and aerospace manufacturing, electronics and semiconductors, and many other industries that had previously made little use of the technique. Second, the fourth era of statistical design began. This era has included a renewed general interest in statistical design by both researchers and practitioners and the development of many new and useful approaches to experimental problems in the industrial world, including alternatives to Taguchi's technical methods that allow his engineering concepts to be carried into practice efficiently and effectively. Some of these alternatives will be discussed and illustrated in subsequent chapters, particularly in Chapter 12. Third, computer software for construction and evaluation of designs has improved greatly with many new features and capability. Fourth, formal education in statistical experimental design is becoming part of many engineering programs in universities, at both undergraduate and graduate levels. The successful integration of good experimental design practice into engineering and science is a key factor in future industrial competitiveness.

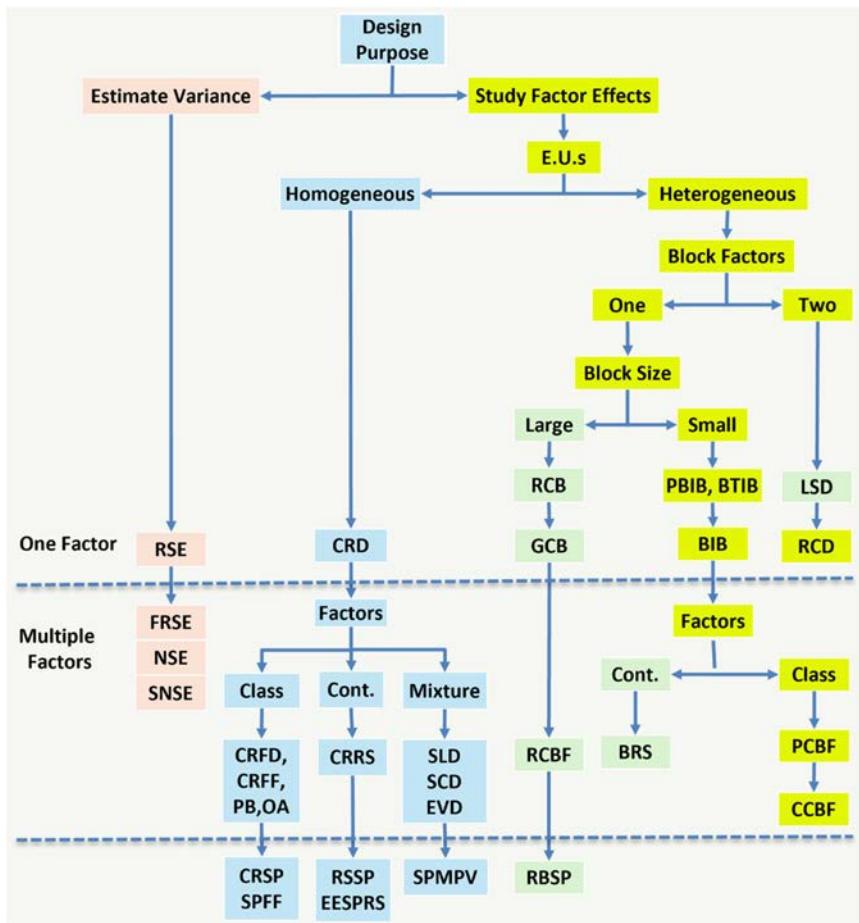
Applications of designed experiments have grown far beyond the agricultural origins. There is not a single area of science and engineering that has not successfully employed statistically designed experiments. In recent years, there has been a considerable utilization of designed experiments in many other areas, including the service sector of business, financial services, government operations, and many nonprofit

business sectors. An article appeared in Forbes magazine on March 11, 1996, entitled “The New Mantra: MVT (Koselka, 1996),” where MVT stands for “multivariable testing,” a term authors use to describe factorial designs. The article notes the many successes that a diverse group of companies have had through their use of statistically designed experiments.

## ***1.12. Review of Important Concepts***

This chapter describes the purpose for experimental designs. In order to determine if cause and effect relationships exist, an experimental design must be conducted. In an experimental design, the factors under study are purposely varied and the result is observed. This is different from observational studies or sampling surveys where data is collected with no attempt to control the environment. In order to predict what will happen in the future, when the environment is controlled, you must rely on cause-and-effect relationships. Relationships obtained from observational studies or sampling surveys are not reliable for predicting future results when the environment is to be controlled.

Experimental designs were first developed in agricultural research but are now used in all situations where the scientific method is applied. The basic definitions and terminology used in experimental design are given in this chapter along with a checklist for planning experiments. In practice there are many different types of experimental designs that can be used. Which design is used in a particular situation depends upon the research objectives and the experimental units. **Figure 1-4** is a diagram that illustrates when the different experimental designs described in this book should be used. As different experimental designs are presented in chapters to follow, reference will be made back to this figure to describe when the designs should be used.



**Figure 1-4. Design Selection Roadmap**

**Table 1-2. Design Name Acronym Index**

Abbr	Description
RSE	random sampling experiment
FRSE	factorial random sampling experiment
NSE	nested sampling experiment
SNSE	staggered nested sampling experiment
CRD	completely randomized design
CRFD	completely randomized factorial design
CRFF	completely randomized fractional factorial
PB	Plackett-Burman design

OA	orthogonal array design
CRSP	completely randomized split plot
RSSP	response surface split plot
EESPRS	equivalent estimation split-plot response surface
SLD	simplex lattice design
SCD	simplex centroid design
EVD	extreme vertices design
SPMPV	split-plot mixture process variable design
RCB	randomized complete block
GCB	generalized complete block
RCBF	randomized complete block factorial
RBSP	randomized block split plot
PBIB	partially balanced incomplete block
BTIB	balanced treatment incomplete block
BIB	balance incomplete block
BRS	blocked response surface
PCBF	partially confounded blocked factorial
CCBF	completely confounded blocked factorial
LSD	Latin-square design
RCD	row-column design

## 2. Completely Randomized Designs - One Factor

---

### 2.1. Introduction

In a **completely randomized design**, abbreviated as **CRD**, with one treatment factor,  $n$  experimental units are divided randomly into  $t$  groups. Each group is then subject to one of the unique levels or values of the treatment factor. If  $n = tr$  is a multiple of  $t$ , then each level of the factor will be applied to  $r$  unique experimental units, and there will be  $r$  replicates of each run with the same level of the treatment factor. If  $n$  is not a multiple of  $t$ , then there will be an unequal number of replicates of each factor level. We hold constant all other known independent variables so that they will not bias the effects. We use this design when there is only one factor under study and the experimental units are **homogeneous**.

### 2.2. Example - Aerogels for Space Operations

For example, silica aerogels are among the lowest density, lowest thermal conductivity materials known, with many possible aeronautics and space applications, including vehicles, habitats, planetary rovers, heat shields, and space suits. In an experiment to determine the effect of time to heat a sheet of aerogel with a functional epoxy on the density of modified aerogel, one homogeneous batch of modified silica aerogel would be divided into  $n$  parts with an equal amount of aerogel and epoxy in each. The parts of modified aerogel would then be divided randomly into  $t$  groups. Each group would be allowed to heat at a constant temperature for a unique time, and the density of the modified aerogel would be measured and recorded for each part. The treatment factor would be the heating time, the experimental unit would be an individual portion of modified aerogel, and the response would be the measured density. Although other factors, such as levels of temperature the factor, are known to affect the density of the resulting aerogel, they would be held constant, and each portion of aerogel would be allowed to react under the same conditions except for the differing heating times.

#### 2.2.1. Replication

Replication and randomization were popularized by Fisher. These are

the first techniques that fall in the category of **error control** that was briefly explained in **Section 1.6**.

The technique of replication dictates that  $r$  aerogel sheets are tested at each of the  $t$  heating times rather than a single sheet at each heating time. By having replicate experimental units in each level of the treatment factor, the variance of the experimental error can be calculated from the data, and this variance will be compared to the treatment effects. If the variability among the treatment means is not larger than the **experimental error variance**, the treatment differences are probably due to differences of the experimental units assigned to each treatment. Without replication it is impossible to tell if treatment differences are real or just a random manifestation of the particular experimental units used in the study. Subsamples or duplicate measurements, described in Chapter 1, cannot substitute for replicates.

### 2.2.2. Randomization

The random division of experimental units into groups is called **randomization**, and it is the procedure by which the validity of the experiment is guaranteed against biases caused by other lurking variables. In the aerogel experiment randomization would prevent lurking variables, such as variability in crosslinking concentration from sheet to sheet and trends in the measurement technique over time, from biasing the effect of the heating time.

When experimental units are randomized to treatment factor levels, an exact test of the hypothesis that the treatment effect is zero can be accomplished using a randomization test, and a test of parameters in the general linear model, normally used in the analysis of experimental data, is a good approximation to the randomization test.

A simple way of constructing a randomized data collection form, dividing  $n$  experimental units into  $t$  treatment groups, can be accomplished using base R commands. For example, in the aerogel experiment, if the experimenter wants to examine three different treatment times (35 minutes, 40 minutes, and 45 minutes) and test four replicate sheets of modified aerogel at each heating time, the following code will create the list.

```
set.seed(7638)
f <- factor( rep( c(35, 40, 45), each = 4 ))
fac <- sample( f, 12 )
eu <- 1:12
plan <- data.frame( aerogel = eu, time = fac )
write.csv( plan, file = "Plan.csv", row.names = FALSE)
```

The R command `factor` creates a vector of the factor levels for (heating time) and stores it in the variable `f`. There is also an ordered command in R that creates a factor that is assumed to have equally spaced numerical levels. R handles factors created by the `factor` and ordered commands differently when making comparisons of treatments after fitting a model. There will be more discussion of this in [Section 2.8](#).

The `sample` function randomizes the order of the factor levels and stores the randomized vector in the variable `fac`. The `seq` function creates a numeric vector of experimental unit (i.e., aerogel sheet) numbers (`eu`). Next, the `data.frame` function combines the two vectors `eu`, `fac` as columns that are stored in the data frame object `plan` with column headings `Aerogel`, and `Time`. Finally, the `write.csv` function writes the data frame to a .csv file called `Plan.csv`. This file can be found in your working directory. To get the path to your working directory, type the command `getwd()` at the R prompt (you can also specify your working directory with the command `setwd()`). Opening `Plan.csv` in a spreadsheet program like Microsoft Excel or Open Office Calc and adding an extra column (as shown in [Figure 2-1](#)) results in a convenient electronic data collection form.

This form shows us that the first aerogel sheet, or experimental unit, should be allowed to heat 35 minutes, the second sheet 45 minutes, etc. If you run the same commands in R repeatedly, you will get the same random order because of the `set.seed` statement. Remove this statement to get a different random order.

In addition to the base R commands shown above, several user written R packages can create randomized lists of experiments, which can be conveniently converted into electronic data collection forms. However, these packages will be illustrated for creating more complicated designs

in forthcoming chapters and will not be shown here.

	A	B	C
1	Aerogel	Time	Density
2		1	35
3		2	45
4		3	40
5		4	40
6		5	35
7		6	40
8		7	35
9		8	45
10		9	35
11		10	45
12		11	45
13		12	40

Figure 2-1. Data Collection Form in a Spreadsheet

### 2.3. A Historical Example

To illustrate the checklist for planning an experiment described in **Section 1.8**, consider a historical example taken from the 1937 *Rothamsted Experimental Station Report* (Johnston, 1986). This illustrates some of the early work done by Fisher in developing the ideas of experimental design and analysis of variance for use on agricultural experiments at the research station. (Since designed experimentation was historically limited to agriculture, we have to provide an example from that domain.)

**Objectives.** The objective of the study was to compare the times of planting, and methods of applying mixed artificial fertilizers (NPK) prior to planting, on the yield of sugar beets. Normally fertilizer is applied, and seeds planted as early as the soil can be worked.

**Experimental Units.** The experimental units were the plots of ground in combination with specific seeds to be planted in each plot of ground.

**Response or Dependent Variable.** The dependent variable would be the yield of sugar beets measured in cwt per acre.

**Independent Variables and Lurking Variables.** The independent variables of interest were the time and method of applying mixed artificial fertilizers. Four levels of the treatment factor were chosen as listed below:

1. (A) no artificial fertilizers applied
2. (B) artificial applied in January (plowed)
3. (C) artificial applied in January (broadcast)
4. (D) artificial applied in April (broadcast)

Lurking variables that could cause differences in the sugar beet yields between plots were differences in the fertility of the plots themselves, differences in the beet seeds used in each plot, differences among plots in the level of weed infestation, differences in cultivation practices of thinning the beets, and hand harvesting the beets.

**Pilot Tests.** Sugar beets had been grown routinely at *Rothamsted*, and artificial fertilizers had been used by both plowing and broadcast for many crop plants; therefore, it was known that the independent variable could be controlled, and that the response was measurable.

**Choose Experimental Design.** The completely randomized design (CRD) was chosen so that differences in lurking variables between plots would be unlikely to correspond to changes in the factor levels listed above.

**Determine the Number of Replicates.** A difference in yield of 6 cwt per acre was considered to be of practical importance and based on historical estimates of variability in sugar beet yields at *Rothamsted*, four or five replicates were determined to be sufficient.

**Randomize Experimental Units to Treatment Levels.** Eighteen plots were chosen for the experiment, and a randomized list was constructed assigning four or five plots to each factor level.

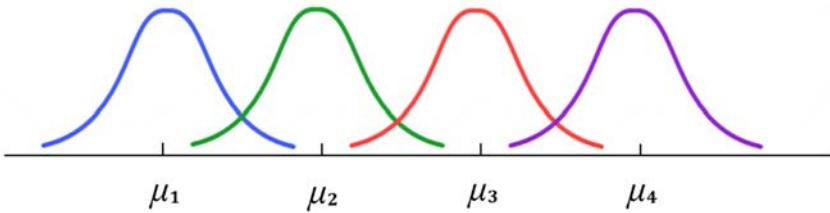
## 2.4. Linear Model for CRD

The mathematical model for the data from a CRD, or completely randomized design, with an unequal number of replicates for each factor level can be written as:

$$Y_{ij} = \mu_i + E_{ij} \quad \text{Eq. 2-1}$$

where  $Y_{ij}$  is the response for the  $j$ th experimental unit subject to the  $i$ th level of the treatment factor,  $i = 1, \dots, t, j = 1, \dots, r_i$ , and  $r_i$  is the number of experimental units or replications in  $i$ th level of the treatment factor.

This is sometimes called the **cell means model** with a different mean,  $\mu_i$ , for each level of the treatment factor. The distribution of the experimental errors,  $\varepsilon_{ij}$ , are mutually independent due to the randomization and assumed to be normally distributed. This model is graphically represented in **Figure 2-2**.



**Figure 2-2. Cell Means Model**

An alternate way of writing a model for the data is

$$Y_{ij} = \mu + \tau_i + \varepsilon_{ij}. \quad \text{Eq. 2-2}$$

This is called the **effects model** and the  $\tau_i$ s are called the effects.  $\tau_i$  represents the difference between the long-run average of all possible experiments at the  $i$ th level of the treatment factor and the overall average. With the normality assumption  $Y_{ij} \sim N(\mu + \tau_i, \sigma^2)$  or  $\varepsilon_{ij} \sim N(0, \sigma^2)$ . For equal number of replicates, the sample means of the data in the  $i$ th level of the treatment factor is represented by

$$\bar{y}_{i\cdot} = \frac{1}{r_i} \sum_{j=1}^{r_i} y_{ij} \quad \text{Eq. 2-3}$$

and the grand mean is given by

$$\bar{y}_{\cdot\cdot} = \frac{1}{t} \sum_{i=1}^t \bar{y}_{i\cdot} = \frac{1}{n} \sum_{i=1}^t \sum_{j=1}^{r_i} y_{ij} \quad \text{Eq. 2-4}$$

where  $n = \sum r_i$ . Using the **method of maximum likelihood**, which is

equivalent to the **method of least squares** with these assumptions, the estimates of the cell means are found by choosing them to minimize the **error sum of squares**

$$sse = \sum_{i=1}^t \sum_{j=1}^{r_i} (y_{ij} - \mu_i)^2 \quad \text{Eq. 2-5}$$

This is done by taking partial derivatives of  $sse$  with respect to each cell mean, setting the results equal to zero, and solving each equation

$$\frac{\partial sse}{\partial \mu_i} = -2 \sum_{i=1}^t \sum_{j=1}^{r_i} (y_{ij} - \mu_i) = 0$$

This results in the estimates:

$$\hat{\mu}_i = \bar{y}_i.$$

#### 2.4.1. Matrix Representation

Consider a CRD with  $t = 3$  factor levels and  $r_i = 4$  replicates for  $i = 1, \dots, t$ . We can write the effects model concisely using matrix notation as:

$$y = X\beta + \varepsilon \quad \text{Eq. 2-6}$$

where

$$y = \begin{bmatrix} y_{11} \\ y_{12} \\ y_{13} \\ y_{14} \\ y_{21} \\ y_{22} \\ y_{23} \\ y_{24} \\ y_{31} \\ y_{32} \\ y_{33} \\ y_{34} \end{bmatrix}, \quad X = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}, \quad \beta = \begin{bmatrix} \mu \\ \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix}, \quad \varepsilon = \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{12} \\ \varepsilon_{13} \\ \varepsilon_{14} \\ \varepsilon_{21} \\ \varepsilon_{22} \\ \varepsilon_{23} \\ \varepsilon_{24} \\ \varepsilon_{31} \\ \varepsilon_{32} \\ \varepsilon_{33} \\ \varepsilon_{34} \end{bmatrix},$$

and  $\varepsilon \sim MVN(0, \sigma^2 I)$ .

The least squares estimators for  $\beta$  are the solution to the normal equations  $X'X\beta = X'y$ . The problem with the normal equations is that

$X'X$  is singular and cannot be inverted. Using the treatment coding for an unordered factor created with the `factor` command, the R function `lm` makes the  $X$  matrix full rank by dropping the column that corresponds to the first level of the factor as shown below.

$$X = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

This treatment coding makes the first level of the factor the standard, and all other levels of the factor are compared to it. For the example with  $t = 3$  factor levels the solution to the normal equations is

$$(X'X)^{-1}X'y = \hat{\beta} = \begin{bmatrix} \hat{\mu} + \hat{\tau}_1 \\ \hat{\tau}_2 - \hat{\tau}_1 \\ \hat{\tau}_3 - \hat{\tau}_1 \end{bmatrix}$$

#### 2.4.2. LS Calculations with R Function lm

**Table 2-1** shows the data from a CRD design for the Aerogel Density experiment described earlier in this chapter.

*Table 2-1. Data from Aerogel Density Experiment*

Treatment Time	Aerogel Density
35 minutes	4.5, 5.0, 5.5, 6.75
40 minutes	6.5, 6.5, 10.5, 9.5
45 minutes	9.75, 8.75, 6.5, 8.25

Using these data we have

$$X'X = \begin{bmatrix} 12 & 4 & 4 \\ 4 & 4 & 0 \\ 4 & 0 & 4 \end{bmatrix}, X'y = \begin{bmatrix} 88.0 \\ 33.0 \\ 33.25 \end{bmatrix}$$

and

$$(X'X)^{-1} = \begin{bmatrix} 0.25 & -0.25 & -0.25 \\ -0.25 & 0.50 & 0.25 \\ -0.25 & 0.25 & 0.50 \end{bmatrix},$$

$$\beta = (X'X)^{-1}X'y = \begin{bmatrix} \hat{\mu} + \hat{\tau}_1 \\ \hat{\tau}_2 - \tau_1 \\ \hat{\tau}_3 - \hat{\tau}_1 \end{bmatrix} = \begin{bmatrix} 5.4375 \\ 2.8135 \\ 2.8750 \end{bmatrix}$$

If the data had been collected and typed into the electronic spreadsheet shown in **Figure 2-1** and resaved as a `.csv` file, then it could be read back into an R data frame called `aerogel_time` with the following command.

```
aero_time <- read.csv("https://raw.githubusercontent.com
/stricje1/Data/master/aerogel_time.csv")
```

The R package `daewr` contains example data sets and several R functions.

```
library(daewr)
mod0 <- lm( Density ~ Time, data = aero_time )
summary( mod0 )
```

The command `library(daewr)` makes this package available, but before this command can be issued the package must be installed as described in the **Appendix A**. The `lm` command fits the linear model and stores the results in the object `mod0`, and the `summary` command prints the results, a portion of which is shown below.

Coefficients:					
	Estimate	Std. Error	t value	Pr(> t )	
(Intercept)	5.4375	0.7655	7.104	5.65e-05	***
time40	2.8125	1.0825	2.598	0.0288	*
time45	2.8750	1.0825	2.656	0.0262	*
---					
Signoff. codes:			0.001 **	0.01 * 0.05 .	1
				0.1	
	0 ***				

Residual standard error: 1.531 on 9 degrees of freedom Multiple R-squared: 0.5056, Adjusted R-squared: 0.3958  
F-statistic: 4.602 on 2 and 9 DF, p-value: 0.042

Since the variable `time` is a factor in the data frame `aero_time`, and default treatment coding was used by function `lm`, the estimates described above are produced.

### 2.4.3. Estimation of $\sigma^2$ and Distribution of Quadratic Forms

The estimate of the variance of the experimental error,  $\sigma^2$ , is  $\frac{ssE}{n-t}$ . It is only possible to estimate this variance when there are replicate experiments at each level of the treatment factor. When measurements on sub-samples or duplicate measurements on the same experimental unit are treated as replicates, this estimate can be seriously biased.

In matrix form,  $ssE$  can be written as

$$ssE = \mathbf{y}'\mathbf{y} - \hat{\boldsymbol{\beta}}'\mathbf{X}'\mathbf{y} = \mathbf{y}'(\mathbf{I} - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}')\mathbf{y}$$

and from the theory of linear models it can be shown that the ratio of  $ssE$  to the variance of the experimental error,  $\sigma^2$ , follows a **chi-square** distribution with  $n - t$  degrees of freedom, that is,  $\frac{ssE}{\sigma^2} \sim \chi^2$ .

### 2.4.4. Estimable Functions

A linear combination of the cell means is called an **estimable function** if it can be expressed as the expected value of a linear combination of the responses, that is,

$$\sum_{i=1}^t b_i(\mu + \tau_i) = E \left[ \sum_{i=1}^t \sum_{j=1}^{r_i} a_{ij} Y_{ij} \right] \quad \text{Eq. 2-7}$$

From this definition it can be seen that effects,  $\tau_i$ , are not estimable, but a cell mean,  $\mu + \tau_i$ , or a contrast of effects,  $\sum c_i \tau_i$ , where  $\sum c_i = 0$ , is estimable.

In matrix notation  $\mathbf{L}\boldsymbol{\beta}$  is a set of estimable functions if each row of  $L$  is a linear combination of the rows of  $X$ , and  $\mathbf{L}\hat{\boldsymbol{\beta}}$  is its unbiased estimator.  $\mathbf{L}\hat{\boldsymbol{\beta}}$  follows the multivariate **normal distribution** with covariance matrix  $\sigma^2 \mathbf{L}'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{L}$ , using the data from the Aerogel Density experiment above,

$$\mathbf{L} = \begin{bmatrix} 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \quad \text{Eq. 2-8}$$

$$\mathbf{L}\boldsymbol{\beta} = \begin{bmatrix} \tau_1 - \tau_2 \\ \tau_1 - \tau_3 \end{bmatrix}, \text{ and } \mathbf{L}\hat{\boldsymbol{\beta}} = \begin{bmatrix} \hat{\tau}_1 - \hat{\tau}_2 \\ \hat{\tau}_1 - \hat{\tau}_3 \end{bmatrix} = \begin{bmatrix} 2.8025 \\ 2.8750 \end{bmatrix}$$

is a vector of contrasts of the effects. The number of **degrees of freedom**, or number of linearly independent contrasts of effects in a CRD, is always the number of levels of the treatment factor minus one, that is,  $t - 1$ . Whenever there is a set of  $t - 1$  linearly independent contrasts of the effects, they are called a **saturated set of estimable contrasts**.

It can be shown that  $(\mathbf{L}\hat{\boldsymbol{\beta}})'(\mathbf{L}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{L}')^{-1}(\mathbf{L}\hat{\boldsymbol{\beta}})$  follows the noncentral chi-square distribution,  $\chi^2(p, \lambda)$  where the noncentrality parameter

$$\lambda = (\sigma^2)^{-1}(\mathbf{L}\boldsymbol{\beta})'(\mathbf{L}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{L}')^{-1}(\mathbf{L}\boldsymbol{\beta}),$$

and  $\mathbf{L}$  is the **coefficient matrix** for an estimable contrast like *Eq. 2-8*, and the degrees of freedom  $p$  is equal to the rank of  $\mathbf{L}$ .

Estimable contrasts can be obtained from the `fit.contrast` function in the R package `gmodels`, (Warnes, 2012). First, install the `gmodels` package as described in the appendix, then we can load the package and call the function as shown in the example code below. There we use it to estimate the average difference in the cell means for the first and second levels of the treatment factor,  $(\mu + \tau_1) - (\mu + \tau_2) = \tau_1 - \tau_2$ .

```
library(gmodels)
fit.contrast( mod0, "time", c(1, -1, 0) )
```

In the function call above, the `mod0` is the name of a model previously fit with the R function `lm`, the string in quotes is the name of the factor in the model whose cell means are compared, and the vector `c(1, -1, 0)` are the contrast coefficients,  $c_i$ . This produces the result  $(-2.8125)$ , which is the negative of the second estimate produced in the R output on the previous page using the default treatment coding in the model `mod0`.

#### 2.4.5. Hypothesis Test of No Treatment Effects

In the model for the CRD, the statistical hypothesis of interest is  $H_0$  :  $\mu_1 = \mu_2 = \dots = \mu_t$  or  $\tau_1 = \tau_2 = \dots = \tau_r$  versus the alternative  $H_a$ : at least two of the  $\tau$ s differ. If the **null hypothesis** is true, the model  $y_{ij} = \mu_i + \varepsilon_{ij} = \mu + \tau_i + \varepsilon_{ij}$  simplifies to  $y_{ij} = \mu + \varepsilon_{ij}$ , which can be represented as a single normal distribution with mean  $\mu$  and variance  $\sigma^2$  rather than multiple normal distributions like those shown in **Figure 2-2**.

The **sums of squares about the mean** is  $ssTotal = \sum_{i=1}^t \sum_{j=1}^{r_i} (y_{ij} - \bar{y}_{..})^2 = \mathbf{y}'\mathbf{y} - \frac{(\mathbf{1}'\mathbf{y})^2}{\mathbf{1}'\mathbf{1}}$ , where  $\bar{y}_{..}$  is the grand mean and  $\mathbf{1}$  is a column vector of ones. This sum of squares can be partitioned as:

$$ssTotal = ssT + ssE \quad \text{Eq. 2-9}$$

where  $ssT = \widehat{\boldsymbol{\beta}}'\mathbf{X}'\mathbf{y} - \frac{(\mathbf{1}'\mathbf{y})^2}{\mathbf{1}'\mathbf{1}} = (\mathbf{L}\widehat{\boldsymbol{\beta}})'(\mathbf{L}(\mathbf{X}'\mathbf{X})^{(-1)}\mathbf{L}')^{-1}(\mathbf{L}\widehat{\boldsymbol{\beta}})$ , and the coefficient matrix for a saturated set of estimable contrasts. This quantity is called the **treatment sums of squares**. Under the null hypothesis  $H_0: \mu_1 = \mu_2 = \dots = \mu_t$ , both  $ssT$  and  $ssE$  follow the chi-squared distribution. These sums of squares and their corresponding **mean squares**, which are formed by dividing each sum of squares by its degrees of freedom, are usually presented in an **analysis of variance** or **ANOVA** table like that shown symbolically in **Table 2-2**.

**Table 2-2. Analysis of Variance Table**

Source	df	Sum of Squares	Mean Squares	F-ratio
Treatment	$t - 1$	$ssT$	$msT$	$F = \frac{msT}{msE}$
Error	$n - t$	$ssE$	$msE$	
Total	$n - 1$	$ssTotal$	$msTotal$	

Under the null hypothesis, the **F-ratio**  $msT/msE$  follows the **F-distribution** with  $t - 1$  and  $n - t$  degrees of freedom, and under the alternative it follows the noncentral F distribution with noncentrality parameter

$$\lambda = (\sigma^2)^{-1}(\mathbf{L}\boldsymbol{\beta})'(\mathbf{L}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{L}')^{-1}(\mathbf{L}\boldsymbol{\beta}) = \frac{r}{\sigma^2} \sum_{i=1}^t (\mu_i - \bar{\mu}_.)^2$$

It is the **generalized likelihood ratio** test statistic for  $H_0$ , and is the formal method of comparing the treatment effects to the experimental error variance described in **Section 2.2.1**.

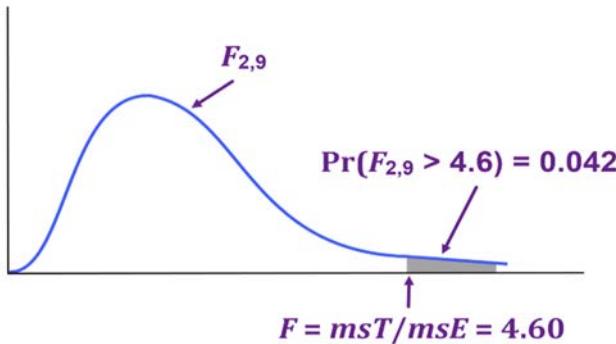
The sums of squares, mean squares, degrees of freedom in the ANOVA table, and associated  $F$ -test statistic can be calculated by the `aov` function in R. The inputs to the `aov` function are the same as those for the `lm` function shown earlier, but the summary of an object created by the `aov` function is the ANOVA table rather than the estimates produced by the `lm` function. The code to produce the ANOVA table for the aerogel sheet density experiment is:

```
mod1 <- aov( Density ~ Time, data = aerogel_df )
summary(mod1)
```

The resulting ANOVA table is shown below.

	Df	Sum Sq	Mean Sq	F-value	Pr(>F)
time	2	21.57	10.786	4.602	0.042 *
Residuals	9	21.09	2.344		
---					
Signif. codes:			0 ***	0.001 **	0.01 * 0.05 . 0.1
			1		

In this table the  $ssT$  and  $msT$  and the associated degrees of freedom are on the line labeled `time`, the  $ssE$  is on the line labeled `Residuals`, and the  $ssTotal$  can be computed by adding the  $ssT$  to the  $ssE$ . The  $F$ -value is the ratio  $msT/msE$  and the last column labeled `Pr(>F)` is the probability of exceeding the calculated  $F$ -value if the null hypothesis is true. This is called the **p-value** and is illustrated graphically in **Figure 2-3**. If the experimenter chooses the **significance level**,  $\alpha$ , for his or her hypothesis test, will reject the hypothesis if the `Pr(>F)` value on the `aov` output is less than the chosen value of  $\alpha$ .



*Figure 2-3. Graph illustrating  $Pr > F$*

For the Aerogel Density experiment there are significant differences among the mean aerogel densities for each heating time at the significance level  $\alpha = 0.05$ , since  $0.042 < 0.05$ .

#### 2.4.6. A Word of Caution

When a completely randomized design in one factor is conducted, the model for analysis is or **Eq. 2-2** and the correct analysis is through the analysis of variance as shown symbolically in **Table 2-2**. The use of computer software like R makes it easy to analyze data and draw conclusions; however, if the experiment was not properly conducted even a sophisticated analysis of the data could be useless. The  $\varepsilon_{ij}$  term in the model **Eq. 2-1** or **Eq. 2-2**, and its associated sums of squares,  $ssE$ , represents replicate experimental units. In many cases experimenters do not have replicate experimental units in each level of the treatment factor and substitute subsamples or duplicates for them in the analysis. In other cases the experimental units are not properly randomized to treatment factor levels. When this is the situation, performing the analysis as if the design had been properly conducted may be completely wrong and misleading. Wrong conclusions can be drawn that do not hold up to later scrutiny, and a bad reputation is unfairly ascribed to statistically designed experiments and statistical analyses of data.

For example, I performed an experiment in the Department of Mathematical Sciences at the United States Military Academy to determine the effect of teaching methods on student test scores. When I first set up my experiment, I used one teaching method for the morning class, another for my evening class, and treated test scores for individual

students as replicates. The results of my analysis was totally wrong.

The experimental unit is the class, since I applied the teaching method to a whole class simultaneously, and the individual students are sub-samples or observational units (since I had to test individual students, not the class as a whole). The treatment effect should be judged against the variability in experimental units or classes. The variability among students in a class may be much different than variability from class average to class average. Sub-sample observations should be averaged before analysis, as explained in **Section 1.5**. By doing this, I had only one observation per class per teaching method and no replicates for use in calculating  $ssE$  in **Table 2-2**. There was no denominator for calculating the  $F$ -test statistic for teaching method. If I used the variability in students within a class to calculate  $ssE$ , it may be too large or too small, causing me to reach the wrong conclusion about the significance of the treatment effect. Further, if I did not randomize which teaching method was used in the morning and evening classes, and I had no replicate classes that were taught with the same teaching method, so my analysis was wide open to biases. Students in the morning classes may be fundamentally different than students in the evening classes (and they were), and any difference in average scores between the two teaching methods may be entirely due to differences among the two groups of students. In fact, I knew there are differences between morning and evening students, so I unknowingly used the teaching method I wants to promote in a better class, thus ruining the objectivity of my research. Needless to say, I had to start the experiment over the next semester and make the necessary adjustments to the study design.

## 2.5. Verifying Assumptions of the Linear Model

Two assumptions required for validity of the analysis based on the linear model presented in **Section 2.4** are **constancy** of the variance of the experimental error,  $\sigma^2$ , across all levels of the treatment factor, and **normality** of the experimental errors. To verify these assumptions, simple graphs can be constructed. A scatter plot of the model residuals versus the factor levels can show whether the variability seen at each level of the factor is approximately equal. The **model residuals** are the differences of the responses  $y_{ij}$  in each cell (or level of the factor) and

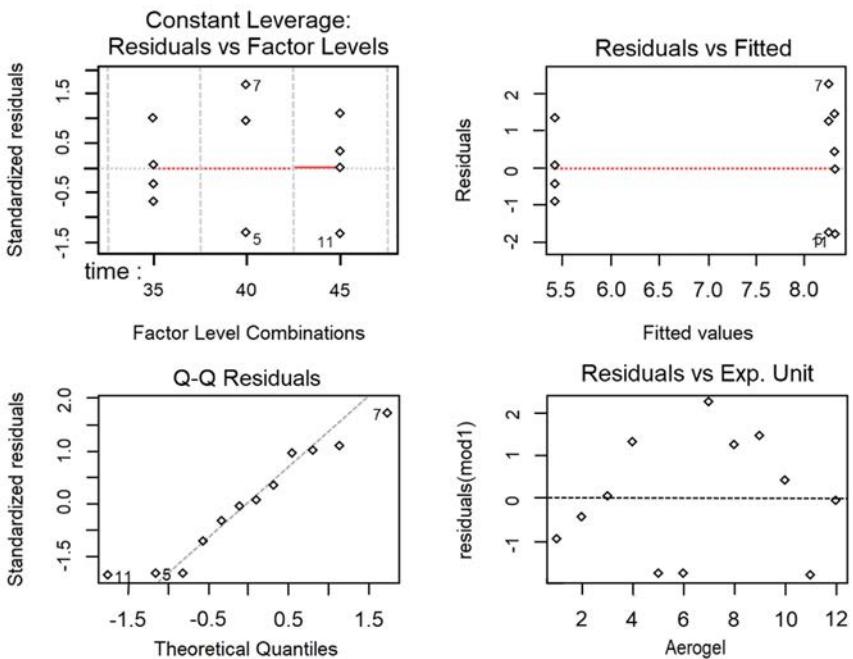
the cell means,  $\mu_i$ . When the variance differs between factor levels, it is often because the variability in the response tends to increase when the mean level of the response increases. A graph that can reveal this tendency is a plot of the model residuals versus the cell means or predicted values. Finally, the normality of the experimental errors can be checked by making a normal probability plot of the model residuals.

The most critical assumption justifying the analysis based on the linear model is independence of the experimental error terms  $\varepsilon_{ij}$ . This assumption is justified if proper randomization of the experimental units to treatment factor levels has been performed and true replicates are included. A simple scatter plot of the model residuals versus the experimental unit number can reveal inadequacies in the randomization. If there is an increasing, decreasing, or cyclical pattern in this plot, it could indicate the randomization did not balance heterogeneous experimental units across levels of the factor.

The four plots used to verify the assumptions of the linear model can be easily made using R. The code below produces these plots.

```
par( mfrow = c(2,2) ) #sets up four plot quadrants
plot(mod1, which=5)
plot(mod1, which=1)
plot(mod1, which=2)
plot(residuals(mod1) ~ Aerogel,
     main = "Residuals vs Exp. Unit", font.main = 1,
     data = aero_df)
abline(h = 0, lty = 2)
```

In this code, the R command `par(mfrow=c(2,2))` splits the plot region into four subregions. The resulting plots are arranged row-wise in **Figure 2-4**. The command `plot(mod1, which=5)` produces a plot of the standardized residuals versus the factor levels in the upper left.



**Figure 2-4.** Graphs used to verify the assumptions of a Linear Model

The command `plot(mod1, which=1)` produces the plot of residuals versus the cell means or fitted values in the top right. The command `plot(mod1, which=2)` produces the normal probability plot of the standardized residuals in the lower left. The final plot statement produces the plot of residuals versus experimental unit numbers in the lower right. In this plot statement, the `residuals(mod1)` extracts the residuals from the object `mod1` that was calculated by the `aov` function. A complete list of the quantities calculated by the `aov` function can be obtained by typing the command `names(mod1)` in the R console.

If the variability of the residuals differs between the factor levels in the plot in the upper left of **Figure 2-4**, it would indicate the variance of the  $\varepsilon_{ij}$ 's is not constant. With only four replicates in each cell this is difficult to determine. The plot of residuals versus cell means, shown in the upper right of **Figure 2-4**, may indicate that the variability in the residuals increases as the cell mean increases, but it is not clear. A better way to determine if this is the case will be shown in the next **Section 2.6**. The

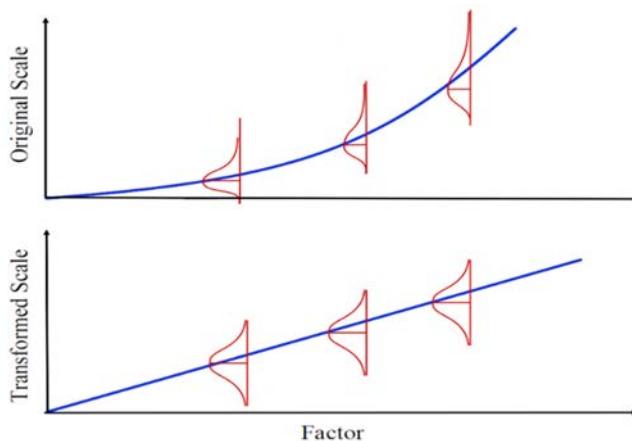
normal probability plot of residuals in the lower left justifies the normality assumption concerning the  $\varepsilon_{ij}$ 's if the points fall along a straight line. When there is more data and more points on the plot, the points must lie closer to a straight line to justify this assumption. In the normal plot in **Figure 2-4**, the points fall away from the line in the lower left and the upper right possibly indicating short tails in the distribution of residuals, but again it is difficult to determine with only 12 data points on the graph. The equal variance assumption is more critical than the normality assumption, but they sometimes go hand in hand. When the equal variance assumption is violated, the normality assumption is often violated as well, and the corrective measures used for modifying the analysis when there is **heterogeneity** of variance will often correct both problems.

## 2.6. Analysis Strategies When Assumptions Are Violated

One common cause of **heterogeneity** of variances between levels of the treatment factor is a nonlinear relationship between the response and stimulus or treatment. For example, in the upper half of **Figure 2-5**, it can be seen that the response increases nonlinearly as a function of the factor levels.

The density functions, drawn on their sides at three treatment levels, represent how nonlinearity often affects the distribution of the response. As the mean or center of the distribution increases, the variance or spread in the distribution also increases, and the distributions have long tails on the right. One way of correcting this situation is to transform the response data prior to analysis.

The bottom half of **Figure 2-5** shows the potential result of a variance **stabilizing transformation**. On the transformed scale, the variance appears constant at different factor levels and the distributions appear more normal.



**Figure 2-5. Effect of Nonlinearities on Response Distribution**

### 2.6.1. Box-Cox Power Transformations

One way to recognize the need for a variance stabilizing transformation is to examine the plot of residuals versus cell means described in the last section. If the spread in the residuals tends to increase proportionally as a function of the cell means (as possibly indicated in the upper right of **Figure 2-4**) a transformation,  $Y = f(y)$  can usually be found that will result in a more sensitive analysis. Box and Cox (1964) proposed a series of power transformations  $Y = y^\lambda$  that normally work well. If the variance tends to increase as the mean increases, choose a value of  $\lambda$  less than one, and if the variance tends to decrease as the mean increases, choose a value of  $\lambda$  greater than one. **Table 2-3** summarizes some common **Box-Cox power transformations**. A common situation where the  $\sigma \propto \mu$  is when the response is actually a measure of variability, like the sample variance  $s^2$ .

**Table 2-3. Box-Cox Power Transformations**

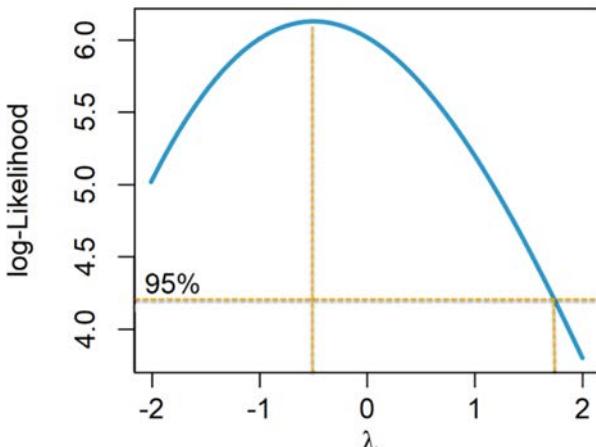
Relation between $\sigma$ and $\mu$	$\lambda$	Transformation
$\sigma \propto \mu^2$	-1	Reciprocal
$\sigma \propto \mu^{\frac{3}{2}}$	-1/2	Square Root of Reciprocal
$\sigma \propto \mu$	0	Log
$\sigma \propto \mu^{\frac{1}{2}}$	1/2	Square Root

In a CRD design with replicate experiments in each level of the treatment factor, one way to determine the most appropriate value of  $L$  to use in

the *Box-Cox transformation* is to plot the maximum of the log likelihood function (which is proportional to the reciprocal of the error sum of squares in the ANOVA) versus the value of  $\lambda$  used in transforming the data. The value of  $\lambda$  that maximizes the log likelihood (or minimizes the error sum of squares) would be most appropriate. This plot is called a **Box-Cox plot**. The `boxcox` function in the R package `MASS` makes this plot automatically. In the example shown below the `boxcox` function is used with the R `lm` object `mod1` that was fit to the data from the Aerogel Density experiment. The plot is shown in **Figure 2-6**, and  $\lambda = -0.0505$  maximizes the log likelihood.

```
library(MASS)
bc <- boxcox(mod1)
lambda <- bc$x[which.max(bc$y)]
lambda
```

[1] -0.5050505



**Figure 2-6. Box-Cox Plot for the Aerogel Density Experiment**

R code to add the transformation to the data frame `tAerogel` and fit the model to the transformed data follows.

```
tAerogel <- transform(aero_df, tDensity =
  density^(-.5050505))
mod2 <- aov( tDensity ~ time, data = tAerogel )
summary(mod2)
```

The resulting ANOVA table below shows the  $p$ -value for the factor time has decreased to 0.0209 from the 0.042 value shown in the earlier ANOVA of the untransformed data. Therefore the transformation has made the analysis slightly more sensitive.

```
Df   Sum Sq  Mean Sq  F value Pr(>F)
Time      2  0.01732  0.008662 6.134    0.0209 *
Residuals 9  0.01271  0.001412
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

For experiments where the variance heterogeneity is more pronounced, the **Box-Cox transformation** can greatly increase the sensitivity in detecting treatment effects.

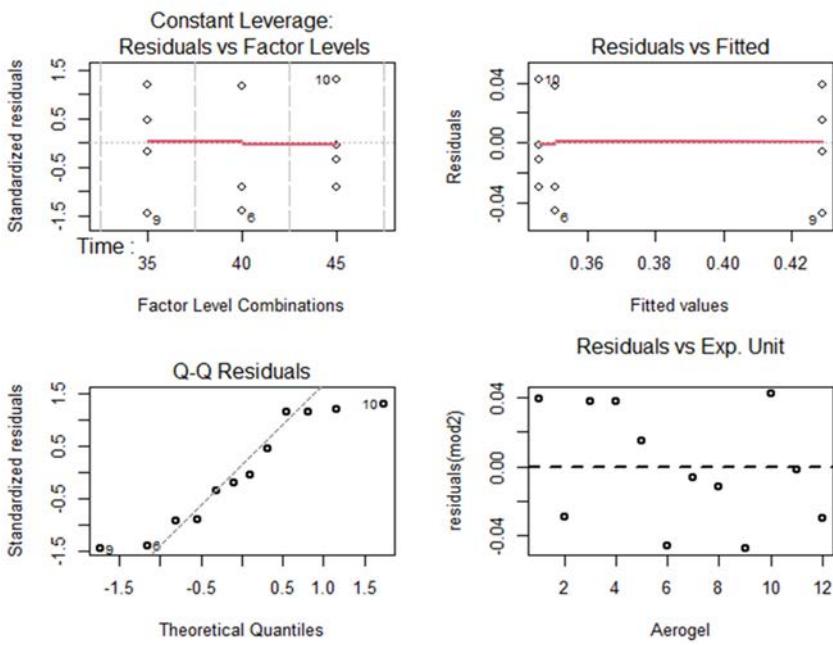
The graphs used to verify the assumptions of the analysis of the transformed data can be made by modifying the code of `mod1` replacing `mod1` with `mod2`. The result is shown in **Figure 2-7**. It can be seen in this Figure that the spread or variability of the residuals is nearly the same for each value of the predicted value or cell mean of responses raised to the  $-0.505$  power.

Here is the R code to add the transformation to the data frame `tAerogel` and fit the model to the transformed data follows.

```
par( mfrow = c(2,2) )
plot(mod2, which = 5)
plot(mod2, which = 1)
plot(mod2, which = 2)
plot(residuals(mod2) ~ ~ Aerogel,
main = "Residuals vs Exp. Unit",
font.main = 1, data = tAerogel)
abline(h = 0, lty = 2)
dev.off()
```

## 2.6.2. Distribution-Based Transformations

The distribution assumption for the effects model for the CRD described in **Section 2.3** was  $Y_{ij} \sim N(\mu + \tau_i, s^2)$ . However, if it is known that the data follow some distribution other than the normal distribution, such as the Binomial, Poisson, or Lognormal, then it would also be known that the **standard deviation** would not be constant.



**Figure 2-7. Plot of Residuals versus Cell Means after  $y^\lambda$  Transformation for Aerogel Density Experiment**

For example, if the response,  $Y$ , was a binomial count of the number of successes in  $n$  trials, then due to the **central limit theorem**,  $Y$  would be approximately normal, but  $\mu_Y = np$  and  $\sigma_Y = \sqrt{(np(1-p))}$ , where  $p$  is the probability of success. In situations like this where the distribution of the response is known to follow some specific form, then an appropriate transformation can be found to stabilize the variance. **Table 2-4** shows the transformation for common situations often encountered.

**Table 2-4. Response Distribution-Based Transformations**

Response Distribution	Variance in Terms of Mean $\mu$	Transformation $f(y)$
<b>Binomial</b>	$\frac{\mu(1-\mu)}{n}$	$\sin^{-1} \sqrt{\frac{y}{n}}$ (radians)
<b>Poisson</b>	$\mu$	$\sqrt{y}$ or $\sqrt{y + 1/2}$
<b>Lognormal</b>	$c\mu^2$	$\log(y)$

### 2.6.3. Alternatives to Least Squares Analysis

When the variance of the experimental error is not constant for all levels of the treatment factor, but it is not related to the cell means, a transformation will not be an appropriate way of equalizing or stabilizing the variances. A more general solution to the problem is to use **weighted least squares**. Using weighted least squares,  $\hat{\beta}$  is the solution to the normal equations  $\mathbf{X}'\mathbf{W}\mathbf{X}\boldsymbol{\beta} = \mathbf{X}'\mathbf{W}\mathbf{y}$ , where  $\mathbf{W}$  is a diagonal matrix whose diagonal elements are the reciprocals of the standard deviation within each treatment level. As an illustration of this method, consider the R code below for analyzing the data from the Aerogel Density experiment.

```
with(aero_df, { std <- tapply(Density, Time, sd)
  weights <- rep( 1/std, each = 4)
  mod3 <- lm( Density ~ Time, weights = weights,
  data = aero_df)
  anova(mod3)
})
```

In this example, the `with(aero_df, f...g)` function causes all statements within the `{ }` brackets to use the variables from the data frame `aero_df`. The `(Density, Time ,var)` function is used to calculate the variance of the response at each level of the factor time. The weights are calculated as the reciprocal of the standard deviations and the `rep()` function is used to expand the vector of weights to the number of rows in the data frame `aero_df`. The `lm` function calculates the weighted least squares estimates and the `anova` function prints the ANOVA table. The results appear on the next page.

#### Analysis of Variance Table

```
Response: Density
          Df  Sum Sq Mean Sq F value    Pr(>F)
Time       1 16.6803 16.6803  21.021 0.001003 **
Residuals 10  7.9351  0.7935
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

With these results, it can be seen that the  $F$ -test from the weighted least squares is more sensitive than the unweighted least squares, and the  $P$ -

value is similar to what was obtained with the *Box-Cox transformation* shown in **Section 2.6.1**.

When the **error distribution** is not normal, an alternative to analyzing a transformation of the response is to use a generalized linear model (see McCullagh and Nelder, 1989). In fitting a **generalized linear model**, the user must specify the error distribution and a link function in addition to the model. The method of maximum likelihood is used to estimate the model parameters and the generalized likelihood ratio tests are used to test the hypotheses. When the link function is the identity and the distribution is normal, the generalized linear model analysis will result in the method of least squares and the ANOVA *F*-test. There are several R functions to fit the generalized linear models and compute the appropriate likelihood ratio test statistics.

To illustrate the use of one of these functions to analyze experimental data, consider the following example from **Section 2.4**. I wanted to compare three different teaching methods to determine how the students would perceive the course. The treatment factor was the teaching method, the experimental unit was a class of students, and the response was the summary of student ratings for the course. The taught two sections of the course for three consecutive semesters resulting in a total of six experimental units or classes. Then I constructed a randomized list so that two classes were assigned to each teaching method. This would reduce the chance that other differences in the classes, or differences in his execution of the teaching methods, would bias the results. At the end of each semester, the students were asked to rate the course on a five-point scale, with 1 being the worst and 5 being the best. Therefore, the response from each class was not a single, normally distributed response,  $y$ , but a vector  $\langle y_1, \dots, y_5 \rangle$  response that followed the multinomial distribution. The summary data from the experiment is shown in **Table 2-5**.

This data is stored in the data frame `teach` in the package `daewr`. The following R code makes this data available and uses the function `polr` from the R package `MASS` (Venables & Ripley, 2002) to fit the full and reduced model.

**Table 2-5. Counts of Student Rating Scores**

Class	Method	1	2	3	4	5
<b>1</b>	1	2	14	12	8	6
<b>2</b>	3	1	11	15	15	10
<b>3</b>	2	3	8	18	14	10
<b>4</b>	3	1	9	17	15	12
<b>5</b>	2	4	12	19	9	7
<b>6</b>	1	3	16	13	10	4

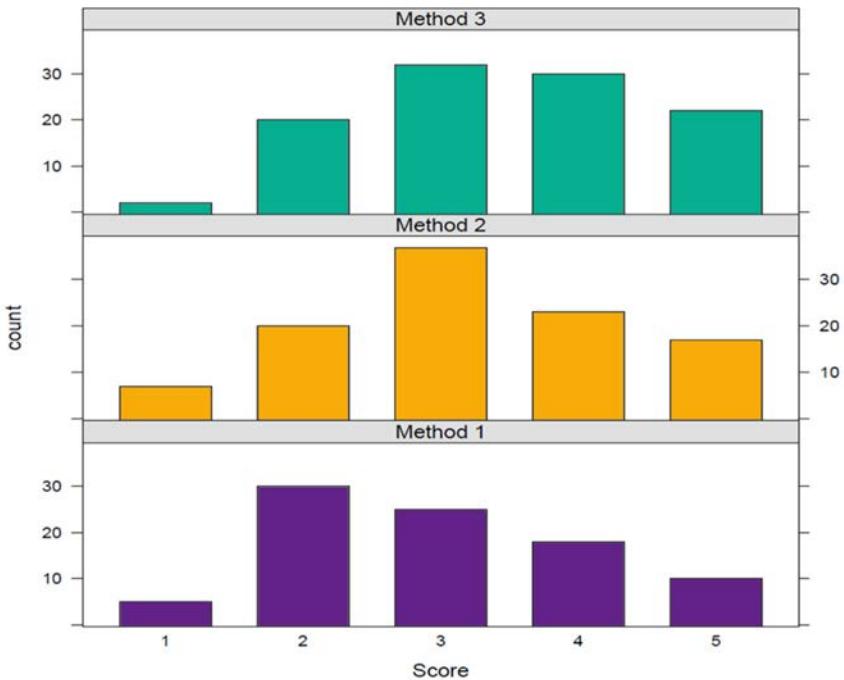
The function `polr` by default uses the logistic link function and the multinomial distribution. The response `score` and the treatment factor `method` in the data frame `teach` are factors, while the variable `score` is a numeric variable containing the counts of the various student rating scores. The formula in the full model, `modf`, includes the treatment factor, while the formula in the reduced model, `modr`, only includes the intercept.

```
library(daewr)
library(MASS)
modf <- polr(score ~ method, weight = count, data=teach)
modr <- polr(score ~ 1, weight = count, data = teach)
anova(modf,modr)
```

The `anova` function displays the likelihood ratio test of the significance of the treatment factor as shown below.

Likelihood ratio tests of ordinal regression models						
Response: score						
Model	Resid.	df	Resid. Dev	Test	Df	LR stat.
1	1	294	885.9465			Pr(Chi)
2	method	292	876.2986	1 vs 2	2	9.64787 0.008035108

The *p*-value for the likelihood ratio **chi-square statistic** is small indicating there is a significant difference between the teaching methods. Teaching method 1 had an average score of 2.98, teaching method 2 had an average score of 3.22, and teaching method 3 appeared to be the best with an average score of 3.47. This can also be visualized in the bar charts in **Figure 2-8**, which shows that the percentage of high scores given increases for teaching method 2 and 3.

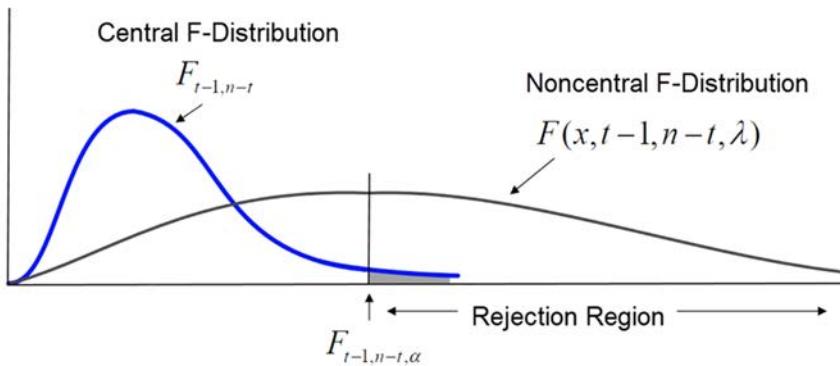


**Figure 2-8. Percentage of Student Rating Scores by Teaching Method**

## 2.7. Determining the Number of Replicates

The significance level,  $\alpha$ , of the ANOVA  $F$ -test of no treatment effect is the probability of rejecting the null hypothesis  $H_0 : \mu_1 = \mu_2 = \dots = \mu_t$ , when it is true. The power of the test is the probability of rejecting the null hypothesis when it is false. The test statistic  $msT/msE$  follows the  $F$ -distribution when the null hypothesis is true, but when the null hypothesis is false it follows the noncentral  $F$ -distribution. The noncentral  $F$ -distribution has a wider spread than the central  $F$ -distribution, as shown in **Figure 2-9**.

The spread in the noncentral  $F$ -distribution and probability exceeding the critical limit from the central  $F$ -distribution is an increasing function of the noncentrality parameter,  $\lambda$ .



**Figure 2-9. Central and Noncentral F-Distribution**

When the distribution is the noncentral  $F$ , the probability of exceeding the critical limit from the central  $F$ -distribution is called the **power**. The power is greater than the significance level,  $\alpha$ , when the null hypothesis is false making the noncentrality parameter greater than zero. The power can be computed for any scenario of differing means, if the values of the cell means, the variance of the experimental error, and the number of replicates per factor level are specified. For a constant difference among cell means, represented by  $\sum_{i=1}^t (\mu_i - \bar{\mu}_.)^2$ , the noncentrality parameter and the power increase as the number of replicates increase.

When the differences among cell means is large enough to have practical importance, the experimenter would like to have high power, or probability of rejecting the hypothesis of no treatment effects. When the difference among the means has practical importance to the researcher we call it **practical significance**. Practical significance does not always correspond to **statistical significance** as determined by the  $F$ -test from the ANOVA. Sometimes the number of replicates in the experiment is too few and the probability or power of detecting a difference of practical significance too low. Statistical significance can be made to coincide with practical significance by determining the appropriate number of replicates that result in the desired power. Doing this is the second technique that falls in the category of *error control* discussed in Chapter 1. The idea that increasing the number of replicates increases the sensitivity of the experiment is also due to Fisher (1966).

For example, if there is a difference among the cell means so that the corrected sum of squares ( $css = \sum_{i=1}^t (\mu_i - \bar{\mu}_.)^2$ ) is greater than zero, then the power or probability of rejecting  $H_0 : \mu_1 = \mu_2 = \dots = \mu_t$ , is given by

$$\pi(\lambda) = \int_{F_{t-1,t(r-1),\alpha}}^{\infty} F(x, t-1, t(r-1), \lambda) dx \quad \text{Eq. 2-10}$$

where  $F_{t-1,t(r-1),\alpha}$  is the  $\alpha$ th percentile of the  $F$ -distribution with  $(t-1)$  and  $t(r-1)$  degrees of freedom,  $F(x, t-1, t(r-1), \lambda)$  is the noncentral  $F$ -distribution with **noncentrality parameter**  $\lambda = r/\sigma^2 \sum_{i=1}^t (\mu_i - \bar{\mu}_.)^2$ . For a fixed value of  $\lambda = r/\sigma^2 \sum_{i=1}^t (\mu_i - \bar{\mu}_.)^2$ , the noncentrality parameter and the power increases as a function of the number of replicates,  $r$ . This probability can be calculated for various values of  $r$  until a value is found with adequate power. In this way the appropriate number of replicates can be determined. The `Fpower` function in the R package `daewr` facilitates these computations.

In the Aerogel Density experiment, suppose less than a  $3 - g/(cm^2)$  difference in treated aerogel densities is of no consequence. However, if changing the heating time from 35 minutes to 45 minutes causes a difference of more than  $3 - g/(cm^2)$  in aerogel density, the experimenter would like to know about it, because he will need to monitor heating time closely in the future to produce aerogel sheets of consistent density. In this case, we can regard  $\Delta = 3.0$  as a **practical difference** in cell means. The smallest  $css = \sum_{i=1}^t (\mu_i - \bar{\mu}_.)^2$  could be, with at least two cell means differing by  $\Delta$ , would be the case when one cell mean was  $\frac{\Delta}{2}$  higher than the grand mean, a second was  $\frac{\Delta}{2}$  less than the grand mean, and a third was equal to the grand mean. This would result in

$$css = \sum_{i=1}^t (\mu_i - \bar{\mu}_.)^2 = \left(\frac{\Delta}{2}\right)^2 + 0^2 + \left(-\frac{\Delta}{2}\right)^2 = \left(\frac{\Delta^2}{2}\right) = \left(\frac{3^2}{2}\right) = \frac{4}{5}$$

Assuming the variance of the experimental  $\hat{\sigma}^2 = 2.1$  was estimated from the sample variance in treated aerogel densities in a pilot experiment where several sheets were allowed to be heated for the

same length of time, then the noncentrality factor can be calculated as  $\lambda = \frac{r}{2,1} \times (4.5)$ . The power is calculated for  $r = 2, \dots, 6$  using the R code shown below. This code illustrates the use of the `Fpower1` function that takes as arguments, `alpha =  $\alpha$` , `nlev = t` (the number of levels of the factor), `nreps = r`, `Delta =  $\Delta$` , and `sigma =  $\sigma$` .

```
library(daewr)
rmin <- 2 #smallest number of replicates considered
rmax <- 6 # largest number of replicates considered
alpha <- rep(0.05, rmax - rmin + 1)
sigma <- sqrt(2.1)
nlev <- 3
nreps <- rmin:rmax
Delta <- 3
power <- Fpower1(alpha, nlev, nreps, Delta, sigma)
power
```

By using a vector argument for `nreps`, the function produces a corresponding vector of calculated power values that are shown in the following output.

	<code>alpha</code>	<code>nlev</code>	<code>nreps</code>	<code>Delta</code>	<code>sigma</code>	<code>power</code>
[1,]	0.05	3	2	3	1.449138	0.1947995
[2,]	0.05	3	3	3	1.449138	0.4041857
[3,]	0.05	3	4	3	1.449138	0.5903406
[4,]	0.05	3	5	3	1.449138	0.7328895
[5,]	0.05	3	6	3	1.449138	0.8329923

From this we can see that with  $r = 5$  replicates there would be a 73% chance of detecting a difference in cell means as large as 3.0, and with  $r = 6$  there is an 83% chance. With fewer than five replicates there is at least a 40% chance this difference will be missed. As a rule of thumb, the number of replicates that result in power between 0.80 and 0.90 is usually sufficient for most experimental studies.

## 2.8. Comparison of Treatments after the F-test

When the *F*-test for the null hypothesis  $H_0: \mu_1 = \mu_2 = \dots = \mu_t$  is rejected, it tells us that there are significant differences between at least two of the **cell means**, but if there are several levels of the treatment factor, it does not necessarily mean that all cell means are significantly

different from each other. When the null hypothesis is rejected, further investigation should be conducted to find out exactly which cell means differ. In some cases the investigator will have preplanned comparisons he would like to make; in other situations he may have no idea what differences to look for.

### 2.8.1. Preplanned Comparisons – Rad-Hard Protection

Considering the treatment factor levels in the *Radiation Space Component Protection (rad-hard)* experiment. As part of the "hardening" process, rad-hard electronics are shielded in a layer of depleted boron and mounted on insulating substrates, instead of on conventional semiconductor wafers. As the layering comes with a cost, the Space Force wants to evaluate several treatment options compared with the conventional *rad-hard* method. Some preplanned comparisons that might have been of interest are:

1.  $H_0: \mu_1 = \frac{1}{3}(\mu_2 + \mu_3 + \mu_4)$  [compared average yield vs traditional]
2.  $H_0: \mu_2 = \mu_3$  [spayed vs. painted]
3.  $H_0: \mu_3 = \mu_4$  [pained vs immersed]

The first comparison asks the question: Does an application of depleted boron significantly change the yield of protection required for geosynchronous orbital satellite electronic components? The second comparison asks the question: Is there a difference in protective yields between spaying the components with boron and painting them with boron? The third comparison asks the question: Does painting the components with boron change the protection yield gained by immersing components in boron?

These hypotheses can all be expressed in the general form  $H_0: \sum_{i=1}^t c_i = 0$ , where  $\sum_{i=1}^t c_i = 0$ . Since  $\sum_{i=1}^t c_i = 0$  are estimable functions, each of these hypotheses can be tested by computing the single estimable function  $L\hat{\beta}$  and its standard error  $s_{L\beta} = \sqrt{\sigma^2 L'(X'X)^{-1}L}$ . The ratio of the estimable function to its standard error follows the *t*-distribution. The `fit.contrast` function in the R package `gmodels` performs this test. For the boron layering experiment the code below loads the data and sets up and prints the contrast matrix  $L$ .

```

library(daewr)
boron <- read.csv("https://raw.githubusercontent.com
/stricje1/Data/master/boron.csv ")
mod4 <- aov( yield ~ treat, data = boron )
con <- matrix( c(1,-1/3,-1/3,-1/3,0,1,-1,0,0,0,1,-1),4,3)
L <- t(con)
rownames(L) <- c("rad-hard effect",
"spray (B) vs paint (C)", "paint (C) vs immerse (D)")
L

```

	[,1]	[,2]	[,3]	[,4]
rad-hard effect	1	-0.333	-0.333	-0.333
spray (B) vs paint (C)	0	1.000	-1.000	0.000
paint (C) vs immerse (D)	0	0.000	1.000	-1.000

The function call below prints the results. The options statement controls the number of digits after the decimal for printing in this book.

```

options(digits = 3)
library(gmodels)
fit.contrast(mod4, "treat", L)

```

The results are as follows.

	Estimate	Std. Error	t value	Pr(> t )
treatrad-hard effect	-8.8	0.825	-10.664	4.19e-08
treatspray (B) vs paint (C)	-3.8	0.975	-3.897	1.61e-03
treatpaint (C) vs immerse (D)	0.1	0.919	0.109	9.15e-01

The *p*-values in the column labeled *Pr > |t|*, in the above output, can be interpreted the same way the *p*-values for the *F*-statistic were interpreted, and we can see that: (1) boron coating enhances yield protection when compared to traditional protection, (2) spaying born results in higher yields of protection than painting boron, and (3) there is no significant difference in protection yield between painting and immersion.

When factor levels are quantitative, such as the heating time in the Aerogel Density experiment, preplanned comparisons often involve looking for the significance of linear or higher order polynomial trends in the response. **Contrast coefficients**,  $c_i$  for testing **orthogonal polynomial** trends, can be obtained from the R `contr.poly` function. The required input for this function is the number of levels of the factor.

The result is an **orthogonal matrix** with the contrast coefficients desired. For example, for the Aerogel Density experiment, the commands on the next page construct and print the contrast matrix.

```
contrasts(boron$treat) <- contr.poly(4)
contrasts(boron$treat)
```

The resulting contrast matrix below has coefficients for the linear and quadratic contrasts.

	.L	.Q	.C
A	-0.671	0.5	-0.224
B	-0.224	-0.5	0.671
C	0.224	-0.5	-0.671
D	0.671	0.5	0.224

The code using the R `aov` and `summary.lm` functions shown below calculates the contrasts and displays the results.

```
Mod5 <- aov( yield ~ treat, boron )
summary.lm(mod5)
```

In the following results, we can see that there is a significant (at the  $\alpha = 0.05$  level) linear trend, but no significant quadratic trend.

```
Call:
aov.default(formula = yield ~ treat, data = boron)

Residuals:
    Min      1Q  Median      3Q      Max 
 -1.80   -1.07   -0.20    1.18    2.60 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  45.300     0.345 131.39 < 2e-16 ***
treat.L      7.558     0.690   10.96  3e-08 ***
treat.Q     -3.200     0.690   -4.64  0.00038 ***
treat.C     -0.313     0.690   -0.45  0.65680  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.45 on 14 degrees of freedom
Multiple R-squared:  0.908, Adjusted R-squared:  0.888 
F-statistic: 45.9 on 3 and 14 DF,  p-value: 1.72e-07
```

If the levels for the factor time were created with the ordered command rather than the factor command, R automatically creates the  $X$  matrix using the **orthogonal polynomial contrasts** and the summary table above can be obtained without creating additional contrasts for time.

### 2.8.2. Unplanned Comparisons – Rad-Hard Protection

When a set of preplanned comparisons can be expressed as a saturated set of orthogonal contrasts, like the examples shown in the last section, these comparisons are independent and equivalent to partitioning the overall  $F$ -test of  $H_0: \mu_1 = \dots = \mu_t$ . However, if the comparisons are not planned in advance of running the experiment, the analyst might be tempted to choose the comparisons he or she would like to make based on the means of the data. This implicitly means that all possible comparisons have been made. When testing all possible comparisons, each at the  $\alpha = 0.05$  significance level, the overall significance level can be much higher than 0.05, greater than 50% in some cases. This means that even when there is no difference in the cell means  $\mu_1 = \dots = \mu_t$  there could be a high probability of finding one or more comparisons significant when each is tested individually. In order to reduce the overall (or experiment-wise) chance of a **type I error**, an adjustment must be made.

For pairwise comparisons of the form  $H_0: \mu_i = \mu_j$  for  $i \neq j$  **Tukey's HSD** (or honestly significant difference) method adjusts the critical region by using the studentized range statistic instead of the student's t-distribution. Using the HSD reject  $H_0: \mu_i = \mu_j$  in favor of the alternative  $H_a: \mu_i \neq \mu_j$  if  $|\hat{\mu}_i - \hat{\mu}_j| > (\sqrt{2})q_{I,n-t,\alpha}s_{\mu_i-\mu_j}$ . where  $q_{I,n-t,\alpha}$  is the upper percentile of the studentized range. This is only approximate when the sample sizes are unequal. If  $X_1, \dots, X_I$  are independent random variables following  $N(\mu, \sigma^2)$  and  $R = \max_i X_i - \min_i X_i$  then  $R/\hat{\sigma}$  follows the studentized range distribution (see Tukey, 1949a).

The R function [TukeyHSD](#) will make pairwise comparisons using Tukey's HSD method. The code below illustrates how this function is called to make the comparisons on the data from the rad-hard (boron) experiment.

```

mod4 <- aov( yield ~ treat, data = boron )
mod4.tukey <- TukeyHSD( mod4, ordered = T )
mod4.tukey

```

A portion of the output is shown below.

```

Tukey multiple comparisons of means
95% family-wise confidence level
factor levels have been ordered

Fit: aov.default(formula = yield ~ treat, data = boron)

$treat
   diff     lwr      upr p adj
B-A  6.3  3.312  9.29 0.000
D-A 10.0  7.166 12.83 0.000
C-A 10.1  7.266 12.93 0.000
D-B  3.7  0.866  6.53 0.009
C-B  3.8  0.966  6.63 0.008
C-D  0.1 -2.572  2.77 1.000

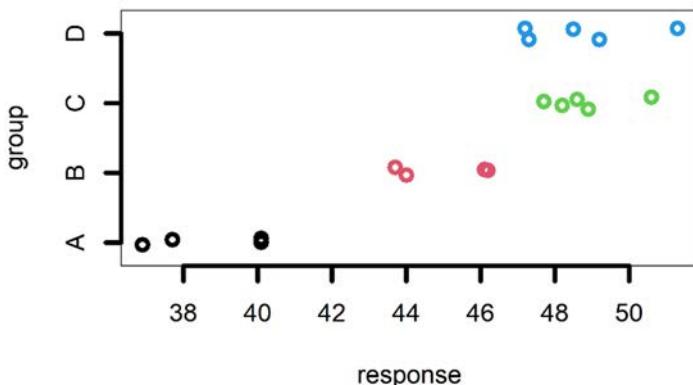
```

The first column of the output lists the comparison made, the next column lists the difference in cell means, and the next two columns are bounds for a 95% confidence interval on the difference of means of the form  $|\mu_i - \mu_j| \pm (\sqrt{2})q_{I,n-t,\alpha} s_{\mu_i - \mu_j}$ . The final column is a *p*-value for the test of the null hypothesis that the two means are equal. For example, the confidence interval for the last comparison,  $\mu_C - \mu_D$ , includes zero and the *p*-value is  $> 0.05$  indicating the boron protection yield for treatment (*C*—boron applied by painting) is not significantly different than the yield for treatment (*D*— boron applied by immersion). All other pairwise comparisons show a significant difference. We can also see this graphically in **Figure 2-10**.

```

stripchart(boron$yield ~ boron$treat, method="jitter",
          pch = 1, xlab = "response",
          ylab = "group", lwd = 3, col = c(1,2,3,4))

```



**Figure 2-10. Stripchart of the four treatments and their yields**

A less conservative method of comparing all possible cell means was developed independently by Newman (1939) and Keuls (Keuls, 1952). This method is also based on the studentized range statistic but is based on the range of the particular pair of means being compared, within the entire set of ordered means, rather than the range of the largest to smallest as *Tukey's HSD*. The means comparison using the student **Newman-Keuls method** can be made using the `Snk.test` function in the R package `agricolae` (de Mendiburu, 2012). The arguments for the `Snk.test` function are similar to the `TukeyHSD` function and are illustrated below using the data from the rad-hard (boron) experiment.

```
library(agricolae)
compare <- SNK.test( mod4, "treat", alpha = 0.05 )
print(compare)
```

A portion of the output is shown below.

\$statistics	\$parameters
MSerror Df Mean CV	test name.t ntr alpha
2.11 14 45.7 3.18	SNK treat 4 0.05
\$snk	
NULL	
\$means	
yield std r se Min Max Q25 Q50 Q75	
A 38.7 1.65 4 0.727 36.9 40.1 37.5 38.9 40.1	
B 45.0 1.33 4 0.727 43.7 46.2 43.9 45.0 46.1	

```

C 48.8 1.10 5 0.650 47.7 50.6 48.2 48.6 48.9
D 48.7 1.68 5 0.650 47.2 51.3 47.3 48.5 49.2

$comparison           $groups
NULL                  yield groups

C 48.8   a
attr("class")
[1] "group"
          D 48.7   a
          B 45.0   b
          A 38.7   c

```

The critical range section of the output lists the critical values for difference in means that range 2, 3, or 4 apart. In the last section of output, means with the same Group indicator on the left are not significantly different. This shows the boron protection yield for treatment (*C*—boron applied by painting) is not significantly different than the yield for treatment (*D*—boron applied by immersion). All other pairwise comparisons show a significant difference (in this case same results as *Tukey's HSD* method).

The last section of the output of the `Snk.test` function illustrates a compact way of presenting the significant differences in treatment means that are found by multiple comparison techniques like *Tukey's HSD* method or the student *Newman-Keuls* method. When reporting results in written text, this method of presentation can be modified by listing the means horizontally in the text from smallest to largest and using an underline in place of the Group indicator to show which means are not significantly different.

Referring to **Figure 2-11**, means that are not significantly different are underlined with the same line. The example below shows the means from an experiment to determine the effect of the download site upon the time to download a file.

B	D	A	C	E
2.84	3.42	3.79	4.33	5.64

**Figure 2-11. Means by treatment (site)**

```

time <- 
  c(2.95,2.71,2.80,3.10,2.75,2.71,2.77,2.99,2.76,2.88,
  3.57,3.48,3.60,3.39,3.22,3.10,3.50,3.27,3.49,3.61,
  3.78,3.61,3.98,3.77,3.87,3.74,3.68,3.60,3.77,4.05,
  4.28,4.09,4.50,4.41,4.07,4.46,4.18,4.37,4.57,4.39,
  5.62,5.66,5.65,5.63,5.64,5.61,5.67,5.60,5.62,5.69)
treat <- factor(rep(c("B", "D", "A", "C", "E"), rep(10,5)))
download <- data.frame(rsp = time, trt = treat)

```

The results show that the download time is not significantly different between sites *B* and *D*, and not significantly different between sites *D* and *A*, but there is a significant difference in the download time between sites *B* and *A*. Likewise, there is no significant difference in download times for sites *A* and *C*, but the download time for site *C* is significantly longer than either site *B* or *D*. Finally, site *E* has a significantly longer download time than any of the other sites. We can view this graphically using a strip chart as shown in **Figure 2-12**.

```

set.seed(1)
par(las = 1)
stripchart(download$time ~
  download$treat, method = "jitter",
  pch = 1, xlab = "response", ylab = "group")

```

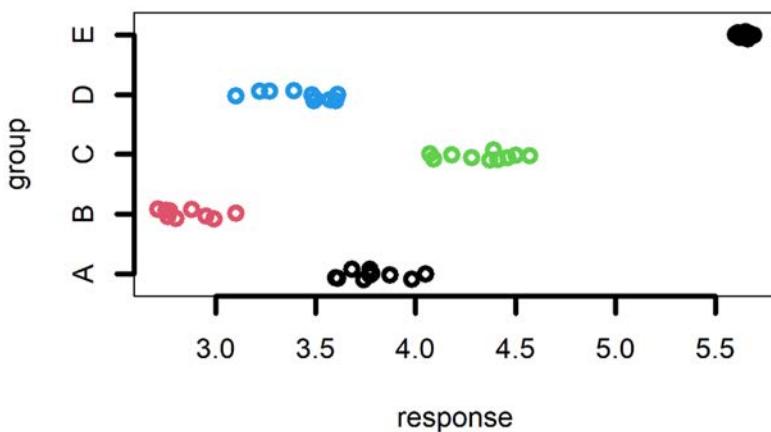


Figure 2-12. Strip chart of download times by cite (treatment)

### 2.8.3. Comparison of All Means to a Control – Rad-Hard Protection

In some experiments one of the treatment levels is the current or default level and the others are new or experimental levels. One of the main objectives in this type of experiment might be to compare the mean of each experimental level to the default, or sometimes called the **control level**. Dunnett (1955) developed a method to do this and control the *experiment-wise type I error rate*. In the boron protection yield experiment, treatment level ( $A$ —traditional treatment) can be thought of as the control. All other treatment levels can be compared to this one using the `glht` function in the R package `multcomp` (Hothorn, Bretz, & Westfall, 2008). To load the `multcomp` package you must also have the `mvtnorm` package (Genz, et al., 2012), and the `survival` package (Therneau, 2012) installed. When using the the `glht` function (by default) uses the first level of the treatment factor as the control.

```
summary(boron)
```

```
treat      yield
A:4    Min.   :36.9
B:4    1st Qu.:43.8
C:5    Median :47.2
D:5    Mean    :45.7
          3rd Qu.:48.6
          Max.   :51.3
```

As can be seen above the treatment level ( $A$ —traditional method) is the first level of the treatment factor in the data frame for the boron protection yield experiment. The code to use Dunnett's method to compare mean at each level of the treatment factor to the control ( $A$ ) by calling the `glht` function is shown below.

```
library(multcomp)
boron.dun <- glht(mod4, linfct = mcp(treat = "Dunnett"),
                  alternative = "greater")
summary(boron.dun)
```

The output below is the result of one-tailed tests. When comparing all treatment levels to a control, the desired direction of the difference is often known. Therefore a one-tailed test, rather than a two-tailed test, may be required. Other options for `alternative` = in the code above are "`less`" or "`two.sided`".

Simultaneous Tests for General  
Linear Hypotheses

Multiple Comparisons of Means: Dunnett Contrasts

```
Fit: aov.default(formula = yield ~ treat, data = boron)
```

Linear Hypotheses:

	Estimate	Std. Error	t value	Pr(>t)
B - A <= 0	6.300	1.028	6.13	1.8e-05 ***
C - A <= 0	10.100	0.975	10.36	< 1e-05 ***
D - A <= 0	10.000	0.975	10.25	< 1e-05 ***
---				
Signif. codes:	0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1			

(Adjusted p values reported -- single-step method)

When there is no control level of the treatment factor, there may still be interest in comparing all treatment levels to the best level. For example, in the experiment to see the effect of the download site on the time to download a file, described at the end of **Section 2.8.2**, it may be of interest to find all sites whose average download times are not significantly longer than the site with the minimum observed average download time. In the means shown in **Section 2.8.2**, site *B* had the shortest observed download time. To compare all treatment means to the best level and control the experiment-wise error rate, the MCB procedure of Hsu (1984) can be used. This procedure turns out to be equivalent to Dunnett's method. To use this method, first look at the observed means and decide which is the best. Next, set up contrasts comparing each level to the best.

Finally, call the `glht` function to perform *Dunnett's test*. For example, if the data for the file download experiment from **Section 2.8.2** were contained in a data frame called `download`, and the second level (or "B") of the factor `site` had the minimum average download time, the code below sets up the contrasts and calls the `glht` function to compare treatment means for sites "A", "C", "D", and "E" to the mean for site "B" using Dunnett's method.

```
aov.ex <- aov(time ~ site, data=download)
K <- rbind( c( 1,-1,0,0,0), c(0,-1,1,0,0),
c(0, -1, 0, 1, 0), c(0, -1, 0, 0, 1) )
```

```

rownames(K) <- c( "A-B", "C-B", "D-B", "E-B" )
colnames(K) <- names(coef (aov.ex))
dht <- glht( aov.ex, linfct =
  mcp( site = "Dunnett" ), alternative = "two.sided")
summary(dht)

```

### Simultaneous Tests for General Linear Hypotheses

#### Multiple Comparisons of Means: Dunnett Contrasts

Fit: aov(formula = time ~ treat, data = download)

#### Linear Hypotheses:

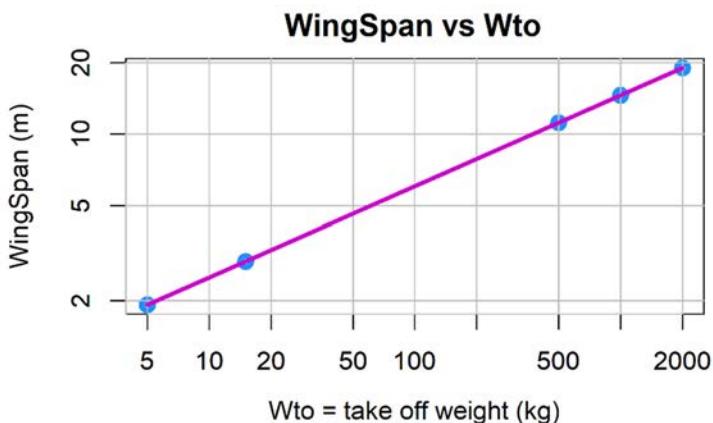
	Estimate	Std. Error	t value	Pr(> t )							
B - A == 0	-0.9430	0.0631	-14.94	<1e-05	***						
C - A == 0	0.5470	0.0631	8.67	<1e-05	***						
D - A == 0	-0.3620	0.0631	-5.73	<1e-05	***						
E - A == 0	1.8540	0.0631	29.37	<1e-05	***						
---											
Signif. codes:	0	'***'	0.001	'**'	0.01	'*'	0.05	.	0.1	' '	1
(Adjusted p values reported -- single-step method)											

## 2.9. Example: Unmanned Aerial Vehicle Design

There are Unmanned Aerial Vehicles (UAV) design guidelines based on trends identified from the best data we can acquire for existing UAVs. These guidelines are for UAVs fitted with either four stroke or Wankel engines. We have omitted two stroke engines because we are mostly interested in long range UAVs that can be used in surveillance work.

The design process for a UAV to be used in a surveillance application starts with the takeoff weight. **Figure 2-13** shows a relationship that allows us to estimate the takeoff weight,  $W_{to}$ , for various levels of the factor, WingSpan. Thus, we can build a response model given a single factor. The data used in the trend analysis has been based on: Aerosonde 1, Shadow 200, Hermes 180, Hermes 450, and Predator MQ-1.

In practice, we seldom test systems that have only one factor. In the next chapter, we will expand this example to include additional factors used in UAV design.



*Figure 2-13. UAV wing span vs takeoff weight (Wto) ( $\log_{10}$  scale)*

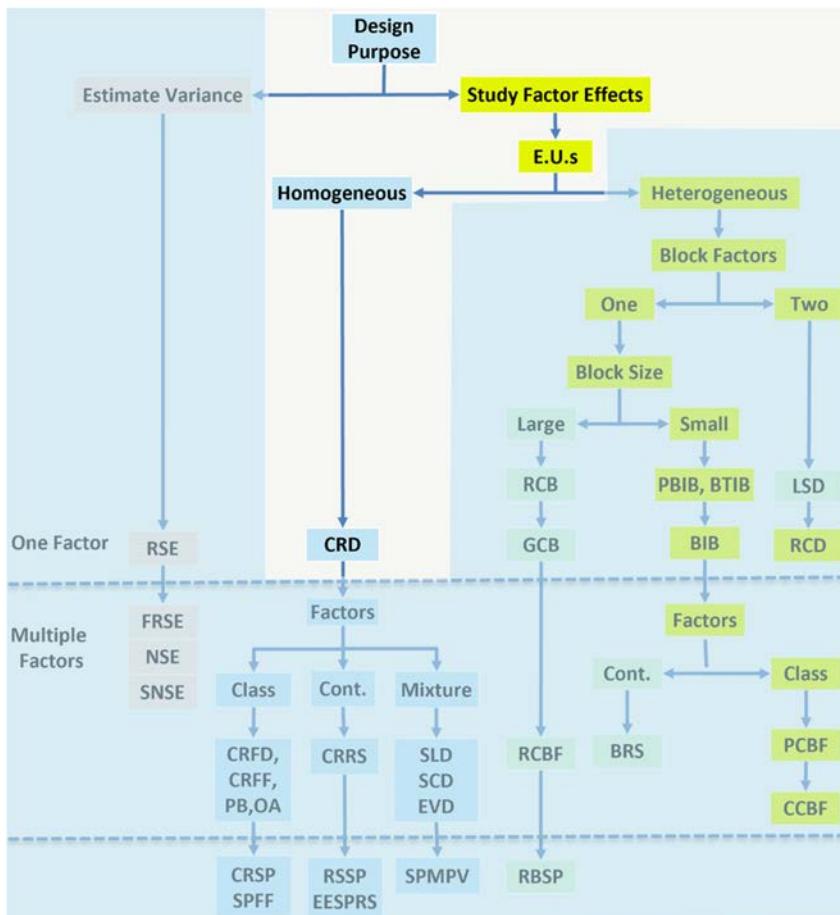
## 2.10. Review of Important Concepts

In order to determine if cause and effect relationships exist and to make predictions about the results of future actions, experiments must be performed wherein certain factors are purposely varied while others are held constant. The one-factor design is the simplest case where one-factor is varied while all other known factors are held constant.

**Figure 2-14** shows a roadmap for selecting an appropriate experimental design. When there is only one factor under study and experimental units are homogeneous, the CRD design should be used as indicated in black in the figure. This is the only situation presented in Chapter 2. As additional designs are presented in subsequent chapters we will explain the other branches in **Figure 2-14**.

Fisher's technique of randomizing experimental units to treatment levels guarantees the long run validity of the CRD and minimizes the chance that changes in unknown factors, or lurking variables, will bias the results. The way a series of experiments is conducted dictates what model should be used for analysis. The model for the analysis of the CRD or completely randomized for one-factor design is  $y_{ij} = \mu_i + \varepsilon_{ij}$  or  $y_{ij} = \mu + \tau_i + \varepsilon_{ij}$ , where  $y_{ij}$  is the observed response for the  $j^{th}$  replicate of the  $i^{th}$  treatment level,  $\mu_i$  is the cell mean for the  $i$ th level, and  $\tau_i$  is the effect.  $\varepsilon_{ij}$  is the experimental error for the  $j$ th observation on treatment level  $i$ .  $\varepsilon_{ij}$ s are assumed to be independent and normally

distributed with constant variance  $\sigma^2$ . The typical analysis is to fit the linear model by the method of least squares - maximum likelihood and perform a likelihood ratio  $F$ -test of the  $H_0: \mu_1 = \dots = \mu_t$ . If the data were not collected in a proper randomized design with replicates, analyzing data in this way may be totally misleading.



**Figure 2-14. Design Selection Roadmap**

The credibility of the conclusions of analysis depends on the degree to which the assumptions are valid. The independence assumption is the most critical and it is guaranteed when replicated experimental units are randomized to treatment factor levels. The other assumptions should be checked. The constant variance and normality assumptions can be checked by plotting the residuals versus cell means and by making a

normal probability plot of the residuals. If these assumptions are violated, the data should be analyzed on a transformed scale or by weighted least squares or the method of maximum likelihood for the generalized linear model.

If a significant difference in cell means is found with the overall F-test, further investigation of the differences can be made. If comparisons are planned in advance of running the experiment and can be described as a set of orthogonal comparisons, the overall *F*-test can be partitioned to test these comparisons. The experiment type I error rate for all possible comparisons of means can be controlled by using Tukey's HSD or the less conservative student Newman-Keuls method. For comparing all means to a control level, Dunnett's method should be used and for comparing all means to the best (largest or smallest), Hsu's Multiple Comparisons to the Best (MCB) method should be used.



## 3. Factorial Designs

---

### 3.1. Introduction

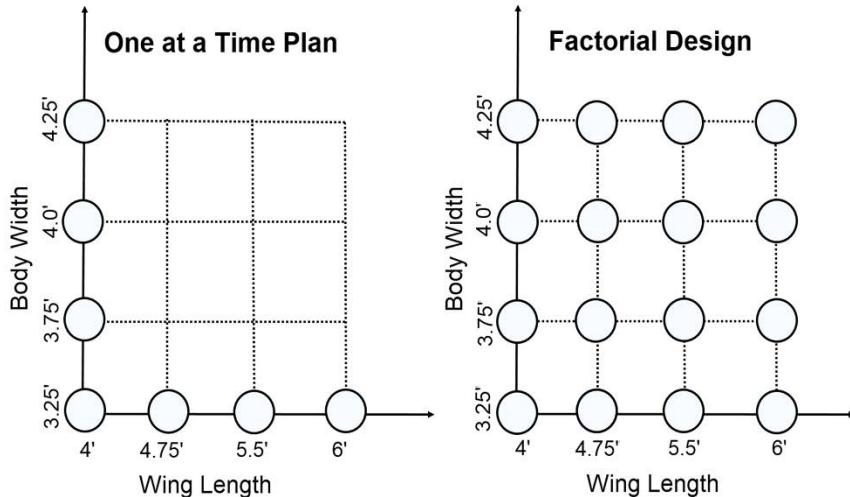
In Chapter 2 we examined one-factor designs. These are useful only when one factor is under study. When multiple factors are under study, one classical approach is to study each separately while holding all others constant. Fisher (1966) pointed out that this approach is useful for demonstrating known relationships to students in laboratory courses when the influence of other factors is known. However, this approach is both inefficient and potentially misleading when it comes to discovering new knowledge through experimentation. A much better strategy for experimenting with multiple factors is to use a factorial design. In a factorial design the cells consist of all possible combinations of the levels of the factors under study. Factorial designs accentuate the factor effects, allow for estimation of interdependency of effects (or interactions), and are the first technique in the category of what is called treatment design.

By examining all possible combinations of factor levels, the number of replicates of a specific level of one factor is increased by the product of the number of levels of all other factors in the design, and thus the same power or precision can be obtained with fewer replicates. In addition, if the effect of one factor changes depending on the level of another factor, it will be seen in a factorial plan. This phenomenon will be missed in the classical approach where each factor is only varied at constant levels of the other factors. The example in the next section will illustrate these facts.

### 3.2. Classical One at a Time versus Factorial Plans

In **Section 2.9**, a set of experiments with Unmanned Aerial Vehicles (UAVs) was described. In those experiments only one factor, the wing span, was under study. However, to maximize the flight time of UAVs, it is best to consider more than one factor. For example, consider varying wing length over four levels and the body width over four levels, such as 4.25 ft, 4.0 ft, 3.75 ft, and 3.5 ft. The left side of **Figure 3-1** represents the classical plan in which one factor is varied at a time. The circles in the

diagram represent experiments or runs. Using this approach, the experiments across the bottom of the figure would be completed by varying wing length while holding body width constant at 3.5 ft. Next, the three additional experiments up the left side of the figure would be completed by varying body width while holding the wing length constant at its low level of 4.0 ft. If eight replicate runs were to be made for each of these experiments, a total of 56 experiments would be required.



**Figure 3-1. Comparison of One-at-a-Time and Factorial Designs**

If the objective were to find the combination with the longest flight time, the classical approach would be to complete the experiments with one factor first. Next one would calculate the cell means and then select the level with the highest mean. Finally, the second factor would be varied while holding the first constant at its optimal level, not the lowest level as shown in **Figure 3-1**. However, Fisher's caution to randomize would tell you this is a bad strategy. If any unknown forces changed after the first set of experiments, the results could be biased. Additionally, the optimal level of one factor may depend upon the level of the other factor. Therefore, by varying one factor at a time, the overall optimum may be missed.

The diagram on the right side of **Figure 3-1** represents a factorial plan for the UAV experiments. Here it can be seen that experiments are run at all combinations of the levels of the two factors. In this plan, if two

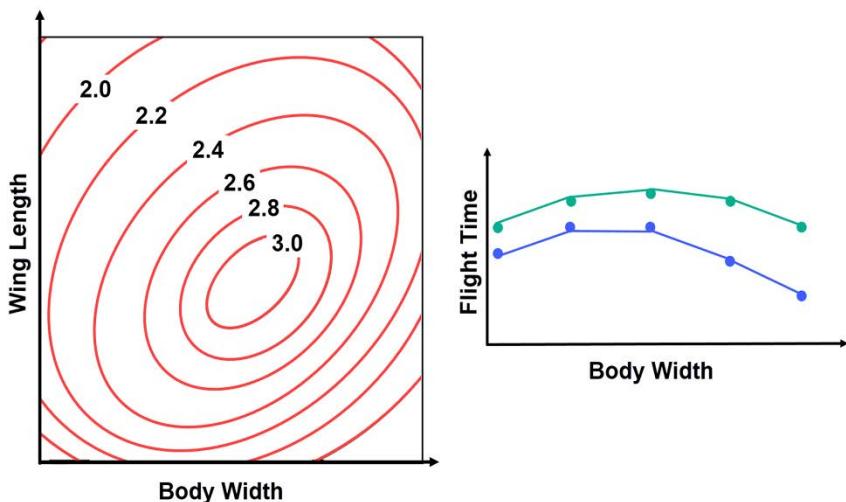
replicates of each cell are completed, there will be eight replicates of each level of wing length, and eight replicates of each level of body width which is equivalent to the one-at-a-time plan. Therefore the factorial plan would have the same precision or power for detecting factor effects as the one-at-a-time plan but is more efficient since it requires only  $2 \times 16 = 32$  total runs as opposed to the 56 required by the one-at-a-time plan. The number of replicates of each factor level in the factorial design is equal to the number of replicates per cell times the product of the levels of all other factors in the design. This multiplication is referred to as hidden replication. In the case shown in **Figure 3-1**, there are only two factors each at four levels; therefore, the number of replicates of each factor level is  $2 \times 4 = 8$ . In the factorial plan, the 32 treatment combinations would be randomized to experimental units, thus preventing biases from unknown sources.

In more complicated research problems many treatment factors may be under study. The efficiency of factorial designs can be demonstrated even in the simplest case where each factor has only two levels. For example, consider a design with four factors. A factorial design would require all combinations of four factors at two levels, or  $2^4 = 16$  cells. If two replicates were run for each cell, there would be a total of  $2 \times 16 = 32$  experiments or runs. To examine the effect of any one of the four factors, half the runs (or  $2 \times 2^3 = 16$  due to the hidden replication) would be at one level of the factor and half at the other level. Thus the treatment effect would consist of a difference of two averages of 16. Results from the same 32 experiments can be used to calculate the treatment effect for each of the four factors. To have equal precision for comparing treatment effects using a one-at-a-time plan, 32 runs would be required for comparing the levels of each factor while holding the others constant. This would result in  $4 \times 16 + 16 = 80$  experiments, or 2.5 times the number required for a factorial design!

### **3.3. Interpreting Interactions**

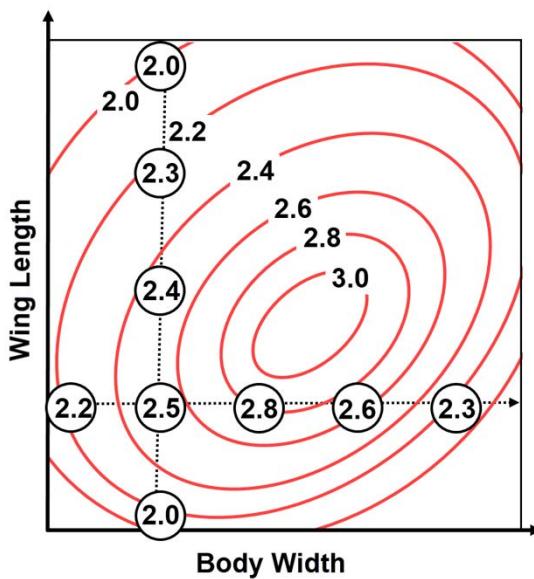
If there is an interaction or joint effect between two factors, the effect of one factor upon the response will differ depending on the level of the other factor. This can be illustrated graphically in **Figure 3-2**. On the left side of the figure is a contour plot representing the results of a series of

experiments with UAVs. This plot can be interpreted like a topological map with the lines representing contours of equal flight time. You can simulate what would happen in a series of experiments where wing length was held constant and body width varied by drawing a straight line parallel to the body width axis across the contour plot. The flight time for various runs can be read off as the label for the contour lines the straight line intersects. For example, if wing length were held constant at a value below its mid-point on the left of the contour plot, the flight times resulting from five runs with varying body width are represented as the black line traced on the graph at the right in **Figure 3-2**. If the wing length were held constant at a higher value, the grey line indicates what the result of a series of experiments with body width might look like. The fact that the two lines or curves on the right side of the figure are not parallel indicates there is an interaction between wing length and body width. They show that the effect of body width depends upon the wing length.



*Figure 3-2. Contour Plot of Flight Time for UAV Experiment*

Interactions are common in the real world but using the classical one-at-a-time strategy of experimentation tacitly assumes that interactions do not exist. To see the fallacy that results from this assumption, examine **Figure 3-3**, which represents what would happen if one were to search for the optimum combination of wing length and body width.



*Figure 3-3. One-at-a-Time Optimization with Paper UAV*

The vertical set of circles are drawn at the wing length, body width combinations for a series of experiments that vary wing length while holding body width constant. The numbers within the circles represent the resulting flight times. After examining the result of this series of experiments, the optimal wing length would be chosen, and another series of experiments would be conducted by holding the wing length constant at its optimal value and varying the body width. The results of these experiments can be visualized as the horizontal series of circles. The maximum result, *Eq. 2-8*, is not the overall optimum, because the optimal wing length depends on the body width and vice versa.

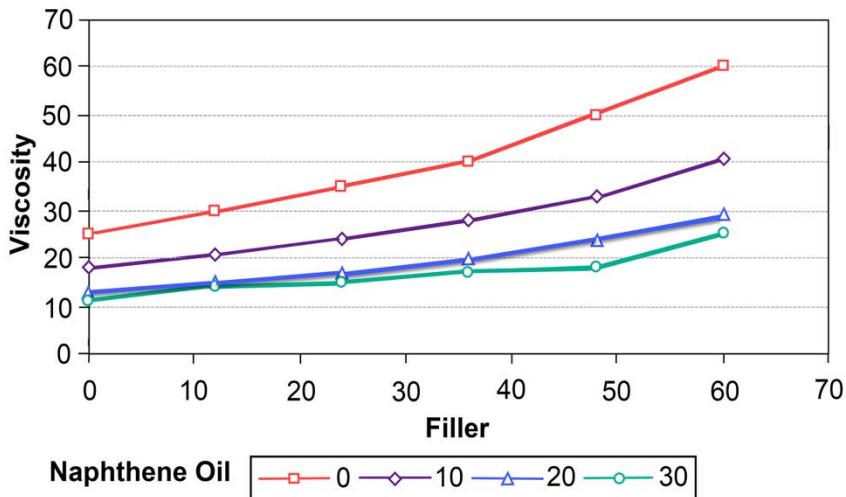
*Table 3-1. Mooney Viscosity of Silica B at 100XC*

Naphthalene Oil (phr)	Filler (phr)					
	0	12	24	36	48	60
0	25	30	35	40	50	60
10	18	21	24	28	33	41
20	13	15	17	20	24	29
30	11	14	15	17	18	25

When the effect of the factors is close to linear, the interaction is easier to explain in words. **Table 3-1** shows the results of a factorial experiment

conducted by Derringer (1974) to determine the effect of elastomer compounds on the viscosity silica at 100XC. The elastomer compounds were Naphthene Oil, studied at 4 levels, and Filler content, studied at 6 levels.

**Figure 3-4** shows a graphical representation of the data in the table. This figure is called an interaction plot. As the Filler is increased from 0 to 60, the viscosity increases along a fairly linear trend. However, the slope of the trend line depends upon the level of Naphthene Oil. When there is no Naphthene Oil added, increasing the Filler from 0 to 60 causes viscosity to increase rapidly from 25 to 60; but when there is 30 phr of Naphthene Oil, increasing the Filler from 0 to 60 causes a more gradual increase in viscosity from 11 to 25.



**Figure 3-4. Interaction Plot of Filler and Naphthene Oil**

Since interactions are common in factorial experiments, it is important to learn how to explain or interpret an interaction in order to clearly present the results of research studies. This is best done by describing the effect of one factor upon the response, and then contrasting or comparing how that effect changes depending on the level of the other factor. An interaction plot, like **Figure 3-4**, can guide the description or interpretation. Many more examples of interpreting interactions will be given throughout this chapter and the remainder of the book.

### 3.4. Creating a Two-Factor Factorial Plan in R

A factorial design can be easily created using R in several ways. For example, nested loops could be used, the base R function `expand.grid`, or several functions available in user developed packages. For example, the `expand.grid` function (which creates a data frame containing all possible combinations of supplied factors) is illustrated below to create a factorial design to study paper UAVs.

```
D <- expand.grid( BW = as.factor(c(3.25, 3.75, 4.25)),  
                  WL = as.factor(c(4, 5, 6)) )
```

D

BW	WL
1 3.25	4
2 3.75	4
3 4.25	4
4 3.25	5
5 3.75	5
6 4.25	5
7 3.25	6
8 3.75	6
9 4.25	6

As can be seen, this code creates an unreplicated  $3^2$  factorial in factors Body width = `BW` and Wing length = `WL` with the supplied levels for these factors. This design is stored in the data frame `D`. To replicate every run in the design, the R function `rbind` (which stacks one copy of the  $3^2$  factorial design on top of the other) is used as shown below.

```
D <- rbind(D, D)
```

To randomize the order of the runs, the `sample` function can be used to create a random order of the run numbers 1 to 18. Next, the rows in the data frame `D` are ordered by this random order list. Finally, the factor columns from the data frame `D` can be copied into a new data frame `Copterdes` and this data frame can be written to a `.csv` file to produce an electronic data collection form like the historical example on [Section 2.2.1](#). This is illustrated in the code on the next page.

```
set.seed(2591)
```

```

D <- D[order(sample(1:18)), ]
CopterDes <- D[ c( "BW", "WL" ) ]
CopterDes
write.csv(CopterDes, file = "CopterDes.csv",
row.names = FALSE)

```

The list that was produced by this code shows that the first experiment would consist of constructing a UAV with a body width of 3.75' and a wing length of 4', dropping it from a fixed height, and timing its flight. The second experiment consists of constructing a UAV with a body width of 3.75' and a wing length of 5', dropping it from the same fixed height, and timing its flight, and so forth. The randomization will help prevent biases from any lurking variables such as changes in air currents, changes in temperature, or learning curves in dropping or timing UAVs. By removing the `set.seed(2591)` statement in the above code, a different random ordering of the experiments will result each time the code is run.

### 3.5. Analysis of a Two-Factor Factorial in R

The mathematical model for a completely randomized two-factor factorial design can be written as:

$$y_{ijk} = \mu_{ij} + \varepsilon_{ijk} \quad \text{Eq. 3-1}$$

where  $i$  represents the level of the first factor,  $j$  represents the level of the second factor, and  $k$  represents the replicate number. This model is called the cell means model and  $\mu_{ij}$  represents the expected response in the  $ij$ th cell.

Another way of representing the model is the effects model

$$y_{ijk} = \mu + \alpha_i + \beta_j + \alpha\beta_{ij} + \varepsilon_{ijk}. \quad \text{Eq. 3-2}$$

In this model,  $\alpha_i$ ,  $\beta_j$  are the main effects and represent the difference between the marginal average of all experiments at the  $i^{th}$  level of the first factor and the overall average, and the difference between the marginal average at the  $j^{th}$  level of the second factor and the overall average, respectively. The interaction effects,  $\alpha\beta_{ij}$ , represent the difference between the cell mean,  $\mu_{ij}$ , and  $\mu + \alpha_i + \beta_j$ . With these

definitions,  $\sum_i \alpha_i = 0$ ,  $\sum_j \beta_j = 0$ ,  $\sum_i \alpha\beta_{ij} = 0$ , and  $\sum_j \alpha\beta_{ij} = 0$ .

$\mu_{ij} = \mu + \alpha_i + \beta_j + \alpha\beta_{ij}$ , are estimable functions but the individual effects,  $\alpha_i$ ,  $\beta_j$ , and  $\alpha\beta_{ij}$  are not estimable functions. Contrasts among the effects such as  $\sum_i c_i \alpha_i$  and  $\sum_j c_j \beta_j$ , where  $\sum_i c_i = 0$ ,  $\sum_j c_j = 0$  are estimable only in the additive model where all  $\alpha\beta_{ij}$ 's are zero. Contrasts of the form  $\sum_i \sum_j b_{ij} \alpha\beta_{ij}$ , where  $\sum_i b_{ij} = 0$ ,  $\sum_j b_{ij} = 0$  are estimable even in the non-additive model. The estimable functions and their standard errors can be computed with the `estimable` function in the R package `gmodels`. The marginal means  $\mu + \alpha_i + \alpha\beta_{i\cdot}$  and  $\mu + \beta_j + \alpha\beta_{\cdot j}$  are estimable functions and they and the cell means can be computed using the R function `model.tables`, which will be illustrated in **Section 5.4.1**.

### 3.5.1. Matrix Representation of Model and Analysis

The effects model can be represented in matrix notation as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} = (\mathbf{1} | \mathbf{X}_A | \mathbf{X}_B | \mathbf{X}_{AB}) \begin{bmatrix} \mu \\ \boldsymbol{\beta}_A \\ \boldsymbol{\beta}_B \\ \boldsymbol{\beta}_{AB} \end{bmatrix} + \boldsymbol{\varepsilon} \quad \text{Eq. 3-3}$$

For example, consider a case where the first factor has two levels, the second factor has three levels, and there are two replicates per cell. Then

$$\begin{bmatrix} y_{111} \\ y_{112} \\ y_{211} \\ y_{212} \\ y_{121} \\ y_{122} \\ y_{221} \\ y_{222} \\ y_{131} \\ y_{132} \\ y_{231} \\ y_{232} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mu \\ \alpha_1 \\ \alpha_2 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \alpha\beta_{11} \\ \alpha\beta_{21} \\ \alpha\beta_{12} \\ \alpha\beta_{22} \\ \alpha\beta_{13} \\ \alpha\beta_{23} \end{bmatrix} + \begin{bmatrix} \varepsilon_{111} \\ \varepsilon_{112} \\ \varepsilon_{211} \\ \varepsilon_{212} \\ \varepsilon_{121} \\ \varepsilon_{122} \\ \varepsilon_{221} \\ \varepsilon_{222} \\ \varepsilon_{131} \\ \varepsilon_{132} \\ \varepsilon_{231} \\ \varepsilon_{232} \end{bmatrix}$$

The  $\mathbf{X}'\mathbf{X}$  is singular and to solve the normal equations (using the default treatment coding) the R function `lm` drops the indicators for the first

level of each factor in the main effect columns and creates the columns for the interaction as all possible products of the main effect columns. This makes the  $\mathbf{X}'\mathbf{X}$  matrix full rank as was the case for the one-factor model in **Sections 2.4.1** and **2.4.2**. For the two-factor model where the first factor has two levels, the second factor has three levels, and there are two replicates per cell, the  $\mathbf{y}$  vector and recoded  $\mathbf{X}$  matrix would be as shown on the next page.

$$\mathbf{y} = \begin{bmatrix} y_{111} \\ y_{112} \\ y_{211} \\ y_{212} \\ y_{121} \\ y_{122} \\ y_{221} \\ y_{222} \\ y_{131} \\ y_{132} \\ y_{231} \\ y_{232} \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

This treatment coding makes the  $\mu_{11}$  cell mean the standard, and the resulting effect estimates are shown below.

$$(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} = \hat{\boldsymbol{\beta}} = \begin{bmatrix} \hat{\mu} + \hat{\alpha}_1 + \hat{\beta}_1 + \hat{\alpha}\hat{\beta}_{11} \\ \hat{\alpha}_2 - \hat{\alpha}_1 + \hat{\alpha}\hat{\beta}_{21} - \hat{\alpha}\hat{\beta}_{11} \\ \hat{\beta}_2 - \hat{\beta}_1 + \hat{\alpha}\hat{\beta}_{12} - \hat{\alpha}\hat{\beta}_{11} \\ \hat{\beta}_3 - \hat{\beta}_1 + \hat{\alpha}\hat{\beta}_{13} - \hat{\alpha}\hat{\beta}_{11} \\ \hat{\alpha}\hat{\beta}_{11} + \hat{\alpha}\hat{\beta}_{22} - \hat{\alpha}\hat{\beta}_{12} - \hat{\alpha}\hat{\beta}_{21} \\ \hat{\alpha}\hat{\beta}_{11} + \hat{\alpha}\hat{\beta}_{23} - \hat{\alpha}\hat{\beta}_{13} - \hat{\alpha}\hat{\beta}_{21} \end{bmatrix}$$

The error sum of squares  $ssE = \mathbf{y}'\mathbf{y} - \hat{\boldsymbol{\beta}}'\mathbf{X}'\mathbf{y} = \mathbf{y}'(\mathbf{I} - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}')\mathbf{y}$ , where  $\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$  are the estimates produced by the `lm` function in R. To test the hypothesis  $H_0: \alpha_1 = \alpha_2 = 0$ ,  $H_0: \beta_1 = \beta_2 = \beta_3 = 0$ , and  $H_0: \alpha\beta_{11} = \alpha\beta_{12} = \alpha\beta_{22} = \alpha\beta_{13} = \alpha\beta_{23} = 0$ , the likelihood ratio F-tests are obtained by calculating ratios of the ANOVA mean squares. What the `lm` designates as the sums of squares for factor A is  $ssA = \hat{\boldsymbol{\beta}}'\mathbf{X}'\mathbf{y} - \frac{(\mathbf{1}'\mathbf{y})^2}{\mathbf{1}'\mathbf{1}}$ , where the model is simplified to include only the effects for the first factor, that is  $\mathbf{X} = (\mathbf{1}|\mathbf{X}_A)$ . The error sums of squares

for this simplified model is  $ssE_A$ . The sums of squares for factor  $A$  is denoted  $R(\alpha|\mu)$ . The sums of squares for factor  $B$  is denoted  $R(\beta|\alpha, \mu) = ssE_A - ssE_B$  where  $ssE_B$  is the error sums of squares from the reduced model where  $\mathbf{X} = (\mathbf{1}|\mathbf{X}_A|\mathbf{X}_B)$ . Finally, the sums of squares for the interaction  $AB$  is denoted  $R(\alpha\beta|\beta, \alpha, \mu) = ssE_B - ssE$ . In general when there are  $a$  levels of factor  $A$ ,  $b$  levels of factor  $B$ , and  $r$  replicates per cell, the ANOVA table for the two-factor factorial design can be presented symbolically as shown in **Table 3-2**.

**Table 3-2. Analysis of Variance Table**

Source	Df	Sum of Squares	Mean Squares	F-ratio
$A$	$a - 1$	$R(\alpha \mu)$	$\frac{ssA}{a - 1}$	$F = \frac{msA}{msE}$
$B$	$b - 1$	$R(\beta \alpha, \mu)$	$\frac{ssB}{b - 1}$	$F = \frac{msB}{msE}$
$AB$	$(a - 1)(b - 1)$	$R(\alpha\beta \beta, \alpha, \mu)$	$\frac{ssAB}{(a - 1)(b - 1)}$	$F = \frac{msAB}{msE}$
Error	$ab(r - 1)$	$ssE$	$\frac{ssE}{ab(r - 1)}$	

The sums of squares  $ssA$ ,  $ssB$ , and  $ssAB$  can also be written in the form

$$ssA = (\mathbf{L}_\alpha \hat{\beta})' (\mathbf{L}_\alpha (\mathbf{X}' \mathbf{X})^{-1} \mathbf{L}')^{(-1)} (\mathbf{L}_\alpha \hat{\beta})$$

$$ssB = (\mathbf{L}_\beta \hat{\beta})' (\mathbf{L}_\beta (\mathbf{X}' \mathbf{X})^{-1} \mathbf{L}')^{-1} (\mathbf{L}_\beta \hat{\beta})$$

$$ssAB = (\mathbf{L}_{\alpha\beta} \hat{\beta})' (\mathbf{L}_{\alpha\beta} (\mathbf{X}' \mathbf{X})^{-1} \mathbf{L}'_{\alpha\beta})^{-1} (\mathbf{L}_{\alpha\beta} \hat{\beta}),$$

where  $\mathbf{L}_\alpha$ ,  $\mathbf{L}_\beta$ , and  $\mathbf{L}_{\alpha\beta}$  are contrast matrices computed internally by the `lm` function, see Goodnight (1980). Under the null hypotheses the  $F$ -ratios  $\frac{msA}{msE}$ ,  $\frac{msB}{msE}$ , and  $\frac{msAB}{msE}$  follow the  $F$ -distribution with the degrees of freedom shown in the table, and under the alternative they follow the noncentral  $F$ -distribution. The noncentrality parameter for  $F = \frac{msA}{msE}$  is given by the expression  $\lambda_\alpha = (\sigma^2)^{-1} (\mathbf{L}_\alpha \hat{\beta})' (\mathbf{L}_\alpha (\mathbf{X}' \mathbf{X})^{-1} \mathbf{L}'_\alpha)^{-1} (\mathbf{L}_\alpha \hat{\beta})$ . The noncentrality parameters for the  $F$ -ratios  $\frac{msB}{msE}$  and  $\frac{msAB}{msE}$  are similarly given. When there is an equal number,  $r$ , of replicates in each cell, the noncentrality parameters can be shown to be equal to

$$\lambda_{\alpha} = br \sum_i \frac{\alpha_i^2}{\sigma^2} \quad \text{Eq. 3-4}$$

$$\lambda_{\beta} = ar \sum_j \frac{\beta_j^2}{\sigma^2} \quad \text{Eq. 3-5}$$

and

$$\lambda_{\alpha\beta} = r \sum_j \frac{\alpha\beta_{ij}^2}{\sigma^2} \quad \text{Eq. 3-6}$$

To illustrate the analysis of a two-factor factorial experiment in using the R function `aov` consider the data in **Table 3-3**. These are the results of a two-factor experiment given by Hunter (1983). In this data, an experiment consisted of burning an amount of fuel and determining the CO emissions released. The experimental unit is the portion of a standard fuel required for one run, and the response,  $y$ , is the carbon monoxide (CO) emissions concentration in grams/meter<sup>3</sup> determined from that run. There were two replicate runs for each combination of factor levels separated by commas in **Table 3-3**. Factor  $A$  is the amount of ethanol added to an experimental unit or portion of the standard fuel, and factor  $B$  is the fuel-to-air ratio used during the burn of that fuel.

**Table 3-3. Data from Ethanol Fuel Experiment**

A=ethanol additions	B=air/fuel ratio	y=CO emissions
0.1	14	66, 62
0.1	15	72, 67
0.1	16	68, 66
0.2	14	78, 81
0.2	15	80, 81
0.2	16	66, 69
0.3	14	90, 94
0.3	15	75, 78
0.3	16	60, 58

The data for this experiment is stored in the data frame `COdata` in the `daewr` package where the levels of ethanol and ratio are stored as the factors `Eth` and `Ratio`. The R commands to analyze the data are shown below.

```
library(daewr)
mod1 <- aov( CO ~ Eth * Ratio, data = C0data )
summary(mod1)
```

The ANOVA table that results follows. There it can be seen that `aov` produces a table of the sums of squares, as described earlier. It can be seen from the tables that the two effects and their interaction are significant as indicated by the *P*-values to the right of the *F*-values.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)						
Eth	2	324	162	31.4	8.8e-05 ***						
Ratio	2	652	326	63.1	5.1e-06 ***						
Eth:Ratio	4	678	170	32.8	2.2e-05 ***						
Residuals	9	46	5								
---											
Signif. codes:	0	'***'	0.001	'**'	0.01	'*'	0.05	'.'	0.1	' '	1

The `model.tables` function produces the results shown on the next page. The top line is the estimate of the overall mean  $\hat{\mu}$ . The next two sections show the marginal means for each factor along with the standard deviation of the values averaged in each mean. If the interaction was not significant, the marginal means would reveal the direction of the factor effects, but further preplanned comparisons or other multiple comparison procedures could be used to draw definite conclusions. The next section shows the cell means, and the final section shows the standard errors of the differences in marginal and cell means.

```
model.tables( mod1, type = "means", se = T )
```

```
Tables of means
Grand mean
72.833
Eth
 0.1  0.2  0.3
66.83 75.83 75.83
Ratio
 14   15   16
78.5 75.5 64.5
Eth:Ratio
  Ratio
Eth  14   15   16
  0.1 64.0 69.5 67.0
  0.2 79.5 80.5 67.5
```

```

0.3 92.0 76.5 59.0
Standard errors for differences of means
      Eth Ratio Eth:Ratio
      1.312 1.312     2.273
replic.    6       6       2

```

Two estimate specific contrasts of the main effects, the estimable function from the R package `gmodels` can be utilized. To use it we must first construct contrasts to replace the default treatment contrasts used by the R function `aov`. For example, in the first statement below we construct the contrast coefficients for comparing the first factor level to the third in a three-level factor. A second contrast orthogonal to the first is also constructed, and the contrast matrix `cm` is created by using the two contrasts as columns.

```

c1 <- c(-1/2, 0, 1/2)
c2 <- c(.5, -1, .5)
cm <- cbind( c1, c2 )

```

In the call of the `aov` function below, the `cm` contrast matrix will be used for both main effects rather than the default treatment contrasts used by the `aov` function. The next lines load the `gmodels` package and create labels for the contrasts which compare the first factor level to the third factor level. The vector following each label is an indicator vector for which model coefficient is displayed. It selects the first coefficient for ethanol and ratio. Finally, the estimable function is called with the inputs being the `mod2`, that was created by the `aov` function, and the contrast labels and definitions in `c`.

```

mod2 <- aov( CO ~ Eth * Ratio, contrasts = list(
  Eth = cm, Ratio = cm ), data = C0data)
library(gmodels)
c <- rbind('Ethanol 0.3 vs 0.1' = c(0,1,0,0,0,0,0,0,0),
           'Ratio 16 vs 14' = c(0,0,0,1,0,0,0,0,0) )
estimable(mod2,c)

```

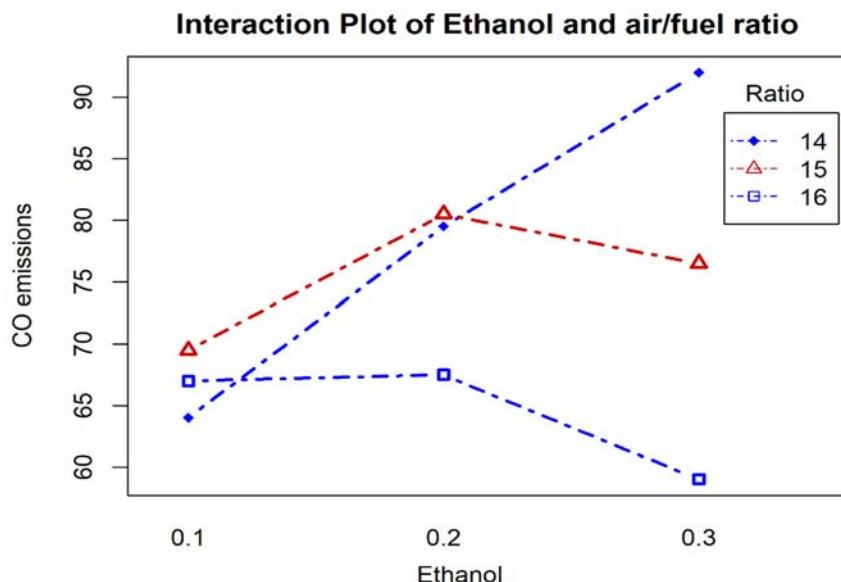
The results are shown below. These are both estimable functions, and the estimates along with their respective standard errors and *t*-ratios for testing the hypotheses,  $H_0: \sum_i c_i \alpha_i = 0$  and  $H_0: \sum_j c_j \beta_j$  are given.

Estimate	Std. Error	t value	DF	Pr(> t )
----------	------------	---------	----	----------

```
Ethanol 0.3 vs 0.1    9    1.3123     6.858    9 7.4066e-05
Ratio 16 vs 14        -14   1.3123   -10.668   9 2.0837e-06
```

These estimates would be estimable and meaningful if there were no significant interaction between ethanol addition level and air/fuel ratio, but in this case there is a significant interaction and the difference in CO emissions caused by changing the amount of ethanol addition will depend on the air/fuel ratio and the difference in CO emission caused by changing the air/fuel ratio will depend on the amount of ethanol added. An interaction graph is a better way of interpreting these results. An interaction plot can be generated using the R function `interaction.plot` as shown below. This code uses the `with` statement to call the `interaction.plot` function using variables names in the data frame `COdata` to produce **Figure 3-5**.

In this plot we can see more clearly the dependence of effects. Increasing the amount of ethanol added to the fuel from 0.1 to 0.3 causes CO emissions to increase linearly from 64 grams/liter to 92 grams/liter when the air/fuel ratio is at its low level of 14. This is shown by the dotted line with black diamonds representing the cell means.



**Figure 3-5. Interaction Plot Ethanol and Air/Fuel Ratio**

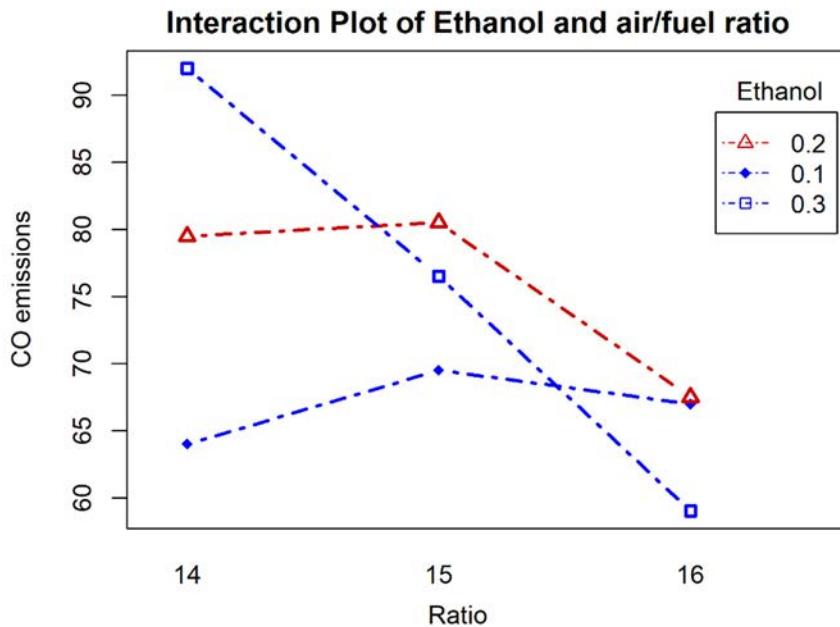
However, when the air/fuel ratio is at its high level of 16 (illustrated by the solid line with squares representing the cell means), increasing the ethanol added to the fuel from 0.1 to 0.3 actually causes a decrease in CO emissions from 67 grams/liter to 59 grams/liter along a nearly linear trend. Finally, when the air/fuel ratio is held constant at its mid-level of 15 (illustrated by the dashed line with triangles representing the cell means), increasing ethanol from 0.1 to 0.2 causes CO emissions to increase by 11 grams/liter; but a further increase in ethanol to 0.3 causes a decrease in CO emissions of 4 grams/liter to 76.5.

The interpretation above again illustrates the principle of comparing the effect of one factor across the levels of the other factor in order to describe an interaction. This was done by comparing the effect of changing the ethanol addition between the levels of air/fuel ratio. It could also be done in the opposite way. For example, the R code below reverses the interaction plot as shown in **Figure 3-6**.

```
Ethanol <- COdata$Eth
with(COdata, (interaction.plot(Ratio, Ethanol, CO,
  type = "b", pch = c(18,24,22), leg.bty = "o",
  main="Interaction Plot of Ethanol and air/fuel
  ratio", xlab = "Ratio", ylab = "CO emissions",
  col = c("blue","red3"), lty = 4, lwd = 2.5)))
```

In this plot the solid line, with squares representing the cell means, shows the effect of increasing air/fuel ratio when ethanol is added at the high rate of 0.3. Carbon monoxide emissions decrease linearly from 92 grams/liter to 59 grams/liter. However, when ethanol is added at the low rate of 0.1, the CO emissions actually increase slightly from 64 grams/liter to 67 grams/liter as a result of increasing air/fuel ratio from 14 to 16. This can be seen on the dotted line with black diamonds representing the cell means. When ethanol is added at the mid-rate of 0.2, there is little change in CO emissions when air/fuel ratio is increased from 14 to 15, but there is a decrease in CO emissions of 13 grams/liter caused by increasing air/fuel ratio from 15 to 16. The latter result can be visualized on the dashed line with triangles representing the cell means.

Either way of presenting and interpreting the interaction is valid as long as we discuss how the effect of one factor changes depending upon the level of the other.



*Figure 3-6. Interaction Plot Ethanol and Air/Fuel Ratio*

The factor effects, that we should compare depend on which one is of most interest in a particular research problem. Another thing to notice about the two interpretations is that we assume cause and effect relationships. We say the change in the response is caused by the change in the factor or the change in the response is the result of changing the factor. We could not make this statement when discussing the results of an observational study.

### 3.5.2. Determining the Number of Replicates

We can follow one of two possible methods to determine the number of replicates for a factorial experiment that will result in a power between 0.80 to 0.90 (for detecting differences that have practical significance). The first method is to consider detecting differences among the cell means. The second method is to consider detecting a practical size difference in the marginal means for the factors in the experiment.

When looking for differences among the cell means, we consider the cells in the factorial to be an unstructured group as in a one-factor design. Using the cell means model  $y_{ijk} = \mu_{ij} + \varepsilon_{ijk}$ , the procedure is the same as it was described for the one-factor model  $y_{ij} = \mu_i + \varepsilon_{ij}$  in **Section 2.6**. The noncentrality parameter for the  $F$ -test is:

$$\lambda_\alpha = (\sigma^2) \sum_{i=1}^a \sum_{j=1}^b (\bar{\mu}_{ij} - \bar{\mu}_{..})^2$$

When looking for differences in the marginal means for the factors, the noncentrality factor for the first main effect is:

$$\lambda_a = br \sum_i \frac{\alpha_i^2}{\sigma^2} = br \sum_i \frac{(\bar{\mu}_{.j} - \bar{\mu}_{..})^2}{\sigma^2}$$

and for the second main effect the noncentrality factor is:

$$\lambda_b = ar \sum_j \frac{\beta_j^2}{\sigma^2} = ar \sum_j \frac{(\bar{\mu}_{i.} - \bar{\mu}_{..})^2}{\sigma^2}$$

If  $\Delta$  is considered to be the size of a practical difference in cell means, then the smallest  $\lambda = r/\sigma^2 \sum_{i=1}^a b \sum_{j=1}^b (\bar{\mu}_{ij} - \bar{\mu}_{..})^2$  could be with two cells differing by at least  $\Delta$  is  $(r\Delta^2)/(2\sigma^2)$ . Likewise, if  $\Delta$  is considered to be the size of a practical difference in marginal means for factor  $A$ , the smallest  $\lambda_a = br \sum_i (\bar{\mu}_{i.} - \bar{\mu}_{..})^2/\sigma^2$  could be with two marginal means differing by at least  $\Delta$  is  $(br\Delta^2)/(2\sigma^2)$ . Here again we can see the efficiency of factorial designs because the noncentrality factor for detecting differences in marginal factor  $A$  means is larger than the noncentrality factor for detecting differences of cell means by a factor of  $b$ , the number of levels of factor  $B$ .

Consider the following example: A UAV experiment is planned to investigate the effects of four levels of factor  $A$  = wing length, and four levels of factor  $B$  = body width, upon the flight time. If pilot experiments with nine replicates of one design resulted in flight times of 2.8, 2.6, 3.5, 3.0, 3.1, 3.5, 3.2, 3.4, and 3.4 seconds. How many replicates would be required to detect a difference in flight times of 1 second with a power of 0.90?

From the pilot tests the standard deviation of experimental error can be estimated as  $s = 0.32$ . If  $\Delta = 1.0$  is considered to be a practical size difference in cell means, the R code in **Section 2.7** can be modified to give the answer. In a 4-by-4 factorial there are 16 cells so the number of levels of the factor is considered to be 16. The modified R code is shown below.

```
library(daewr)
rmin <- 2 # smallest number of replicates
rmax <- 8 # largest number of replicates
sigma <- .32
alpha <- .05
Delta <- 1
nlev <- 16
nreps <- c(rmin:rmax)
power <- Fpower1(alpha, nlev, nreps, Delta, sigma)
options(digits = 5)
power
```

The results of running this code show that 6 replicates per cell would be required to obtain a power of at least 0.90.

	alpha	nlev	nreps	Delta	sigma	power
[1,]	0.05	16	2	1	0.32	0.24173
[2,]	0.05	16	3	1	0.32	0.48174
[3,]	0.05	16	4	1	0.32	0.69246
[4,]	0.05	16	5	1	0.32	0.83829
[5,]	0.05	16	6	1	0.32	0.92326
[6,]	0.05	16	7	1	0.32	0.96664
[7,]	0.05	16	8	1	0.32	0.98655

If we consider  $\Delta = 1.0$  to be a practical size difference in marginal means for one of the factors, the results will be different. The degrees of freedom for the numerator would be  $v_1 = 4 - 1$ , the degrees of freedom for the denominator would be  $v_2 = 16(r - 1)$ , the noncentrality factor for a main effect  $A$  would be  $\lambda_a = (br\Delta^2)/(2\sigma^2)$ , and the noncentrality factor for a main effect  $B$  would be  $\lambda_b = (ar\Delta^2)/(2\sigma^2)$ . The R code below demonstrates the use of the **Fpower2** function in the **daewr** package for determining the number of replicates required to detect a difference of  $\Delta$  in marginal means of the factors in a two-factor factorial. The arguments to **Fpower2** that we must supply

are: `alpha`, `nlev` (a vector of length 2 containing the number of levels of the first factor ( $A$ ) and the second factor ( $B$ )), `nreps= r`, `Delta= Δ`, and `sigma= σ`.

```
library(daewr)
rmin <- 2 # smallest number of replicates
rmax <- 4 # largest number of replicates
alpha <- .05
sigma <- .32
Delta <- 1.0
nlev <- c(4,4)
nreps <- c(rmin:rmax)
result <- Fpower2(alpha, nlev, nreps, Delta, sigma)
options(digits = 5)
result
```

The results of running this code appear below. Here it can be seen that with only  $r = 2$  replicates per cell the power for detecting a  $\Delta = 1.0$  difference in marginal means for factor  $A$  or  $B$  is greater than the power for detecting differences of  $\Delta = 1.0$  in cell means with  $r = 8$  replicates per cell. Again this demonstrates the efficiency of factorial experiments through hidden replication.

	<code>alpha</code>	<code>a</code>	<code>b</code>	<code>nreps</code>	<code>Delta</code>	<code>sigma</code>	<code>powera</code>	<code>powerb</code>
[1,]	0.05	4	4	2	1	0.32	0.99838	0.99838
[2,]	0.05	4	4	3	1	0.32	1.00000	1.00000
[3,]	0.05	4	4	4	1	0.32	1.00000	1.00000

With the ability to calculate power quickly, it is possible to explore many potential designs before actually running the experiments. The number of factors, the number of levels of each factor, and the number of replicates in each cell all affect the power to detect differences. Power calculations help us determine an efficient use of our resources.

### 3.5.3. Analysis with an Unequal Number of Replicates per Cell

Although it would be unusual to plan a factorial experiment with an unequal number of replicates per cell, the data from a factorial experiment may end up with an unequal number of replicates due to experiments that we could not complete, or responses that we could not measure, or simply lost data. As long as the chance of losing an

observation was not related to the treatment factor levels, we can still analyze the data from a factorial experiment with an unequal number of replicates per cell and we can interpret in a manner similar to the way it would for the equal replicate case. However, the computational formulas for analyzing the data differ for the case with an unequal number of replicates.

To illustrate why the analysis shown in **Section 3.5.1** is inappropriate, consider again the data from the ethanol fuel experiment described in **Section 3.5.1**. This time assume one observation in the cell where air/fuel ratio = 16 and ethanol level = 0.3 was missing. Then **Table 3-4** shows the data with each response value written above its symbolic expected value.

The R code below the table creates a data frame containing the data in **Table 3-4**, by inserting a missing value into the 9th row and third column.

**Table 3-4. Fuel Experiment with Unequal Reps**

Eth	air/fuel		
	14	15	16
0.1	66	72	68
	62	67	66
	$\mu + \alpha_1 + \beta_1 + \alpha\beta_{11}$	$\mu + \alpha_1 + \beta_2 + \alpha\beta_{12}$	$\mu + \alpha_1 + \beta_3 + \alpha\beta_{13}$
0.2	78	80	66
	81	81	69
	$\mu + \alpha_2 + \beta_1 + \alpha\beta_{21}$	$\mu + \alpha_2 + \beta_2 + \alpha\beta_{22}$	$\mu + \alpha_2 + \beta_3 + \alpha\beta_{23}$
0.3	90	75	60
	94	78	
	$\mu + \alpha_3 + \beta_1 + \alpha\beta_{31}$	$\mu + \alpha_3 + \beta_2 + \alpha\beta_{32}$	$\mu + \alpha_3 + \beta_3 + \alpha\beta_{33}$

```
C0datam <- C0data
C0datam[18, 3] <- NA
```

The marginal column means for the levels of air/fuel ratio factor computed using the `model.tables` statement as shown on page 66 and the modified data frame `C0datam` would be 78.5, 75.5, and 65.8, respectively. The expected value of the marginal means for the first two columns would be:  $\mu + \beta_1$ ,  $\mu + \beta_2$ , since  $(\alpha_1 + \alpha_2 + \alpha_3)/3 = 0$  and

$(\alpha\beta_i + \alpha\beta_i + \alpha\beta_i)/3 = 0$  for  $i = 1, 2$ . However, the expected value of the last marginal column mean would be  $\mu + \beta_3 + (2\alpha_1 + 2\alpha_2 + \alpha_3)/5 + (2\alpha\beta_{13} + 2\alpha\beta_{23} + \alpha\beta_{33})/5$  and is not an unbiased estimate of  $\mu + \beta_3$ . The comparison between the first and third column means would not be an unbiased estimate of  $\beta_1 - \beta_2$ . Likewise, the last marginal row mean would not be an unbiased estimate of  $\mu + \alpha_3$ .

If we produce the ANOVA table of the data in `Codatam` with the R function `aov`, the  $F$ -tests will not test the same hypotheses that they do in the case of equal number of replicates per cell. When there is an unequal number of replicates in the cells, the noncentrality parameter for the  $F$ -test of  $H_0: \alpha_1 = \dots = \alpha_a$ , that is based on  $R(\alpha|\mu)$  will not be  $\lambda_a = rb \sum_i \alpha_i^2$  but a quadratic form involving the elements of  $\alpha, \beta$  as well as  $\alpha\beta$ . The noncentrality for the  $F$ -test of  $H_0: \beta_1 = \dots = \beta_b$  based on  $R(\beta|\mu, \alpha)$  will be a quadratic form involving the elements of  $\beta$  and  $\alpha\beta$ .

To calculate adjusted sums of squares for the null hypothesis for the main effects, use the `contr.sum` option in the R `lm` function and the `Anova` function from the R `car` package (Fox & Weisberg, 2011). The option type II in the ANOVA function computes the type II sums of squares, and the option type III produces the type III sums of squares. The type II sum of squares for the factors  $A$  and  $B$  can be represented as  $ssA_{II} = R(\alpha|\mu, \beta)$ , and  $ssB_{II} = R(\beta|\mu, \alpha)$ .  $R(\alpha|\mu, \beta)$  is the difference in the error sums of squares for the reduced model where  $X = (\mathbf{1}|X_B)$  and the full model where  $X = (\mathbf{1}|X_A|X_B|X_{AB})$ . The corresponding noncentrality factor for the corresponding  $F$ -test will be a quadratic form that only involves  $\alpha' = (\alpha_1, \alpha_2, \alpha_3)$  and  $\alpha\beta'$ . When there is an equal number of replications per cell, the sums of squares computed by the `aov` function are identical to the type II sums of squares.

The type III sum of squares for the factors  $A$  and  $B$  can be represented as  $ssA_{III} = R(\alpha|\mu, \beta, \alpha\beta)$ , and  $ssB_{III} = R(\beta|\mu, \alpha, \alpha\beta)$ .  $R(\alpha|\mu, \beta, \alpha\beta)$  is the difference in the error sums of squares for the reduced model where  $X = (\mathbf{1}|X_B|X_{AB})$  and the full model where  $X = (\mathbf{1}|X_A|X_B|X_{AB})$ . The corresponding noncentrality factor for the corresponding  $F$ -test will be a quadratic form that only involves  $\alpha' = (\alpha_1, \alpha_2, \alpha_3)$ . When there is an equal number of replications per cell, the sums of squares computed by the `aov` on are identical to the type III sums of squares.

Some analysts prefer to use the type II sums of squares and others prefer the type III sums of squares when there is an unequal number of replicates per cell. In this book we illustrate the type III sums of squares and hypothesis tests, although the type II sums of squares can be obtained by changing the option from `type = "III"` to `type = "II"` in the call to the `Anova` function.

The code to produce the type III ANOVA table ethanol fuel experiment after removing the observation with the value of 58 (from the cell with the air/fuel ratio = 16 and the ethanol level = 0.3) is shown below.

```
library(car)
mod2 <- lm( CO ~ Eth * Ratio, data = C0datam,
            contrasts = list( Eth = contr.sum,
                               Ratio = contr.sum ))
Anova( mod2, type = "III" )
```

The results follow.

	Sum Sq	Df	F value	Pr(>F)
(Intercept)	86198	1	15496.4	1.9e-14 ***
Eth	319	2	28.7	0.00022 ***
Ratio	511	2	46.0	4.1e-05 ***
Eth:Ratio	555	4	24.9	0.00014 ***
Residuals	44	8		
---				
Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1				

In order to get means that have expectations  $\mu + \beta_1$ ,  $\mu + \beta_2$ ,  $\mu + \beta_3$  when there are an unequal number of replicates per cell, we should calculate the adjusted means. We compute the **adjusted means** (sometimes called **least-squares means** or **lsmeans** for short) by calculating the marginal means of the predicted cell means,  $(\mu + \alpha_i + \beta_j + \alpha\beta_{ij})$  obtained from the least squares estimates of the model parameters. Remember that the cell means are estimable functions. The R code below calculates the predicted cell means using the effect estimates from the model `mod2` created by the `lm` function shown above and then computes the adjusted or ls marginal means for the air/fuel

ratio using the R function `tapply`.

```
p <- data.frame( expand.grid( Eth = c(.1, .2, .3),
Ratio = c(14,15,16)

p[] <- lapply(p, factor)
p <- cbind( yhat = predict( mod2, p), p)
with(p, tapply(yhat, Ratio, mean) )
```

```
14      15      16
78.500 75.500 64.833
```

In these results, we can see that the means for the first two columns (14 and 15) are the same as the simple arithmetic average of the responses in the first two columns of the ration calculations in [Table of means](#). However, the mean from the third column is different, and it is a more accurate estimate of  $\mu + \beta_3$ . The R package `lsmeans` automatically computes the adjusted or `lsmeans`, and in addition it computes their standard errors and confidence limits. The R code below illustrates the use of this package to compute the marginal adjusted means for both ethanol and air/fuel ratio. The NOTE: printed by the `lsmeans` function tells us what we already know: interpretation of the marginal means may be misleading when there is a significant interaction.

```
library(lsmeans)
lsmeans(mod2,~ Eth)
```

```
Eth lsmean     SE df lower.CL upper.CL
0.1   66.8 0.963  8     64.6     69.1
0.2   75.8 0.963  8     73.6     78.1
0.3   76.2 1.112  8     73.6     78.7
```

```
Results are averaged over the levels of: Ratio
Confidence level used: 0.95
```

```
lsmeans(mod2,~Ratio)
```

```
Ratio lsmean     SE df lower.CL upper.CL
14      78.5 0.963  8     76.3     80.7
15      75.5 0.963  8     73.3     77.7
```

16      64.8 1.112 8      62.3      67.4

Results are averaged over the levels of: Eth  
Confidence level used: 0.95

In general the type II or III sums of squares and [1smeans](#) should be used, because they will test the correct hypotheses and provide unbiased factor level means whether there is an equal or unequal number of replications per cell.

### 3.5.4. Testing for Interaction with One Replicate per Cell

When there is adequate power for detecting main effects with  $r - 1$  replicate per cell, it would make sense to run a factorial design with only one observation per cell and  $a \times b$  total observations. Adding an additional replicate to each cell would double the effort, and it would usually not be required. However, with only one replicate per cell in a factorial design, there is no way to compute the ANOVA  $ssE$  and therefore no way to make  $F$ -tests on the main effects and interaction in the usual manner. If we assume the interaction term to be zero, then  $F$ -tests on the main effects can be made by using the additive model  $y_{ij} = \mu + \alpha_i + \beta_j + \varepsilon_{ij}$ . Even so, this could be risky if the interaction actually exists. There are ways we can determine whether the interaction is zero.

If the levels of both factors are quantitative as in the proposed UAV experiments or the ethanol fuel experiment, we can partition the sums of squares for the interaction term into **orthogonal polynomial single degrees of freedom**. For example, if there are three equally spaced quantitative levels of factor  $A$ , and three equally spaced quantitative levels for factor  $B$ , then the sums of squares for the interaction can be partitioned into four single degrees of freedom (namely: *linear*  $\times$  *linear*, *linear*  $\times$  *quadratic*, *quadratic*  $\times$  *linear*, and *quadratic*  $\times$  *quadratic*). Using the Taylor Series philosophy that low-order polynomials can approximate most functional relationships, we could assume that the three higher order terms are negligible and pooled to estimate the  $ssE$ . We could then use the  $ssE$  estimate as an error term to test the *linear*  $\times$  *linear* portion of the interaction. **Table 3-4** illustrates this with the data from the ethanol fuel experiment.

First, consider the averages of the two replicates in each cell of *Table 3-3* to be the result of a single experiment. The R code shown below averages the data in each cell to produce the data frame cells with one observation per cell. Fitting the *Eq. 3-2* to this data with the R function `lm` results in an ANOVA with zero degrees of freedom for  $ssE$ , and no  $F$ -tests.

```
library(daewr)
data(C0data)
Cellmeans <- tapply( C0data$C0, list(C0data$Eth,
                                         C0data$Ratio), mean )
dim(Cellmeans) <- NULL
Eth <- factor(rep(c(.1, .2, .3), 3))
Ratio <- factor(rep(c(14,15,16), each = 3))
cells <- data.frame( Eth, Ratio, Cellmeans )
modnr <- lm(Cellmeans ~ Eth*Ratio, data = cells )
anova(modnr)
```

The results follow.

#### Analysis of Variance Table

Response: Cellmeans						
	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Eth	2	162	81.0	NaN	NaN	
Ratio	2	326	163.0	NaN	NaN	
Eth:Ratio	4	339	84.7	NaN	NaN	
Residuals	0	0	NaN			

To get the sums of squares for the  $linear \times linear$  portion of the interaction, we convert the factors `Eth` and `Ratio` to *ordered factors* as shown.

```
Ethc <- as.ordered(cells$Eth)
Ratioc <- as.ordered(cells$Ratio)
```

When we use ordered factors, the R function `lm` uses orthogonal polynomial contrasts (see *Section 2.8.1*) for columns in the  $X$  matrix rather than the default treatment codings. In the R code below, the model `mbo` is fit using only the  $linear \times linear$  orthogonal polynomial contrasts for the interaction of `Ethc` and `Ratioc`.

```

EthLin<-contr.poly(Ethc)[Ethc,"L"]
RatioLin <-contr.poly(Ratioc)[Ratioc,"L"]
mbo <-lm(Cellmeans ~ Ethc + Ratioc + EthLin:RatioLin,
          data = cells)
anova(mbo)

```

### Analysis of Variance Table

```

Response: Cellmeans
            Df Sum Sq Mean Sq F value Pr(>F)
Ethc           2   162    81   16.2 0.0247 *
Ratioc          2   326   163   32.6 0.0092 **
EthLin:RatioLin 1   324    324   64.8 0.0040 **
Residuals       3     15      5
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

The error or residual sum of squares in this ANOVA table is the difference in the 4 degree of freedom interaction sums of squares shown in the Analysis of Variance Table above, and the single degree of freedom linear by linear interaction sums of squares. This difference is used to construct the denominator for the  $F$ -tests in the table above. The results show that the linear-by-linear portion of the interaction is significant and accounts for most of the interaction sums of squares. Since the interaction is significant, the additive model  $y_{ij} = \mu + \alpha_i + \beta_j + \varepsilon_{ij}$  is inappropriate, and the effects of the main effects will differ depending on the level of the other factor. We can interpret the results by examining the interaction plot.

We can generate the interaction plot that includes only the *linear*  $\times$  *linear* part of the interaction using the predictions from the model `mbo`. In the code below the R command `predict` is used to get the model predictions and the `aggregate` command is used to create the data frame `pred.means` that combines the model predictions with the factor levels. Next, the `interaction.plot` command is used as previously to create the plot.

```

Pred <-predict(mbo, newdata = data.frame(Ethc, Ratioc,
                                         EthLin, RatioLin))
pred.means <- aggregate(Pred,

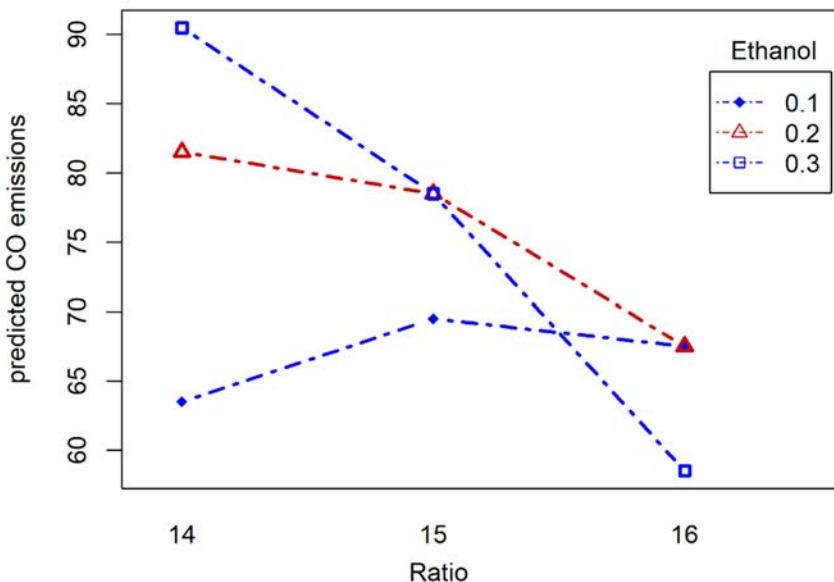
```

```

by = list(Ethc = Ethc, Ratioc = Ratioc), "mean")
Ethanol <- pred.means$Ethc
interaction.plot(pred.means$Ratioc, Ethanol,
                 pred.means$x, type = "b", pch = c(18,24,22),
                 leg.bty = "o", xlab = "Ratio", ylab = "predicted
CO emissions", col = c("blue","red3"), lty=4, lwd=2.5)

```

**Figure 3-7** shows the results, which we should compare with **Figure 3-6**. **Figure 3-7** is similar to **Figure 3-6** confirming what we saw in the ANOVA table (i.e., the majority of the variation caused by the interaction is captured in the linear-by-linear part).



**Figure 3-7. Linear by Linear Interaction Plot Ethanol and Air/Fuel Ratio**

When ethanol is at its high level (0.3) increasing air/fuel ratio from 14 to 16 causes a steep decrease in CO emissions. When ethanol is at its mid-level (0.2) increasing air/fuel ratio from 14 to 16 causes a slight decrease in CO emissions represented by the gentle negative sloping line. However, when ethanol is at its low level (0.1) increasing air/fuel ratio from 14 to 16 actually causes an increase in CO emissions illustrated by the positively sloped line.

When there is only one replicate per cell in a factorial experiment and

the factors do not have quantitative levels, partitioning the interaction sums of squares into *orthogonal polynomial contrasts* and combining the higher order terms as an **error sums of squares** may not be proper. However, Tukey (1949b) developed an alternate method for testing a single degree of freedom partitioned from interaction sums of squares. This method is equivalent to restricting the  $\alpha\beta_{ij}$  in **Eq. 3-2** of **Section 3.5** to be a second-degree polynomial function of the main effects  $\alpha_i$  and  $\beta_j$ , (see Scheff'e, 1959). By doing this, the sums of squares

$$ssAB = \frac{ab[\sum_i \sum_j y_{ij} \bar{y}_i \bar{y}_j - (ssA + ssB + aby^2_{..})y_{..}]}{(ssA)(ssB)} \quad \text{Eq. 3-7}$$

for testing the restricted hypothesis  $H_0: \alpha\beta_{ij} = 0$  for all  $i$  and  $j$  will have one degree of freedom. Moreover, the difference between  $ssAB$  and the error term for the additive model will form the error sums of squares similar to the example above with quantitative factor levels.

To illustrate the use of Tukey's single degree of freedom test for interaction, consider the data in **Table 3-5**, which is a portion of the data from a study to validate an assay of viral contamination reported by Lin and Stephenson (1998). We can use analyses of viral contamination to determine the presence (and amount) of a specific virus in biological products such as blood clotting Factor Eight. An experiment, or run, consists of making a solution with a known viral contamination (i.e., COVID-19), allowing the virus in a contaminated solution to grow, then measuring the result.

**Table 3-5.  $\log_{10}(\text{PFU/mL})$  Assay of Viral Contamination**

		Sample					
		1	2	3	4	5	6
Dilution	3	1.87506	1.74036	1.79934	2.02119	1.79934	1.59106
	4	1.38021	1.36173	1.25527	1.39794	1.20412	1.25527
	5	0.60206	0.90309	0.95424	1.00000	0.60206	0.60206

The experimental unit is the specific viral sample in combination with the place and time where it is allowed to grow. Factor  $A$  represents the sample number, or solution with which the viral sample is mixed (or spiked). Factor  $B$  represents different dilutions of the spiked sample. The

measured response is the  $\log_{10}$  of the plaque forming units per mL of solution.

Since factor  $A$  (sample) is not a quantitative factor it would be wrong to use orthogonal polynomial contrasts to partition its sums of squares or the sums of squares of its interaction with factor  $B$  (Dilution). To determine if the additive model  $y_{ij} = \mu + \alpha_i + \beta_j + \varepsilon_{ij}$  is applicable for this data, we test to see whether there is a significant interaction using Tukey's method. The function `Tukey1df` in the R package `daewr` calculates the non-additivity or interaction sums of squares, shown in **Eq. 3-7**, and prints a report. The code to produce the Tukey test output is shown below. The first column in the data frame used by this function is a numeric response, the second column is the indicator for the factor  $A$ , and the third column is the indicator for the factor  $B$ . The number of rows in the data frame should be exactly equal to the number of levels of factor  $A$  times the number of levels of factor  $B$ , since the design has no replicates.

```
library(daewr)
Tukey1df(virus)
```

Source	df	SS	MS	F	Pr>F
A	5	0.1948	0.039		
B	2	3.1664	1.5832		
Error	10	0.1283	0.0513		
NonAdditivity	1	0.0069	0.0069	0.51	0.4932
Residual	9	0.1214	0.0135		

In the results, we can see that the interaction (or non-additivity) is not significant. Therefore, for this data, it would be appropriate to fit the additive model,  $y_{ij} = \mu + \alpha_i + \beta_j + \varepsilon_{ij}$ , with the R function `lm` or `aov`.

### 3.6. Factorial Designs with Multiple Factors—CRFD

Two-factor factorial designs are more efficient than studying each factor separately in one-factor designs. Likewise, when many factors are under study, it is more helpful to study them together in a multi-factor factorial design than it is to study them separately in groups of two using two-factor factorial designs. When multiple factors are studied simultaneously, the power for detecting main effects is increased over

what it would be in separate two-factor factorial designs. Also, the possibility of detecting interactions among any of the factors is possible. If the factors were studied separately in two-factor factorials, two-factor interactions could only be detected between factors studied together in the same design. In a multi-factor factorial not only is it possible to detect two-factor interactions between any pair of factors, but it is also possible to detect higher order interactions between groups of factors. A three-factor interaction between factors *A*, *B*, and *C*, for example, means the effect of factor *A* differs depending on the combination of levels of factors *B* and *C*. Examples of higher order interactions will be presented in examples to follow.

The treatment combinations in a multi-factor factorial consist of all possible combinations of the levels of all factors. A design can be produced using `expand.grid` function in R (similar to the randomized plan created in [Section 3.4](#)), using the `gen.factorial` function in the `AlgDesign` package, or using functions from other packages that will be described later. The model for analysis is an extension of [Eq. 3-2](#), and the analysis can be made using the R function `lm` similar to the examples shown earlier.

A marketing firm may want to conduct an experiment pertaining to web sales of an online company called Anaconda. Suppose the marketing firm is considering adding aesthetics to their existing Web sites consisting of all possible combinations of the four factors described here.

The experimental units in this study will be individuals who visit the company Web site. The response is binary; the customer either signs up or does not. The factors under study were characteristics that change the appearance of the Web page. For example, factor *A* was the background alternatives for the page with three options. Factor *B* was the font size in the main banner, with three levels; factor *C* was the text color with two alternatives; and factor *D* was a choice between a sign-up button or link. Based on these factors there were  $3 \times 3 \times 2 \times 2 = 36$  possible configurations of the Web page when considering all possible combinations of the levels of each factor. A four-factor factorial experiment would consist of randomly assigning visitors to the Web site to one of the possible configurations and recording their binary

response. There are lurking variables that could affect the chance that a site visitor will sign up. For example, the position order that the link (for the company's Web site) comes up in a Web search for the products they sell, promotions offered by competitors, and attractiveness of competitors' Web sites. Random assignment of each sequential visitor to the site to one of the alternative configurations under study should minimize the chance of bias from changes in the lurking variables over time. The probability that a site visitor would sign up can be expressed by the model:

Each potential customer who visited the company's Web site during the trial period was randomly redirected to one of the 36 configurations. The number of visitors  $n_{ijkl}$  to the  $ijkl^{th}$  configuration and the number that signed up  $x_{ijkl}$  was logged.  $x_{ijkl}$  is then binomially distributed

$$p_{ijkl} = \mu + \alpha_i + \beta_j + \alpha\beta\gamma_k + \alpha\gamma_{ik} + \beta\gamma_{jk} + \alpha\beta\gamma_{ijk} \\ + \delta_l + \alpha\delta_{il} + \beta\delta_{jl} + \alpha\beta\delta_{ijl} + \gamma\delta_{kl} \\ + \alpha\gamma\delta_{ikl} + \beta\gamma\delta_{jkl} + \alpha\beta\gamma\delta_{ijkl} \quad \text{Eq. 3-8}$$

where  $\alpha_i$  represents the effect of background choice,  $\beta_j$  represents the effect of font size in the main banner,  $\gamma_k$  represents the effect of text color, and  $\delta_l$  the effect of sign-up link versus button.

$$B(x_{ijkl}, n_{ijkl}, P_{ijkl}) = \binom{n_{ijkl}}{x_{ijkl}} p_{ijkl}^{x_{ijkl}} (1 - p_{ijkl})^{n_{ijkl} - x_{ijkl}} \quad \text{Eq. 3-9}$$

where  $n_{ijkl}$  is the number of visitors to the  $ijkl^{th}$  configured Web page during the testing period.

This is a unique problem but one that is more common than it was as little as 10 years ago. This study is looking for the probability that certain groups of people will take some action. Traditional DOE problems dealt with which fertilizer to use to grow better corn, or which concentration of compounds to add to battery electrodes to increase their discharge (see **Section 3.7**), or which treatment process to use to improve aerogels used in space exploration (see Sections 3.7 and 3.9).

### **3.7. Example: Screening Variables Affecting Zinc Battery Electrodes**

Consider an example of a multi-factor factorial design in Technology development. A defense research laboratory who performs research and development activities on zinc-silver oxide batteries for special applications, USAELRDL is investigating the preparation of zinc electrodes by dry processes. These involve the application of dendritic zinc powders under pressure to the grids. The interlocking properties of the dendritic zinc particles make it possible to form the electrodes with moderate pressures such that the porosity and related high surface area of the electrode is not destroyed. It is expected that dry process zinc electrodes will have many advantages over conventional electrodeposited sponge zinc electrodes, including higher discharge efficiency, greater uniformity of performance and better adaptability to mechanized production with resultant economics. (Wilburn, 1963)

Due to the large number of variables affecting the discharge performance of the electrodes, the lab decided to design and conduct a factorial experiment with multiple factors each having two levels, high and low to isolate the significant variables. These variables, and any controlling interactions between them, could then be studied further to arrive at the optimum conditions for the production of electrodes of maximum discharge efficiency. (In Chapter 6, we'll perform a fractional factorial experiment for this problem)

The experimental units in this study will be individual batteries. The response is the discharge efficiency of the electrode expressed in percent as the ratio of the output capacity to the theoretical capacity of the zinc active material. For comparison the conventional sponge zinc electrodes give an average efficiency of about 20% under comparable discharge conditions. The average of the sixteen responses is 31.2%. Moreover, the lab used a technique known as the Yates' Algorithm which is a rapid method for obtaining the same mean effects that would be obtained from a formal and lengthy analysis of variance. The Yates' Algorithm is applicable to any factorial experiment. Its advantages become more apparent the larger the experiment (we'll have more to say about this in **Section 3.8**).

### 3.7.1. Experiment Variables

The factors under study were variables that have been significant in similar studies as well as theoretical ones. There were two categories of variables in the electrode investigation, those related to the electrode preparation itself (see **Table 3-6**) and those related to the electrolytic formation of the dendritic zinc powders (see **Table 3-7**). Although several more variables were considered, the lab decided to limit the number of variables to eight, keeping all other factors constant.

*Table 3-6. Pressing Variables in the battery experiment*

Pressing Variables		Units	High	Low
A	Zinc weight	grams	5.23	2.62
B	Pressure	psi	1,840	1,230
C	Particle size	sieve mesh	100	200
D	Pressure time	minutes	15	1
E	Pressure temp.	OF	300	80

*Table 3-7. Formative Variables in the battery experiment*

Formation Variables		Units	High	Low
F	ZnO in electrolyte	gm./liter	20	0
G	Electro to temp	F	100	0
H	Current density	amp./sq. in.	1.0	0.75

Variable *G*, the electrolyte temperature during the plating operation, turned out to be impossible to control with the plating equipment which was prepared for the experiment and with the selected plating current densities.

### 3.7.2. The Experimental Design

Having established the variables and the high and low levels, the fractional replicate design was established, as shown in **Figure 3-8**. This is the first design (we'll use a second design in Chapter 6). The design involves eight variables (or seven) each at two levels to be studied with a total of sixteen different electrodes. A full  $2^8$  factorial experiment, eight variables each at two levels, would involve  $2^8$  or 256 trials (128 trials, for a  $2^7$  factorial).

Since we'll revisit this problem in Chapter 6, we simplify it here, omitting the variables *G* and *H*. Therefore, this design represents a  $2^{8-2}$  factorial (technically we're using a fractional factorial  $2^{8-4}$  design that we'll see

in Chapter 6). The design is based on extension of a basic  $2^4$  or 16 trials. Thus the first four variables are arranged in standard order for the  $2^4$  factorial. The other variables are then introduced by making a basic assumption that three factor interactions between the first four variables are negligible. Therefore  $E$  is introduced by equating it to the interaction between variables  $A$ ,  $B$ , and  $C$ . Regarding the high level as plus and the low level as minus, the level of  $E$  for the first box is  $-x - x - = -$ , for the second box  $+x - x - = +$ , etc. Similarly, variable  $F$  is introduced to equating it to the  $BCD$  interaction, variable  $G$  to the  $ABD$  interaction and variable  $H$  to the  $ACD$  interaction. Dropping the variables  $G$  and does not affect the  $2^{8-2}$  experiment in any way.

No.	A	B	C	D	E	F	G	H	Response
1									33.6
2	Y				Y		Y	Y	28
3	Y	Y			Y	Y	Y	Y	33.2
4	Y	Y	Y		Y	Y	Y	Y	23.9
5	Y	Y	Y	Y	Y	Y	Y	Y	23.8
6	Y	Y	Y	Y	Y	Y	Y	Y	30.6
7	Y	Y	Y	Y	Y	Y	Y	Y	40
8	Y	Y	Y	Y	Y	Y	Y	Y	23.4
9	Y	Y	Y	Y	Y	Y	Y	Y	34.3
10	Y	Y	Y	Y	Y	Y	Y	Y	28.6
11	Y	Y	Y	Y	Y	Y	Y	Y	33.3
12	Y	Y	Y	Y	Y	Y	Y	Y	40.3
13	Y	Y	Y	Y	Y	Y	Y	Y	38.4
14	Y	Y	Y	Y	Y	Y	Y	Y	29.8
15	Y	Y	Y	Y	Y	Y	Y	Y	38
16	Y	Y	Y	Y	Y	Y	Y	Y	19.7

Figure 3-8. Design matrix for the battery electrode experiment (yellow represents low and green represents high)

The electrode discharge can be expressed by the modified model:

$$\text{Discharge} = c_0 + c_1A + c_2B + c_3C + c_4D + c_5E + c_6F + \varepsilon$$

Below is the R code to open the raw data and print the first six lines of the data frame.

```
batts <- read.csv(
  "https://raw.githubusercontent.com/stricje1/Data/master/
  batteries2.csv")
head(batts)
```

	No.	A	B	C	D	E	F	G	H	Response
1	1	2.62	1230	100	15	300	20	100	1.00	33.6
2	2	5.23	1230	100	15	80	20	0	0.75	28.0
3	3	2.62	1840	100	15	80	0	0	1.00	33.2
4	4	5.23	1840	100	15	300	0	100	0.75	23.9
5	5	2.62	1230	200	15	80	0	100	0.75	23.8
6	6	5.23	1230	200	15	300	0	0	1.00	30.6

The correct procedure must be utilized to analyze the data, determine if any of the factor effects are significant, and to predict the optimal technology improvement to zinc electrodes. The arcsin square root transformation shown in *Table 2-4* of **Section 2.6.2** could be applied and the R `lm` function could be used for analysis. However, the problem with using this procedure is that the individual responses were summed to get the aggregate responses, and when using these aggregate responses there are no replicate observations in any of the cells, and thus no way to calculate  $ssE$ . This would be similar to summing or averaging the replicate responses in each cell if the data were normally distributed, leaving only one observation per cell and no way to compute  $ssE$ . The alternative is to use the method of maximum likelihood to fit model **Eq. 3-8**. This can be done using the R `glm` function. It will automatically set  $\sigma^2 = 1.0$  and the type III sums of squares of the form  $(\mathbf{L}\hat{\beta})'(\mathbf{L}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{L}')^{-1}(\mathbf{L}\hat{\beta})$  will be asymptotically distributed as chi-squares under the null hypothesis.

Before proceeding further, we'll formulate the problem as described above by manipulating the data, including factorizing the independent variables and forming a dataframe. The R code is shown here.

```
# Define variables to drop
drop <- c("No.", "G", "H", "Response")
# Drop variables and form a dataframe
df = batts[, !(names(batts) %in% drop )]
# Convert the dependent variables to factors
bat_fac <- factorize(c(2,2), df)
```

```

# Rename the response variable
zinc <- batts$Response
# Generate a datafame that include the response, zinc
batb <- data.frame(df,zinc)
head(batb)

```

	A	B	C	D	E	F	zinc
1	2.62	1230	100	15	300	20	33.6
2	5.23	1230	100	15	80	20	28.0
3	2.62	1840	100	15	80	0	33.2
4	5.23	1840	100	15	300	0	23.9
5	2.62	1230	200	15	80	0	23.8
6	5.23	1230	200	15	300	0	30.6

The commands to analyze the data using `glm` are shown here.

```

mod_a <- glm( zinc ~ A+B+C+D+E+F, data = batb,
               family = gaussian() )
anova(mod_a, test = "Chisq")

```

The option `family = gaussian` declares the response to be normally distributed, and the option `test = "Chisq"` in the call to the `anova` function requests a table of the type III sums of squares and chi-square tests. A portion of the results are shown below. The command `summary(modb)` prints a table of the parameter estimates produced by the maximum likelihood method, which is similar to the summary of an object created by the function `lm`, and it is not shown here.

```

Analysis of Deviance Table
Model: gaussian, link: identity
Response: zinc
Terms added sequentially (first to last)

```

	Df	Deviance	Resid. Df	Resid. Dev	Pr(>Chi)						
NULL			15	603							
A	1	158.1	14	444	0.0087 **						
B	1	1.4	13	443	0.8064						
C	1	8.3	12	435	0.5487						
D	1	41.9	11	393	0.1768						
E	1	110.8	10	282	0.0281 *						
F	1	75.3	9	207	0.0704 .						
---											
Signif. codes:	0	'***'	0.001	'**'	0.01	'*'	0.05	.	0.1	' '	1

Here are some model metrics in a table.

```

# create matrix with 3 columns and 1 rows
data = matrix(c(mod_a$deviance, mod_a$NULL.deviance,
    mod_a$aic),
    ncol=3, byrow=TRUE)
# specify the column names and row names of matrix
colnames(data) = c('Model Deviance |',
'Model NULL Deviance |', 'Model AIC')
rownames(data) <- c('Value')
# assign to table
Final = as.table(data)
# display
final

```

	Model Deviance	Model Null Deviance	Model AIC
Value	206.83	602.56	102.36

In this output we can see that (at the  $\alpha = 0.05$  level of significance) factors  $A$  (*Zinc weight*) and factor  $E$  (*Pressure temp.*) were significant. Recall that to the interaction between variables  $A$ ,  $B$ , and  $C$ , where factor  $C$  represents the *Particle Size*. Since there is a significant interaction, the main effects  $A$  and  $B$  cannot be interpreted separately. Whether it is better to change the *Zinc Weight*, or the *Particle Size* depends on whether the *Pressure Time* changes. To interpret the three-way interaction, it is necessary to make a table of the proportion signing up in each combination of factors  $A$ ,  $B$ , and  $C$  and a series of interaction graphs. The R code to do this are shown below.

```

par ( mfrow = c(1,1) )
resp <- batb$zinc
batp <- data.frame(batb, resp)
par ( mfrow = c(1,3) )
batp1 <- subset(batp, A == 2.62)
interaction.plot(batp1$C, batp1$B, batp1$resp,
    type = "l", legend=FALSE, ylim = c(30,40),
    main = "Factors B & C", xlab = "PSI",
    ylab = "Electrode Discharge", lwd = 2,
    col=c("red", "blue"))
batp2 <- subset(batp, A == 5.23 )
interaction.plot( batp2$C, batp2$D, batp2$resp,
type = "l", legend = FALSE, ylim =
c(20,35), main = "Factors C & D", xlab =

```

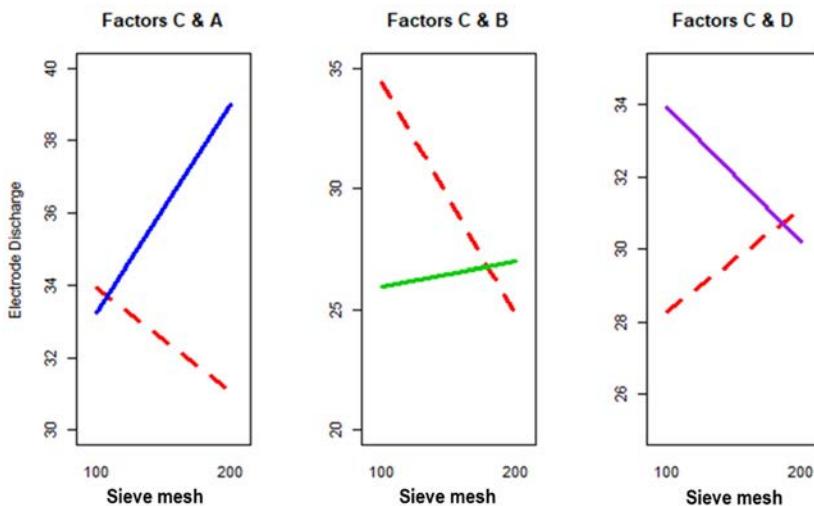
```

"Sieve mesh", ylab = "", lwd = 2,
  col=c("red", "blue"))
batp3 <- subset(batp, B == 1230)
interaction.plot(batp3$C,batp3$E,batp3$resp,type = "l",
legend=FALSE, ylim = c(25,35),
  main="Factors D & E", xlab = "Minutes", ylab = "",
  lwd = 2, col=c("red", "blue"))
par ( mfrom = c(1,1) )

```

The result is shown in **Figure 3-9**. It shows the effect of factor C (*Particle Size*) for each combination of the levels of factors A (*Zinc weight*), B (*Pressure*), and D (*Pressure Time*).

The common way of interpreting the interaction is to compare the effect of the variable represented on the horizontal axis between combinations of levels of the other factors. For this example an interpretation can be made as follows. When decreasing the *Particle Size* (or increasing the sieve mesh, we can see that changing the weight from Low to High causes an increase in the electrode discharges. The increase (represented by the slope of the lines) is greater when Zinc Weight is low. However, when Zinc Weight is high, the effect of Particle Size is altogether different. In this case changing the *Sieve Mesh* from 100 to 200 causes a decrease in electrode discharge.



**Figure 3-9. Particle Size Effect by Pressing Variable**

Any one of the three factors could be placed on the horizontal axis and an equivalent interpretation could be made. Sometimes the interpretation that results when placing the factor with the largest main effect on the horizontal axis is easiest to explain.

When interpreting a three-factor interaction, only one graph was necessary to illustrate the fact that the effect of one factor depended on the level of another factor. However, in this case more than one graph is required to illustrate how the effect of one factor depends on the combination of levels of the other two factors. The two lines on each graph show how the effect of text color changes when there is a sign-up link versus a button, and the different graphs show how the effect changes when the background is changed.

From inspection of the three graphs, or the table of averages that could be produced with the `tapply` function, it can be seen that the highest discharge is reached when the Particle size is high (Sieve mesh = 1000).

```
# create matrix with 3 columns and 1 rows
Data = matrix(c(CellmeansA, CellmeansB, CellmeansC,
    CellmeansD), ncol = 4, byrow = TRUE)
# specify the column names and row names of matrix
colnames(data) = c('Cell Means A |', 'Cell Means B
    |', 'Cell Means C |', 'Cell Means A |')
rownames(data) <- c('High', 'Low')
# assign to table
Final = as.table(data)
# display
final
```

	Cell Means A	Cell Means B	Cell Means C	Cell Means A
High	34.3250	28.0375	30.8875	31.4750
Low	31.9000	30.4625	32.8000	29.5625

### **3.8. Two-Level Factorials**

As additional factors are added to a factorial design, the number of treatment combinations (runs) in the design increases exponentially. The example in the last section contained four factors and 36 treatment

combinations. If there were five factors in a design each having four levels, the number of treatment combinations would be  $4 \times 4 \times 4 \times 4 \times 4 = 1024$  runs in the design. It can be seen that it would not take too many factors to render the design impractical.

In other words, it would have too many treatment combinations to run in a reasonable period of time. However, it is better to reduce the number of levels of each factor and stay with the factorial design using all factors than it is to revert to one-at-a-time or two-at-a-time experiments and lose the efficiency of factorial experiments. With separate experiments the ability to detect higher order interactions, and the ability to detect interactions between any pair of factors, is lost. If five factors in a factorial design were studied with only two levels each, the number of treatment combinations would be reduced to  $2^5 = 32$ . For this reason factorial designs with two levels for each factor, or two-level factorials, are popular. A shorthand for a two-level factorial with  $k$  factors is a  $2^k$  design.

In two-level factorials, if a factor has quantitative levels, the two levels are denoted symbolically by (-) and (+), where (-) represents the lowest level the experimenter would consider, and (+) represents the highest level the experimenter would consider. The high and low are usually spread out as far as feasibly possible in order to accentuate the signal or difference in response between the two levels. If a factor has qualitative levels, the (-) and (+) designations are arbitrary, but the two levels chosen normally would be two that the experimenter believes should result in the maximum difference in response.

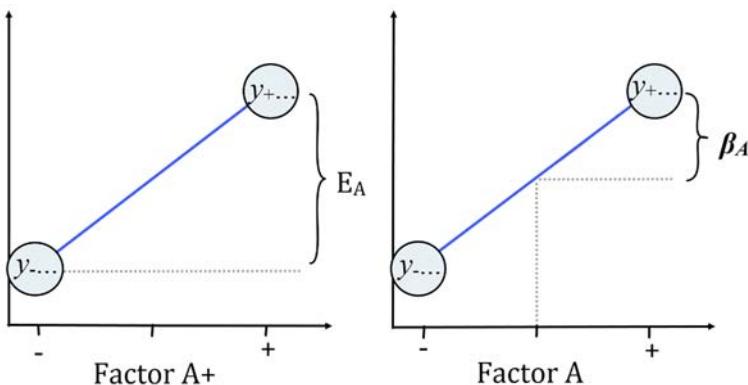
### 3.8.1. Main Effects and Regression Slopes

The model for a factorial experiment with three factors can be written as:

$$y_{ijkl} = \mu + \alpha_i + \beta_j + \alpha\beta_{ij} + \gamma_k + \alpha\gamma_{ik} + \beta\gamma_{jk} + \alpha\beta\gamma_{ijk} + e_{ijkl} \quad \text{Eq. 3-10}$$

where  $\alpha_i$ ,  $\beta_j$ , and so forth, are the effects as defined earlier. However, in the case where each factor has only two levels represented by (-) and (+),  $i, j, k, l$  can be replaced with either (-) and (+), and  $\alpha_- = -\alpha_+$ , since

$\alpha_- = y_{-...} - y_{....}$ ,  $\alpha_+ = y_{+...} - y_{....}$ , and  $y_{....} = \frac{y_{-...} + y_{+...}}{2}$ . A similar equality will be true for all the effects and interactions. Since the two effects for each factor are the same value with different signs, a more compact way of defining the main effects for a two-level factorial is  $E_A = y_{+...} - y_{-...}$ . This can be visualized on the left side of **Figure 3-10** and represents the change in the average response caused by a change in the factor from its low (-) level to its high (+) level. This effect can then be represented by the difference in two averages  $y_{+...}$  and  $y_{-...}$ .

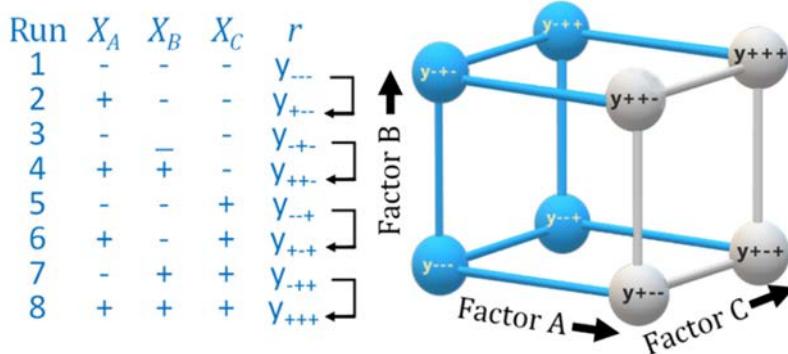


**Figure 3-10. Effect and Regression Coefficient for Two-Level Factorial**

The regression slope  $\beta_A$  shown in the right side of **Figure 3-10** is the vertical change in the average response for a one-unit change (i.e., from 0 to +1) in the factor level in symbolic units. Therefore the slope,  $\beta_A$ , is just one half the effect,  $E_A$ , or the difference in two averages divided by 2.

The treatment combinations in a two-level factorial can also be represented geometrically as the corners of a cube as shown in **Figure 3-11**. On the left side of this figure is a list of the treatment combinations or runs listed in standard or Yates' order with the first column changing fastest with alternating – and + signs, the second column changing in pairs of – and + signs, the second column changing slowest in groups of four – and + signs. The treatment combinations in two-level factorial designs have traditionally been written in standard order to facilitate the computation of main effects and interaction effects by hand using Yates' algorithm (see Daniel (1976)).

The main effect for factor  $A$  can be visualized in the figure as the difference of the average of the responses on the right side of the cube in the grey-shaded circles and the average of the responses on the left side of the cube in the white circles. With modern computer programs such as the R `lm` function one half of the main effects, or regression coefficients (shown on the right side of **Figure 3-10**), can be computed by regression and we no longer need Yates' algorithm.



$$E_A = (y_{+-..} + y_{++..} + y_{-+..} + y_{-++..})/4 - (y_{...} + y_{+..} + y_{-+..} + y_{-++..})/4$$

**Figure 3-11. Geometric Representation of  $2^3$  Design and Main Effect Calculation**

One of the desirable properties of a  $2^k$  factorial plan is that factor effects are not obscured by planned changes in other factors. In the list of experiments for  $2^k$  design, shown in **Figure 3-11**, this is evident by the fact that at the high level of each factor, there are an equal number of high and low levels of every other factor. Also at the low level of each factor, there are an equal number of high and low levels of every other factor. Thus the effect of a factor, or difference in average response between the high and low level of that factor, represents the effect of that factor alone, because the influence of all other factors has been averaged out. Mathematically this property is called **orthogonality**.

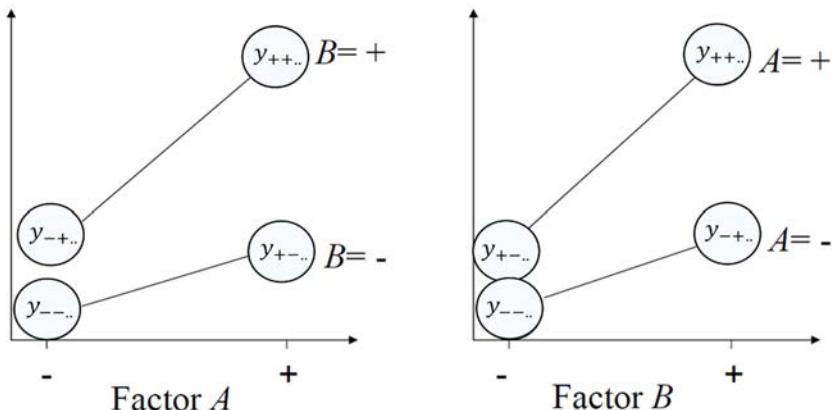
### 3.8.2. Interactions

When all the factors have only two levels, the  $AB$  interaction effect is defined as one-half the difference in the simple effect of factor  $A$ , ( $y_{+..} - y_{-..}$ ), factor  $B$  is held constant at its high (+) level, and the simple effect of factor  $A$ , ( $y_{+-..} - y_{-+..}$ ), when factor  $B$  is

he(l+d)constant at its low (-) level, that is,  $((y_{++..} - y_{-+..}) - \frac{y_{+-..} - y_{--..}}{2})$ .

This is illustrated on the left side of **Figure 3-12**. The interaction effect could also be defined as one half the difference in the simple effect of factor  $B$ ,  $(y_{-+..} - y_{-+..})$ , when factor  $A$  is held constant at its high (+) level, and the simple effect of factor  $B$ ,  $(y_{-+..} - y_{--..})$ , held constant at its low (-) level. This is illustrated on the right side of **Figure 3-12**. Either way the interaction effect is  $E_{AB} = (y_{++..} + \frac{y_{--..}}{2} - \frac{y_{+-..} + y_{-+..}}{2})$  is the difference of two averages.

It is easy to determine which responses should be averaged and which average should be subtracted from the other in order to calculate an interaction effect as illustrated in **Figure 3-13**. To calculate the  $AB$  interaction, we add a column of signs,  $X_A \cdot X_B$  to the list of treatment combinations on the left side of the figure. The elements in this new column are just the elementwise products of signs in the column for  $X_A$  and  $X_B$  (i.e.,  $(-)(-) = +$ ,  $(-)(+) = -$ , etc.).



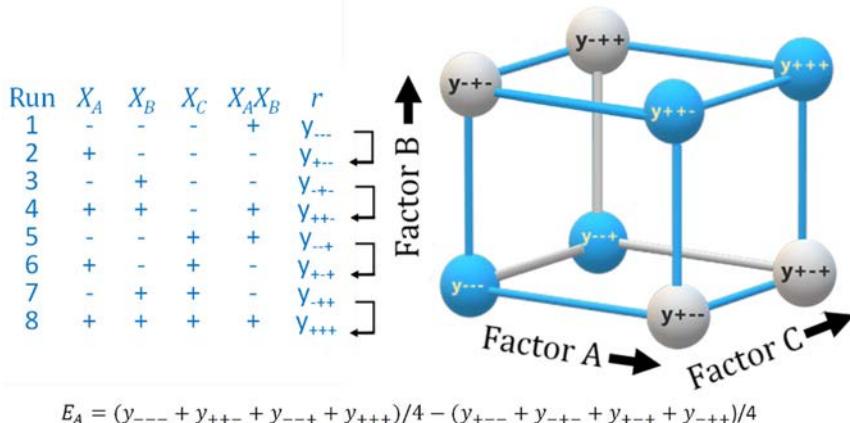
**Figure 3-12. Definition of an Interaction Effect for Two-Level Factorial**

Now the interaction effect can be visualized in the figure as the difference in the average response on one diagonal represented by grey circles and the average response on the other diagonal represented by white circles. From this representation, it can also be seen that interaction effects are not obscured by planned changes in other factors, or in other words they are orthogonal to main effects.

One-half of this interaction effect (or the regression coefficient) can be

calculated using a regression program such as the R `lm` function by adding a  $X_A \times X_B$  term to the model. Higher order interaction effects can be similarly defined. Therefore, a simpler way of writing the model for a two-level factorial is by using the familiar regression equation,

$$y = \beta_0 + \beta_AX_A + \beta_BX_B + \beta_{AB}X_AX_B + \beta_CX_C + \beta_{AC}X_AX_C + \beta_{BC}X_BX_C + \beta_{ABC}X_BX_AX_C + \varepsilon \quad \text{Eq. 3-11}$$



$$E_A = (y_{---} + y_{-+-} + y_{-+} + y_{-++})/4 - (y_{-+-} + y_{-+} + y_{-++} + y_{-++})/4$$

**Figure 3-13. Geometric Representation of  $2^3$  Design and Interaction Effect**

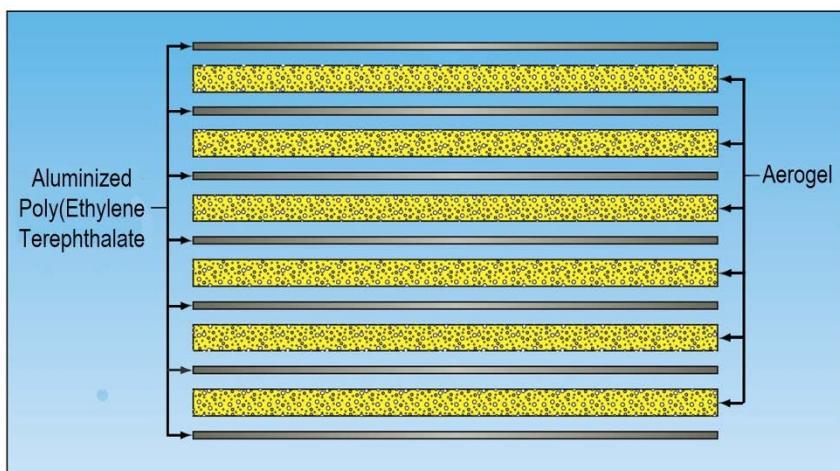
where the  $\beta$ s are one-half of the effects and  $X_A = -1$  if factor A is at its low level and  $X_A = 1$  if factor A is at its high level. If we write this model in matrix terms,  $y = \mathbf{X}\boldsymbol{\beta} + \varepsilon$ , the orthogonality property of the design is expressed by the fact that the columns of the  $\mathbf{X}$  matrix are orthogonal and the  $\mathbf{X}'\mathbf{X}$  matrix is diagonal with diagonal elements  $r2^k$ , where  $r$  is the number replicates of each cell.

### 3.9. Example - Silica Aerogel 2k Factorial

#### 3.9.1. Introduction

Aerogels comprise a special class of low-density open-cell solid foams (typically, with porosity over 90%) which exhibit many unique properties such as exceptionally lightweight, high surface area, low thermal conductivity, extremely low dielectric constant, low sound wave transmission, high optical transparency in a wide range of wavelengths close to that of glass, and a very low refractive index (Fricke, 1988)

(Emmerling, et al., 1995) (Ma, Roberts, Prévost, Jullien, & Scherer, 2000) (Woignier & Phalippou, 1987). These properties result from the microstructure of aerogels, which consists of a three-dimensional amorphous solid skeleton network with interconnected nanometer-sized pores in between. Silica aerogels are used for thermal and electrical insulation (see **Figure 3-14**), especially in **space applications, oxygen and humidity sensors, aerosol particle collectors, space mirror protectors, catalyst supports, battery electrodes**, etc. (Woignier & Phalippou, 1988). NASA Glenn Research Center and the Ohio Aerospace Institute have recently demonstrated that templated polymerization of *di-, tri-, and tetra-*-isocyanates on the surface of the nanoparticle building blocks of silica aerogels increases resulting conformal coatings increase the density of the native aerogels by a factor of 2-3 but the strength of the resulting materials may increase by more than two orders of magnitude.



**Figure 3-14.** An Improved Aerogel Vacuum Insulation Panel contains multiple layers of aerogel interspersed with layers of aluminized poly(ethylene terephthalate). The panel is shown here in the uncompressed form at an intermediate stage of fabrication. Once the interior of the panel is evacuated, exterior atmospheric pressure squeezes the layers together.

Aerogels are used in multilayer insulation for space and air vehicles, lightweight aircraft antennae, thermal shields (re-entry), liquid nitrogen boil-off, cryogenic systems, and aerogel-based insulation systems are for Mars exploration. The mesoporous surfaces of *tetramethoxysilane*

(TMOS)-derived silica aerogels have been modified with amines by copolymerization of TMOS with *aminopropyltriethoxysilane* (APTES). The amine sites have become anchors for crosslinking the nanoparticles of the skeletal backbone of the aerogel by attachment of *di*-, *tri* and *tetra*-functional epoxies. Processing variables such as amount of APTES used to make the gels, the epoxy type and concentration used for crosslinking, as well as the crosslinking temperature and time were varied according to a multivariable DOE model. They found that while elastic modulus follows a similar trend with density, maximum strength is attained neither at the maximum density nor at the highest concentration of -NH<sub>2</sub> groups. Aerogels crosslinked with the *tri*-functional epoxide always show improved strength compared with aerogels crosslinked with the other two epoxides under identical conditions. Solid <sup>13</sup>C NMR studies show residual unreacted epoxides, which condense with one another by heating crosslinked aerogels at 150 °C.

The effect of polymer accumulation on the particles has been quantified as a function of the processing parameters, *a*, *e*, *c*, *t* and  $\theta$  as described above using a statistical experimental design approach, by following aerogel properties such as physical dimensions, density, surface area, porosity, strength, and flexibility. It was deemed reasonable to assume that linear and nonlinear effects of any variable on any physical property could be captured adequately by a full quadratic model of the form:

$$\begin{aligned}
 \text{Physical property} &= A + Ba + Ce + Dc + Et + F\theta + Ga^2 \\
 &\quad + Hc^2 + It^2 + J\theta^2 + Kae + Lac + Mat \quad \text{Eq. 6.1} \\
 &\quad + Nat + Oec + Pe\theta + Qet + Rc\theta + Set \\
 &\quad + Tt\theta
 \end{aligned}$$

where *A* through *T* are coefficients that would be derived empirically from experimental data. The model contains terms for first order effects of all five variables and second order terms for *a*, *c*, *t* and  $\theta$ , as well as all possible two-way interaction terms. (Owing to the discrete nature of variable epoxy type, *e*, (epoxy type) there is no physical meaning to a second order term, *e*<sup>2</sup>.) To evaluate first and second order terms for *a*, *c*, *t* and  $\theta$ , a minimum of two levels of each variable must be considered. The three levels of variable *a* (APTES Percent) were 0%, 25% and 50%

v/v APTES in TMOS+APTES. The three levels of variable  $c$  (Epoxy Percent) were 15%, 45% and 75% v/v epoxy in THF+epoxy. Variable Time was evaluated as *Low* ( $\leq 50\text{hr}$ ) and *High* otherwise. The  $t$  or Temp variable is assigned two levels, *Low* @  $\leq 72$  and otherwise *High*. The discrete variable epoxy type,  $e$ , was also considered at three levels corresponding to the *di*-, *tri*- and *tetra*-functional epoxies discussed.

In Chapter 5, we consider a full-factorial design to evaluate this model (the original approach), which would contain at least 243 experiments ( $3^5$  experiments representing three levels each of five variables), not counting repeats. However, for now we want to evaluate a  $2^k$  model with  $k = 2$  using the set of experimental runs is computer-generated from the 243 candidate experiments. In total, there are only 24 usable samples. Thus, at  $2^2 = 4$  experiments are required, we can repeat experiments. To develop the  $2^2$  model, we must have at least two factors with two levels each (i.e., Time and Temp). These were prepared according to Scheme 1 in random order, and were analyzed for their physical dimensions, density, surface area, porosity, strength, and flexibility. **Table 3-8** summarizes the design runs and the experimental results.

**Table 3-8: Aerogel density modification using epoxies**

Run	Density	APTES_percent	Epoxy_type
Epoxy_percent			
Min. : 1	Min. :0.2000	Min. : 0.00	Length:33
Min. :15			
1st Qu.: 9	1st Qu:0.3000	1st Qu.:25.00	Class :character
1st Qu.:15			
Median :17	Median:0.4200	Median :25.00	Mode :character
Median :45			
Mean :17	Mean :0.3876	Mean :25.97	Mean :44
3rd Qu.:25	3rd Qu:0.4700	3rd Qu.:50.00	3rd Qu.:62
Max. :33	Max. :0.5900	Max. :50.00	Max. :75
Time	Temp	Surface_Area	Average_Pore_Diam
Min. :16.00	Min. :50.00	Length:33	Length:33
1st Qu.:44.00	1st Qu.:50.00	Class :character	Class :character
Median :44.00	Median :72.50	Mode :character	Mode :character
Mean :46.55	Mean :70.42		
3rd Qu.:72.00	3rd Qu.:72.50		

```
Max.    :72.00   Max.    :95.00
Load_Force      Max_Stress        Modulus Weight_loss_percent
Length:33       Length:33        Length:33       Length:33
Class :character Class :character Class :character Class
:character
Mode :character Mode :character Mode :character Mode
:character
```

The density of the samples, surface area, maximum stress to breakpoint, average pore diameter, and load force, are response variables determined from their physical features. Surface area and pore diameters were determined by nitrogen adsorption *porosimetry*. Mechanical strength data (i.e., stress at break point and elastic modulus) were obtained by a three-point bend test method.

```
library(AlgDesign)
library(daewr)
library(DoE.base)
library(FrF2)
library(leaps)
library(lme4)
```

### 3.9.2. Data Dictionary

The data elements include:

- Run Number
- Density in grams/centimeters ( $g/(cm^3)$ )
- Epoxy Type: Three different epoxy crosslinkers were used
- Time in hours ( $h$ ): duration of chemical treatment
- Temperature in degrees Celsius ( $^{\circ}C$ )
- Epoxy percent used in treatment
- Surface area in centimeter-squared per gram ( $cm^2$ )
- Average pore diameter in angstroms ( $\text{\AA}$ )
- Load force in kilograms (kg): the force exerted on a surface or body
- Max Stress at rupture in newtons per meters squared
- Modulus of rupture in mega-pascals (MPa): maximum bending stress that can be applied
- Weight loss percent during treatment

For our purposes we will use a subset of these factors (see **Table 3-9**) each having three levels

**Table 3-9. Table of factors and levels**

Factor	Attribute Levels
APTES_percen	1 = 9
	2 = 25
	3 = 50
Epoxy_percent	1 = 15%
	2 = 45%
	3 = 75%
Epoxy_type	1 = di-epoxy
	2 = tri-epoxy
	3 = tetra-epoxy
Time	1 = Low
	2 = High
Temp	1 = Lo
	2 = High

### 3.9.3. Factorial design $2^2$ : Input values

For this design and analysis, we enter the treated aerogel densities directly as a one-column matrix.

```
Density <- matrix(c(0.42, 0.42, 0.44, 0.30, 0.48, 0.47, 0.51,
0.32, 0.30, 0.49, 0.59, 0.36, 0.28, 0.49, 0.42, 0.48,
0.32, 0.40, 0.24, 0.44, 0.41, 0.32, 0.47, 0.49, 0.47,
0.33, 0.29, 0.45), byrow = T, ncol = 1)
head(Density)
```

```
[,1]
[1,] 0.42
[2,] 0.42
[3,] 0.44
[4,] 0.30
[5,] 0.48
[6,] 0.47
```

### 3.9.4. Define and Enter Input Values

Next, we enter the basic measurement information for the two-factor ( $2^k$ ), where  $k = 2$ . Since we are treating the response with factor having two levels, our design is  $2^2 = 4$ , so we have enough data for six

runs.

```
A <- rep(c(-1, 1), 14)
B <- rep(c(-1, -1, 1, 1), 7)
AB <- A*B
Total <- apply(Density, 1, sum)
n <- 3
cbind(A, B, AB, Total)
```

	A	B	AB	Total
[1,]	-1	-1	1	0.42
[2,]	1	-1	-1	0.42
[3,]	-1	1	-1	0.44
[4,]	1	1	1	0.30
[5,]	-1	-1	1	0.48
[6,]	1	-1	-1	0.47
[7,]	-1	1	-1	0.51
[8,]	1	1	1	0.32
[9,]	-1	-1	1	0.30
[10,]	1	-1	-1	0.49
[11,]	-1	1	-1	0.59
[12,]	1	1	1	0.36
	...			
[22,]	1	-1	-1	0.32
[23,]	-1	1	-1	0.47
[24,]	1	1	1	0.49
[25,]	-1	-1	1	0.47
[26,]	1	-1	-1	0.33
[27,]	-1	1	-1	0.29
[28,]	1	1	1	0.45

### 3.9.5. Effects

Here we apply effects for each of the three variables examined ( $A$ ,  $B$ ,  $AB$ ).

```
Abeff <- (Total %% AB)/(2*n)
Effects <- t(Total) %% cbind(A,B,AB)/(2*n)
Summary <- rbind( cbind(A,B,AB), Effects )
head(Summary)
```

	A	B	AB
[1,]	-1.00	-1.000000000	1.00
[2,]	1.00	-1.000000000	-1.00
[3,]	-1.00	1.000000000	-1.00
[4,]	1.00	1.000000000	1.00

```
[5,] -1.00 -1.00000000 1.00  
[6,] 1.00 -1.00000000 -1.00
```

### 3.9.6. Regression Model

To analyze the data using the  $2^k$  model, we can run a regression model or an ANOVA model. We'll do both, starting with regression.

```
Density.vec <- c(t(Density))  
Af <- rep(as.factor(A), rep(1,28))  
Bf <- rep(as.factor(B), rep(1,28))  
options(contrasts = c("contr.sum","contr.poly"))  
Density.lm <- lm(Density.vec ~ Af*Bf)  
summary(Density.lm)
```

Call:

```
lm.default(formula = Density.vec ~ Af * Bf)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.18286	-0.08357	0.02214	0.07286	0.16714

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )		
(Intercept)	0.407143	0.017113	23.791	<2e-16 ***		
Af1	-0.004286	0.017113	-0.250	0.804		
Bf1	-0.007143	0.017113	-0.417	0.680		
Af1:Bf1	-0.012857	0.017113	-0.751	0.460		
---						
Signif. codes:	0'***'	0.001'**'	0.01'*'	0.05'.'	0.1 ' '	1

Residual standard error: 0.09055 on 24 degrees of freedom

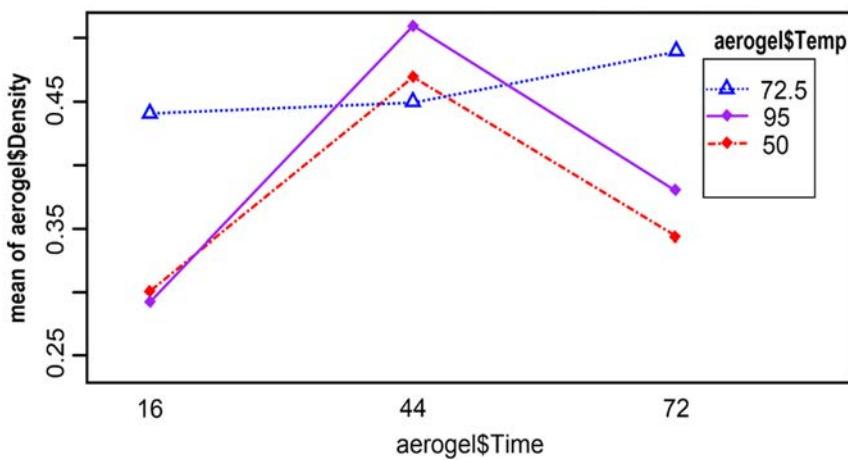
Multiple R-squared: 0.03231, Adjusted R-squared:-0.08865

F-statistic: 0.2671 on 3 and 24 DF, p-value: 0.8484

### 3.9.7. Interaction Plots

**Figure 3-15** provides an **interaction plot** to demonstrate how the factors Time and Temp interact.

```
interaction.plot(aerogel$Time, aerogel$Temp,  
                 aerogel$Density, type = "b", pch = c(18,24,22),  
                 leg.bty = "o")
```



*Figure 3-15. Temperature interactions affecting density*

### 3.9.8. ANOVA Model

Modeling the Aerogel problem using an ANOVA model skips redefining the independent variables as factors. This was implicitly performed when the effect were defined.

```
aero.aov <- data.frame(cbind(A,B,AB,Total))
Density.aov <- aov(Density ~.^2, data = aero.aov)
summary(Density.aov)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
A	1	0.00051	0.00051	8.734e+29	<2e-16 ***
B	1	0.00143	0.00143	2.426e+30	<2e-16 ***
AB	1	0.00463	0.00463	7.861e+30	<2e-16 ***
Total	1	0.19680	0.19680	3.342e+32	<2e-16 ***
A:Total	1	0.00000	0.00000	9.700e-02	0.7589
B:Total	1	0.00000	0.00000	3.828e+00	0.0645 .
AB:Total	1	0.00000	0.00000	2.385e+00	0.1381
Residuals	20	0.00000	0.00000		
---					
Signif.codes:					0'***' 0.001'**' 0.01'*' 0.05'. ' 0.1 ' ' 1

Using the ANOVA model approach results in all the factors being significant and we can conclude that the treatments for aerogels with different epoxy types, treatment duration, a treatment duration are not all the same. To determine what are specifically different, we would need to analyze the results by running contrasts.

### 3.9.9. ANOVA Model (Alternate)

This model demonstrates that the first ANOVA model using the regression model factors (`Af`, `Bf`, and `Af*Bf`) renders the same results as the first ANOVA model (`Density.aov`).

```
aero.aov2 <- data.frame(cbind(Af,Bf,AB,Total))
Density.aov2 <- aov(Density~.^2, data = aero.aov2)
summary(Density.aov2)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Af	1	0.00051	0.00051	7.490e+29	<2e-16 ***
Bf	1	0.00143	0.00143	2.081e+30	<2e-16 ***
AB	1	0.00463	0.00463	6.741e+30	<2e-16 ***
Total	1	0.19680	0.19680	2.866e+32	<2e-16 ***
Af:Total	1	0.00000	0.00000	2.000e-02	0.889
Bf:Total	1	0.00000	0.00000	2.753e+00	0.113
AB:Total	1	0.00000	0.00000	1.035e+00	0.321
Residuals	20	0.00000	0.00000		
---					
Signif. codes:		0'***'	0.001'**'	0.01'*'	0.05'. '
					0.1 ' '
					1

### 3.9.10. Display a Raincloud Plot

`plot_raincloud` creates a raincloud plot to display the distribution of data by a combination of a `boxplot`, a kernel `density plot`, and a `scatterplot`. We plot two rainclouds below as **Figure 3-16** and **Figure 3-17**, which combines an illustration of data distribution (the 'cloud'), with jittered raw data (the 'rain'). This is supplemented by adding boxplots to the left of the distribution plot (or under if you rotate it so that Epoxy Type becomes the vertical axis) or other standard measures of central tendency and error. The first shows the relative locations of the mean temperatures. The second plot shows relative locations of the epoxy type means.

```
library(sdamr)
dat <- data.frame(aerogel)
dat$temp2 <- dat$Temp
set.seed(20201104) # replicate figure with random jitter
plot_raincloud(dat, temp2, groups = Epoxy_type)
```

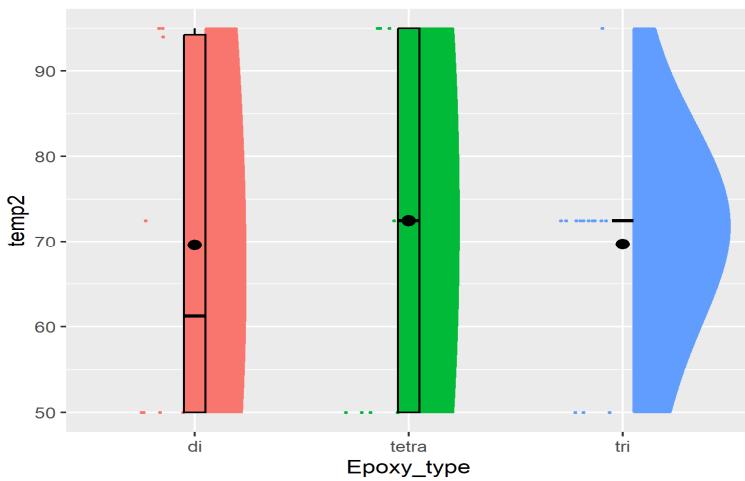


Figure 3-16. Raincloud plots for Epoxy type versus treatment temperature

```
dat2 <- data.frame(aerogel)
dat2$time2 <- dat2$Time
plot_raincloud(dat2, time2, groups = Epoxy_type)
```

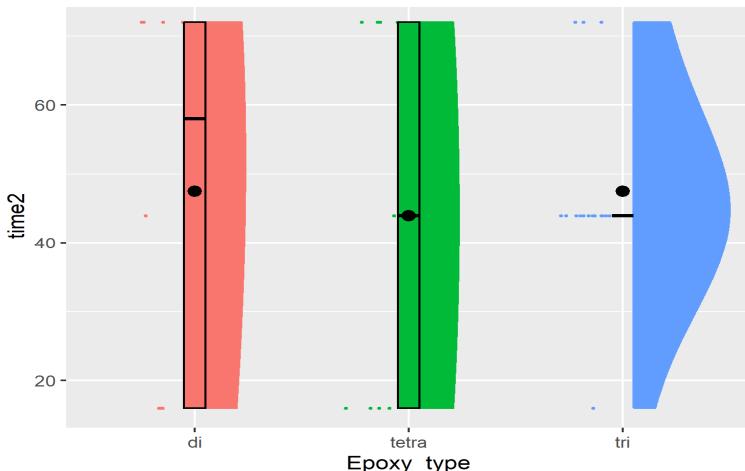


Figure 3-17. Raincloud plots for Epoxy type versus treatment duration

### 3.10. Example of a $2^3$ Factorial

To illustrate the design and analysis of a  $2^3$  factorial experiment, consider the following example. Researchers in a government electronics lab often complained that voltage measurements made on a circuit they constructed were inconsistent. The lab technical director (TD) assistant

decided to conduct an experiment to try to identify the source of the variation. The three factors she varied were  $A$  = the ambient temperature where the voltage measurement was made,  $B$  = the voltmeter warm-up time, and  $C$  = the time the power was connected to the circuit before the measurement was taken. The response was the measured voltage in millivolts. The two levels for factor  $A$  were  $22^{\circ}\text{C}$  (room temperature) and  $32^{\circ}\text{C}$  (close to the temperature in some industrial settings). An oven was used, and the circuit was allowed to stabilize for at least five minutes prior to measurements. The settings for factors  $B$  and  $C$  were  $- = 30$  seconds or less, and  $+ = 5$  minutes. The same circuit was measured for each combination of treatment factors, so the experimental unit was nothing more than the trial or point in time at which the particular combination of treatment factor levels were applied to make the measurement. Two replicates of each of the eight experimental combinations were run in a random order to help prevent biases. The results of the experiment are shown in **Table 3-10**.

**Table 3-10. Factor Settings and Response for Voltmeter Experiment**

Run	Factor Levels			Coded Factors			Rep	Order	y
	A	B	C	$X_A$	$X_B$	$X_C$			
1	22	0.5	0.5	-	-	-	1	5	705
2	32	0.5	0.5	+	-	-	1	14	620
3	22	5	0.5	-	+	-	1	15	700
4	32	5	0.5	+	+	-	1	1	629
5	22	0.5	5	-	-	+	1	8	672
6	32	0.5	5	+	-	+	1	12	668
7	22	5	5	-	+	+	1	10	715
8	32	5	5	+	+	+	1	9	647
1	22	0.5	0.5	-	-	-	1	4	680
2	32	0.5	0.5	+	-	-	1	7	651
3	22	5	0.5	-	+	-	1	2	685
4	32	5	0.5	+	+	-	1	3	635
5	22	0.5	5	-	-	+	1	11	654
6	32	0.5	5	+	-	+	1	16	691
7	22	5	5	-	+	+	1	6	672
8	32	5	5	+	+	+	1	13	673

In this table, the actual factor settings are shown on the left, and the coded  $-$  and  $+$  levels are shown on the right. The actual settings on the

left form a list  $s$  or directions for performing each experiment. The order number on the far right next to the response was created with a random number generator and represents the order in which the experiments should be run. The coded factor levels are used as the independent variables in a regression program in order to calculate the regression coefficients or half effects.

The coded factor levels can be easily calculated from the actual factor settings using the coding and scaling formula. In this formula we subtract the mid-point of the two factor settings, then divide by half the range. For example, for factor  $A$  the mid-point between 22 and 32 is 27, and half the range is 5, thus

$$X_A = \left( \frac{\text{ActualFactorSetting} - 27}{5} \right)$$

The R function `contr.FrF2` performs this coding and scaling on R factors. A data frame `volt` (in the `daewr` package) contains R factors with the actual factor levels and response from **Table 3-10**. The code to open the data frame, code and scale the factors, and fit the regression model with the `lm` function (along with the resulting table of regression coefficients) are shown below. The `contr.FrF2` function labels the coded and scaled linear contrasts  $A_1$ ,  $B_1$ , and  $C_1$  in the output, instead of  $X_A$ ,  $X_B$ , and  $X_C$  as in **Eq. 3-11**.

```
library(daewr)
library(FrF2)
modv <- lm( y ~ A*B*C, data=volt,
contrast = list(A = contr.FrF2,
                 B = contr.FrF2,
                 C = contr.FrF2))
summary(modv)
```

**Call:**  
`lm.default(formula = y ~ A * B * C, data = volt, contrasts =`  
`list(A = contr.FrF2,`  
 `B = contr.FrF2, C = contr.FrF2))`

**Residuals:**

Min	1Q	Median	3Q	Max
-21.5	-11.8	0.0	11.8	21.5

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )	
(Intercept)	668.562	4.518	147.99	4.9e-15	***
A1	-16.813	4.518	-3.72	0.0059	**
B1	0.937	4.518	0.21	0.8408	
C1	5.437	4.518	1.20	0.2632	
A1:B1	-6.687	4.518	-1.48	0.1771	
A1:C1	12.563	4.518	2.78	0.0239	*
B1:C1	1.813	4.518	0.40	0.6988	
A1:B1:C1	-5.813	4.518	-1.29	0.2342	
---					
Signif. codes:	0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1				

Residual standard error: 18.1 on 8 degrees of freedom

Multiple R-squared: 0.772, Adjusted R-squared: 0.572

F-statistic: 3.87 on 7 and 8 DF, p-value: 0.0385

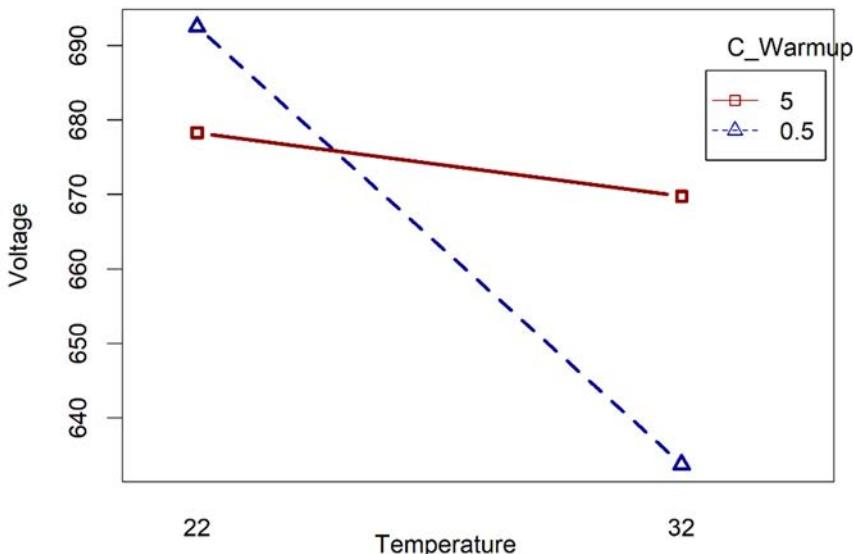
In the output, it can be seen that factor  $A$  (ambient temperature) and the  $A \times C$  interaction, or interaction between the ambient temperature and the circuit warm-up time, are significant. The main effect has direct interpretation. The effect of factor  $A$  is twice the regression coefficient shown above or  $E_A = 2 \times \hat{\beta}_A = 2(-16.8125) = -33.625$ . ambient temperature is increased from  $22^\circ$  to  $32^\circ$ , the voltage measurement will decrease by 33.6 millivolts. However, since the interaction is significant in this example, it really is not meaningful to talk about the average main effect because the effect of ambient temperature depends on the circuit warm-up time.

Describing or interpreting the interaction is best done by looking at the interaction plot shown in **Figure 3-18**. Here it can be seen that when the circuit warm-up time is short (0.5 minutes or 30 seconds) changing the ambient temperature from  $22^\circ$  to  $32^\circ$  causes a large (58.7 millivolt) decrease in the voltage reading. However, when the circuit warm-up time is long (5 minutes), changing the ambient temperature from  $22^\circ$  to  $32^\circ$  only causes a small (8.5 millivolt) decrease in the voltage reading. Therefore, to make voltage readings more consistent, the lab TA recommended that his students allow their circuits to warm up 5 minutes before making voltage measurements.

We performed the regression on the coded factor levels so that the regression coefficients produced by the `lm` function are exactly one-half

of the effects. However, the actual factor names and levels should be used for clarity when presenting the results graphically for inclusion in a report or presentation, as shown in **Figure 3-18**. Most readers or listeners will not remember what the – and + levels represent.

The code to produce **Figure 3-18** is shown below the figure. In this code there was no need to produce a table of predicted values from the reduced model that contains only the significant factors *A* and *AC* (as was done when creating **Figure 3-7**) since the orthogonal design guarantees that the average response in the four *A* by *C* combinations will be the same as the predicted values from the reduced model.

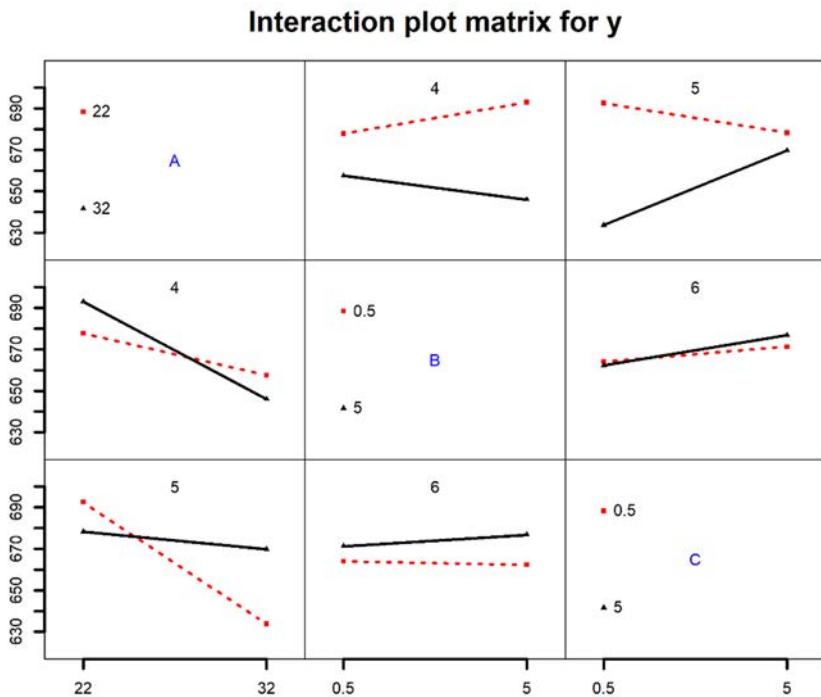


**Figure 3-18. Interaction between Ambient Temperature and Circuit Warm-Up Time**

```
C_Warmup=volt$C
with(volt, (interaction.plot(A, C_Warmup, y,
  type = "b",
  pch = c(24,22), leg.bty = "o",
  xlab = "Temperature", ylab = "Voltage",
  lwd=2.5,col=c("darkblue","red4"))))
```

An even simpler interaction plot, like **Figure 3-19**, can be made quickly using the `IAPlot` function in the `FrF2` package.

```
IAPlot(modv, las = 1)
# interaction plots with attention drawn to aliases
aus <- IAPlot(modv, show.alias=TRUE)
# alias groups corresponding to interaction plots
aliases(modv)$aliases[9:15]
```



**Figure 3-19.** Interaction plots for all factors and levels using *IAPlot*.

The orthogonality of the design also allows a reduced prediction equation to be written from the regression results by simply eliminating the insignificant terms. This equation can be used to predict the voltage reading in millivolts for any ambient temperature between 22° and 32°, and any circuit warm-up time between 30 seconds and 5 minutes.

$$y = 668.563 - 16.813 \left( \frac{\text{Temp} - 27}{5} \right) - 6.688 \left( \frac{\text{CWarm} - 2.75}{2.25} \right) \left( \frac{\text{Temp} - 27}{5} \right)$$

### 3.10.1. Shortcut Formula for Determining the Number of Replicates

Wheeler (1974) has developed a shortcut approximation formula for calculating the **number of runs** necessary to achieve power equal to 0.95 when the significance level for a two-level factorial is  $\alpha=0.05$ . **Eq. 3-12** provides the approximation.

$$N = ((8\sigma)\Delta)^2 \quad \text{Eq. 3-12}$$

where  $\sigma$  is the standard deviation of the experimental error,  $\Delta$  is the practical size of an effect. In this case the difference in average response between the low and high level of a factor and  $N = r \times 2^k$  is the total number of experiments in the  **$2^k$  factorial**. Since this formula is so compact, it is easy to use on the spot in meetings where experiments are being planned. As an illustration of its use, consider an example based on the voltage meter experiment presented in the last section.

The lab instructor felt that the standard deviation of the experimental error was about  $\sigma = 15.0$  and the practical size of an effect was about  $\Delta = 30.0$ .  $\sigma$  would be known by the lab instructors experience in making repeat voltage measurements of the same circuit under exactly the same conditions (i.e., factor levels), and  $\Delta$  would be known from the amount of inconsistency in measurements claimed by the students who were getting inconsistent readings.  $\Delta$  is the size of the effect the TA would like to detect in his experiments. Using the shortcut formula, this says that

$$N = \frac{8 \times 15.0}{(30.0)^2} = 16$$

or that  $r = 2$  replicates of each of the  $2^3 = 8$  runs should result in a power of 0.95 for detecting effects of size 30.0 at significance level  $\alpha = 0.05$ . Using a more exact formula like shown in Section 3.5.2, the actual power for  $r = 2$  replicates is closer to 0.94 than 0.95. However, this approximate formula is accurate enough for most planning purposes.

The simple formula can also be used backwards by solving for  $\Delta$  as a function of  $N$ , that is,  $\Delta = 8 \times \frac{\sigma}{\sqrt{N}}$ . That way, if an experimenter knows his budget for experimentation, which dictates the largest  $N$  can be, he

can calculate the size of the effect  $\Delta$  that he is likely to be able to detect. If the experimenter does not have an accurate estimate of  $\sigma$ , the formula can still be used by talking about practical effect size in units of the unknown  $\sigma$ . For example, if an experimenters' budget allows him to make at most  $N = 64$  experiments, he can hope to detect effects that are no more than one standard deviation of the experimental error, that is,  $\Delta = 8 \times \frac{\sigma}{\sqrt{64}} = \sigma$ . This result will be true regardless of the number of factors in the two-level experiment. Consequently, with 64 runs he may have one factor with  $r = 32$  replicates of each level, or six factors with  $r = 1$  replicate of each of the  $2^6 = 64$  treatment combinations.

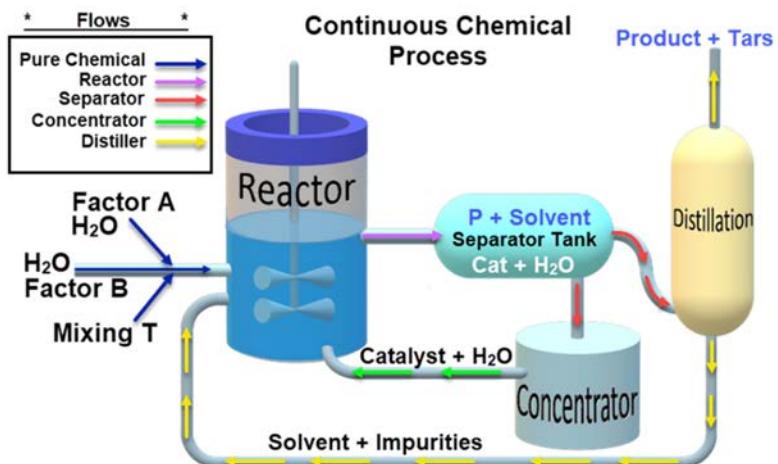
### 3.10.2. Analysis with One Replicate per Cell

Factorial designs with one replicate per cell are often referred to as non-replicated designs. When there is adequate power for detecting effects with  $r = 1$  replication per cell, or treatment combination, there is no need to double the experimental work by replicating each experiment. However, in a non-replicated factorial, the same problem arises that was discussed in [Section 3.5.4](#). There will be zero degrees of freedom for calculating  $ssE$  and thus no  $F$ -tests for the effects. However, when there are multiple factors in a two-level factorial, there are simple graphical tools that allow detection of the significant effects. Since not all main effects and interactions in a  $2^k$  experiment are expected to be significant, the levels of insignificant factors and combinations of levels defined by the insignificant interactions are equivalent to having replicates in the design. Graphical tools allow the significant effects (or equivalently regression coefficients) to be recognized.

The most common graphical tool used to spot significant effects are normal or [half-normal plots](#) that were first suggested by Daniel (1959). These are easy to produce using the [DanielPlot](#) function in the R package [FrF2](#) (Groemping, 2011a) or the [LGB](#) function in the package [daewr](#). Additional graphical tools such as Lenth Plots and Bayes Plots are also useful for detecting significant effects and interactions and can be generated using functions in the [BsMD](#) package (Barrios, 2009). These graphical tools are also available for interactive analysis via the R [DoE](#) plugin (Groemping, 2011b) for the graphical user interface for R called

R Commander (Fox, 2005).

To illustrate the analysis of a non-replicated, two-level factorial, consider an example from the chemical industry. Experimental design principles were developed by Fisher in the early part of the twentieth century and were originally used in agricultural experiments. Within 40 years there was extensive use of experimental design techniques in the chemical industry. **Figure 3-20** is a diagram of a continuous chemical process. In this process continuous streams of two reactants, A and B, are combined at a juncture called the *mixing-T* where they begin to react. The mixture then flows into a reactor and is combined with solvent and a catalyst, and the reaction is completed. The result of the reaction flows into a separator tank where the final product floats to the top in a solvent phase while the catalyst and water go to the bottom of the tank.



**Figure 3-20. Diagram of a Continuous Chemical Process**

The catalyst is concentrated and sent back into the reactor, while the product, byproducts, and solvent are taken to a distillation column where the product is removed, and the solvent is recycled to the reactor.

One of the problems experienced in this process was the production of byproduct (tars). Over time these tars would clog the reactor and force a shutdown of the process for cleaning. It also required an additional

process step to purify the final product. Engineers decided to conduct experiments to see if they could increase the percent conversion which would lower the amount of byproducts. The factors they thought might affect the percent conversion are shown in **Table 3-11** below.

**Table 3-11. Factors thought might affect the percent conversion**

Symbol	Factor Name
A.	Excess of Reactant A (over molar amount)
B.	Catalyst Concentration
C.	Pressure in the Reactor
D.	Temperature of the Coated Mixing-T

Two levels of each factor were chosen that were spread apart as wide as the engineers thought feasible in order to maximize the chance of detecting factor effects with only two levels. During experimentation, the factor levels would be changed after a fixed interval of time. The experimental unit for this would be the particular reactants, catalyst, and solvent entering the reaction zone during a given run, and the response,  $Y$ , would be the percent conversion calculated from the product produced during a run.

The researchers felt that if the percent conversion could be increased by  $\Delta = 12\%$  (or more) it would substantially reduce the maintenance and extra processing required. Thus, the additional yield would be worth detecting. From past experience with the process, the standard deviation in percent conversion on this process for product produced in the same length intervals as the runs in the proposed experimental design (with no changes in the factor levels) was  $\sigma = 6\%$ . Using the shortcut formula, the number of runs required to have a power of 0.95 for detecting factor effects of  $\Delta = 12\%$  or more was

$$N = \left(\frac{8\sigma}{\Delta}\right)^2 = \left(\frac{(8)(6)}{12}\right)^2 = 16$$

Sixteen runs with four factors means the design would be non-replicated. The experimental conditions (in coded units), a list of random run orders, and results of the experiments are shown in

**Table 3-12. List of Experiments and Results for Chemical Process**

Random Run No.	A	B	C	D	Y
15	-	-	-	-	45
13	+	-	-	-	41
11	-	+	-	-	90
1	+	+	-	-	67
10	-	-	+	-	50
2	+	-	+	-	39
3	-	+	+	-	95
12	+	+	+	-	66
16	-	-	-	+	47
8	+	-	-	+	43
9	-	+	-	+	95
14	+	+	-	+	69
6	-	-	+	+	40
5	+	-	+	+	51
7	-	+	+	+	87
4	+	+	+	+	72

The R code to open the data frame and perform a regression analysis to estimate the half effects or regression coefficients are shown below. The data frame chem in the `daewr` package contains numeric vectors of coded and scaled values for the factor levels, and the `contr.FrF2` function was not needed.

```
library(daewr)
data(chem)
modf <- lm( y ~ A * B * C * D, data = chem)
summary(modf)
```

In the results below, unlike the example in **Section 3.7**, there will be no estimate of  $ssE$  and thus no  $t$ -tests on the regression coefficients in the `lm` summary. The regression coefficients for main effects  $A$  and  $B$  along with the  $A \times B$  interaction are the largest effects, but a graph must be used to determine which are significant.

Call:  
`lm.default(formula = y ~ A * B * C * D, data = chem)`

Residuals:

ALL 16 residuals are 0: no residual degrees of freedom!

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	62.3125	NaN	NaN	NaN
A	-6.3125	NaN	NaN	NaN
B	17.8125	NaN	NaN	NaN
C	0.1875	NaN	NaN	NaN
D	0.6875	NaN	NaN	NaN
A:B	-5.3125	NaN	NaN	NaN
A:C	0.8125	NaN	NaN	NaN
B:C	-0.3125	NaN	NaN	NaN
A:D	2.0625	NaN	NaN	NaN
B:D	-0.0625	NaN	NaN	NaN
C:D	-0.6875	NaN	NaN	NaN
A:B:C	-0.1875	NaN	NaN	NaN
A:B:D	-0.6875	NaN	NaN	NaN
A:C:D	2.4375	NaN	NaN	NaN
B:C:D	-0.4375	NaN	NaN	NaN
A:B:C:D	-0.3125	NaN	NaN	NaN

Residual standard error: NaN on 0 degrees of freedom

Multiple R-squared: 1, Adjusted R-squared: NaN

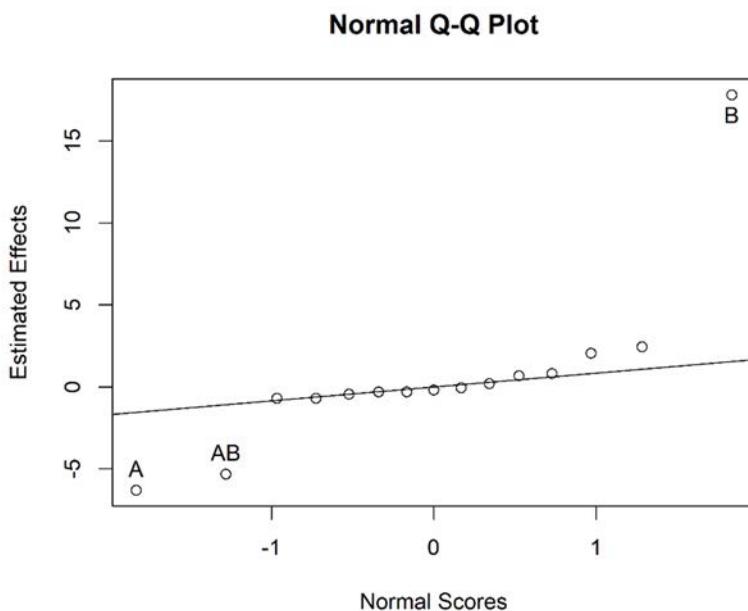
F-statistic: NaN on 15 and 0 DF, p-value: NA

The effects in a two-level factorial are the difference of two averages. If changes in factor levels do not cause a change in the response, the effect will be just the difference in averages of random data (due to random fluctuations in experimental error). If none of the factors or interactions cause changes in the response, the entire set of effects, or regression coefficients, should appear as a sample from the normal distribution with zero mean due to the **Central Limit Theorem**. Therefore if we make a normal probability plot of the effects, the insignificant effects should lie along a straight line and any significant effects or interactions should appear as outliers on the plot.

The R code illustrates how to create a normal plot of the regression coefficients in the object `modf` that was created by the `lm` function. This code calls the function `fullnormal` from the `daewr` package. A similar plot, with the axis reversed, can be made with the `DanielPlot` function in the `FrF2` package.

In this normal plot (**Figure 3-21**), most of the points lie along a straight line drawn through the origin at  $(0, 0)$ . However, the points representing main effects  $A$ ,  $B$ , and the  $A \times B$  interaction tend to be below and to the left or above and to the right of the straight line. This indicates that these three effects are significant. The points along the straight line are insignificant and the slope of this line is an estimate of the standard error of an effect  $\hat{\beta}_\beta$ .

```
library(daewr)
fullnormal(coef(modf)[-1],alpha=.025)
```



**Figure 3-21. Normal Probability Plot of Regression Coefficients**

One way to create a half-normal plot of the coefficients in `modf` is to use the `LGB` function in the package `daewr`. Light Gradient Boosting (LGB) is an ensemble learning framework. By default this function draws a significance limit and produces a report indicating which effects are significant. The option `rpt = FALSE` suppresses the printed report. The code below produces **Figure 3-22**.

```
library(daewr)
LGB( coef(modf)[-1], rpt = FALSE)
```

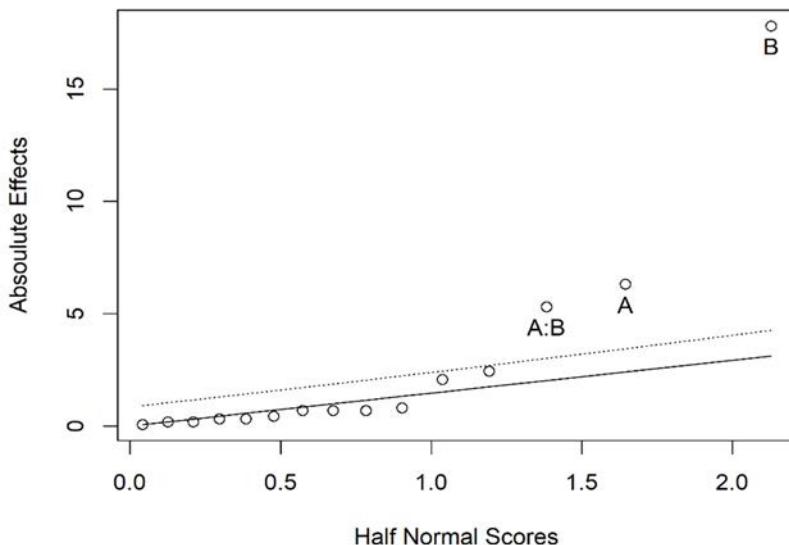


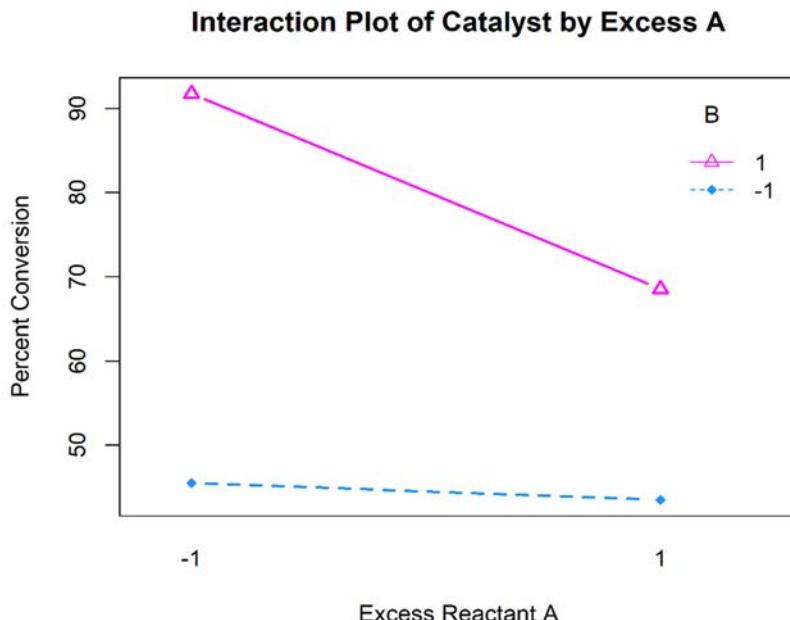
Figure 3-22. Half-Normal Plot of Absolute Regression Coefficients

The half-normal plot shows effects  $A$ ,  $B$ , and the  $A \times B$  interaction fall above the upper prediction limit line and are clearly significant. The reference line and upper limit in **Figure 3-22** were added automatically by the function `LGB` using the method described by Lawson et al. (1998). When drawing the reference line by hand on a plot, it is easier to use the half-normal plot than the normal plot, because the first half to two thirds of the points trending from the lower left will almost always form a straight line. However, on the half-normal plot the signs of the coefficients are lost. For example, in the `lm` summary and the normal plot it can be clearly seen that main effect  $A$  (the excess of reactant  $A$ ) has a negative effect, and that increasing the excess of reactant  $A$  will on the average cause a decrease in the percent conversion. In the half-normal plot, it can be seen that main effect  $A$  is significant but one must refer back to the table of coefficients to see whether it has a positive or negative effect.

Of course, in this example there is a significant interaction, and therefore the main effects cannot be interpreted separately. To assist in interpreting the interaction, an interaction plot should be created. The R code to create the interaction plot between factor  $A$  (excess of reactant  $A$ ) and factor  $B$  (catalyst concentration) is below, and the

resulting graph is shown in **Figure 3-23**.

```
library(daewr)
with(chem, (interaction.plot( A, B, y, type = "b",
    pch = c(18,24), main = "Interaction Plot of
    Catalyst by Excess A", xlab = "Excess Reactant
    A", ylab = "Percent Conversion", lwd = 2,
    col=c("dodgerblue","magenta"))))
```



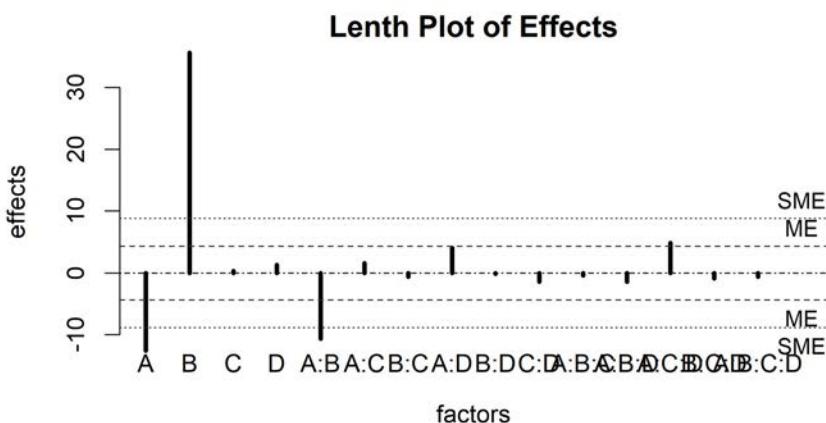
**Figure 3-23. Interaction Plot Catalyst by Excess A**

This figure shows that increasing the level of catalyst increases conversion. Increasing the excess of reactant A has little effect on conversion when a low level of catalyst is used. However, if a high level of catalyst is used, increasing the excess of reactant A decreases conversion by more than 20%. Therefore, to achieve the highest level of conversion, a high level of catalyst and a low level of excess reactant A should be used.

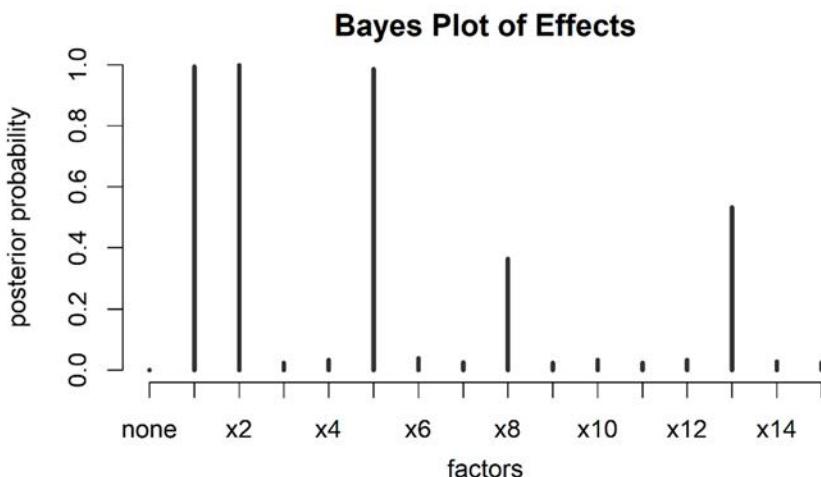
Two other graphical tools that are used to identify significant effects and interactions in non-replicated factorials are the Lenth plot (Lenth, 1989), and the **Bayes plot** (Box & Meyer, 1986b). The **Lenth plot** (**Figure 3-24**) is similar to the analysis of means plot (Ott, 1967) with additional limits

provided. The estimates that fall within the **margin of error (ME)** limits are unlikely to be significant; the estimates that fall within the **simultaneous margin of error (SME)** limits but outside the ME limits are possibly significant, while those that fall outside the SME limits are likely significant. The Bayes plot (*Figure 3-25*) graphically represents the **Bayes posterior probabilities** that the effects are active. The `LenthPlot` and the `BsProb` functions in the package `BsMD` facilitate making these plots. Lenth's plot of effects is called mainly for its side effect. The R code to make these plots with the data in the chemical process experiment is shown below.

```
par( mfrw = c(2,1) )
library(BsMD)
LenthPlot(modf, main = "Lenth Plot of Effects",
          lwd = 3)
X <- model.matrix(modf)[ , 2:16]
y <- chem$y
Chem.BsProb <- BsProb( X = X, y = y, blk = 0,
                      mFac = 15,
                      mInt = 1, p = 0.2, g = 2.49, ng = 1, nMod = 10)
plot( Chem.BsProb, main = "Bayes Plot of Effects" )
```



*Figure 3-24. Lenth of Effects from Chemical Process Experiment*



*Figure 3-25. Bayes Plots of Effects from Chemical Process Experiment*

The result is shown in **Figure 3-25** where it can be seen that effects  $A$ ,  $B$ , and the  $AB$  interaction are identified as likely significant on the Lenth plot. The Bayes plot labels the effect by their Yates order (i.e.,  $x_1 = A$ ,  $x_2 = B$ ,  $x_3 = AB$ ,  $x_4 = C$ ,  $x_5 = AC$ ,  $x_6 = BC$ , ... etc.) rather than by their name. However, the same effects ( $A$ ,  $B$ , and  $AB$ ) are identified as having large posterior probabilities. These results are consistent with what was seen in the normal and half-normal plot of effects.

### 3.10.3. Verifying Assumptions of the Model

When there are replicate experimental units in each cell of a factorial model, or when an interaction term can be assumed negligible and removed from the model (as a result of a preliminary test like those described in **Section 3.5.2** or **3.7.5**), the normality and constant variance assumptions of the factorial model can be verified with residual plots as described in **Section 2.4**.

However, in the case of  $2^k$  design with only one replicate per cell it is a little more difficult to check the assumption of normality. The normality assumption is most often violated by having one outlier or atypical value. The calculated main effects and interactions in a two-level factorial can always be represented as the difference of two averages,  $\bar{y}_+ - \bar{y}_-$ . When experimental errors follow a normal distribution, the calculated effects for factors and interactions that have a negligible influence on

the response should be normally distributed with mean zero. The significance of potential influential factors are judged by their relation to a reference line of points on a normal or half-normal plot of effects formed by the negligible factors. However, one atypical value will bias each calculated effect positively or negatively away from zero. The variability of the calculated effects for the non-influential factors and interactions will be much larger and it will be more difficult to judge significance of effects, much less check the normality assumption of residuals.

Daniel (1960) proposed a manual method for detecting and correcting an outlier or atypical value in a **non-replicated  $2^k$  design**. This method consists of three steps. First, the presence of an outlier is detected by a gap in the center of a normal plot of effects. Second, the outlier is identified by matching the signs of the insignificant effects with the signs of the coded factor levels and interactions of each observation. The third step is to estimate the magnitude of the discrepancy and correct the atypical value.

As an example, consider the normal plot of effects from a non-replicated  $2^4$  experiment described by Box (1991). In this plot it appears that main effects  $B$  and  $C$  may be significant, but there is a vertical gap in the line of insignificant effects that indicates an outlier may be present.

Lawson and Gatlin (2006) automated Daniel's procedure identifying and correcting an atypical value by making two passes through the data. If the **gap statistic** (the ratio of the vertical gap in **Figure 3-24** divided by Lenth's **pseudo standard error (PSE)** statistic) is above the 50th percentile of its reference distribution in the first pass through the data, PSE is recalculated after correcting the outlier and the gap statistic is tested again on a second pass through the data. If the gap statistic is above the 95th percentile in the second pass through the data, the function prints a table showing which observation is the potential outlier, and which effects are significant after correcting the outlier. A half-normal plot is used to identify the significant effects after correcting the outlier.

An R function called **Gapttest** for performing this procedure is available

in the package `daewr`. The function call is shown below, and results are below it and continue onto the next page. The data frame `BoxM` in the call statement contains the data from Box (1991). The first four columns in the data frame are the factors  $A$ ,  $B$ ,  $C$ , and  $D$  in the unreplicated  $2^4$  design and the last column is a numeric response  $y$ .

```
library(daewr)
data(BoxM)
GapTest(BoxM)
```

#### Effect Report

Label	Half Effect	Sig(.05)
A	-0.400	no
B	-2.110	no
C	1.855	no
D	0.505	no
AB	0.455	no
AC	-1.245	no
AD	-0.290	no
BC	-0.400	no
BD	-0.590	no
CD	0.745	no
ABC	0.600	no
ABD	0.360	no
ACD	0.200	no
BCD	-0.790	no
ABCD	0.760	no

Lawson, Grimshaw & Burt Rn Statistic = 1

95th percentile of Rn = 1.201

Initial Outlier Report

Standardized-Gap = 3.3532 Significant at 50th percentile

Final Outlier Report

Response	Corrected Response	Detect Outlier
47.46	47.46	no
49.62	49.62	no
43.13	43.13	no
46.31	46.31	no
51.47	51.47	no
48.49	48.49	no
49.34	49.34	no
46.10	46.10	no
46.76	46.76	no

48.56	48.56	no
44.83	44.83	no
44.45	44.45	no
59.15	52.75	yes
51.33	51.33	no
47.02	47.02	no
47.90	47.90	no

#### Effect Report

Label	Half Effect	Sig(.05)
A	-4.5143e-15	no
B	-1.7100e+00	yes
C	1.4550e+00	yes
D	1.0500e-01	no
AB	5.5000e-02	no
AC	-8.4500e-01	yes
AD	1.1000e-01	no
BC	2.1316e-15	no
BD	-1.9000e-01	no
CD	3.4500e-01	no
ABC	2.0000e-01	no
ABD	-4.0000e-02	no
ACD	6.0000e-01	no
BCD	-3.9000e-01	no
ABCD	3.6000e-01	no

Lawson, Grimshaw & Burt Rn Statistic = 1.6261

95th percentile of Rn = 1.201

The method detected an outlier on the 13th run, and it corrected the response by changing 59.15 to 52.15. Reanalysis of the corrected data shows that main effects *B*, *C*, and the *AC* interaction are significant. The *AC* interaction could not be detected in **Figure 3-26** because the outlier inflated the estimates of the insignificant effects, and the *AC* effect was buried in the noise.

**Figure 3-26** shows the half-normal plot of the coefficients calculated with the corrected data. In this plot it can be seen that the *AC* interaction is clearly significant in addition to the main effects *B* and *C* that were identified in **Figure 3-25**.

Whenever an outlier is discovered, using this method or residual plots, and the conclusions of the analysis change when the outlier is removed

or corrected, the experimenter should proceed cautiously. When there are more than two replicates at the same factor settings where the outlier was found, it may be clear that something is amiss. However, if there are two or less observations at factor settings where the outlier was found, it may be advisable to rerun the questionable experiment.

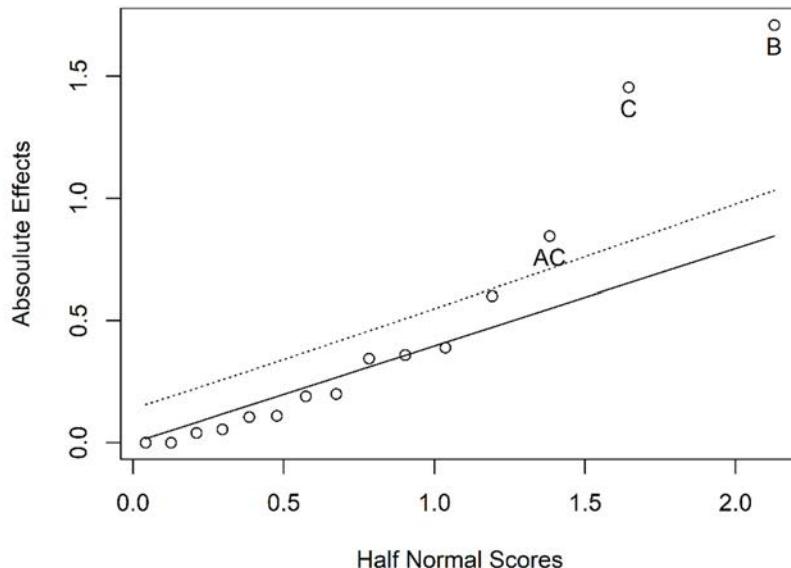


Figure 3-26. Half-Normal Plot of Coefficients Calculated with Corrected Data

### 3.11. Experiment Example – Antireflection Coating (ARC)

Nyberg (1999) has shown that silicon nitride ( $\text{SiN}_x$ ) grown by Plasma Enhanced Chemical Vapor Deposition (PECVD) is a promising candidate for an antireflection coating (ARC) on military grade crystalline silicon solar cells. Silicon nitride was grown on polished (100)-oriented 4A silicon wafers using a parallel plate Plasma Technology PECVD reactor. The diameter of the electrodes of the PECVD is 24 cm and the diameter of the shower head (through which the gases enter) is 2 cm. The RF frequency was 13.56 MHz. The thickness of the silicon nitride was one-quarter of the wavelength of light in the nitride, the wavelength being 640 nm. This wavelength is expected to be close to optimal for silicon solar cell purposes. The process gases were ammonia and a mixture of

3% silane in argon. The experiments were carried out according to a  $2^5$  factorial design. The results are shown in the table below.

### 3.11.1. Factors and Responses

The dataset `pecvd.csv` is comprised of the following variables and responses. we will need to convert the variables to factors for our experimental design.

$A = SiH_4$  to  $NH_3$  Flow Rate Ratio  
 $B =$  Total Gas Flow Rate (sccm)  
 $C =$  Pressure (mtorr)  
 $D =$  Temperature (Co)  
 $E =$  Power (W)  
 $y_1 =$  Refracttion Index  
 $y_2 =$  Growth Rate ("nm" /"min" )

### 3.11.2. Data Prep

#### *Load required packages*

Here we load most of the required packages needed for this experiment:

```
library(daewr)
library(gmodels)
library(car)
library(lsmeans)
library(FrF2)
library(AlgDesign)
library(agricolae)
library(DoE.base)
```

#### *Load the Data*

Next, we load the data using the `read.csv` function and inspect its structure. I am loading it from my local drive but the data file is also in a GitHub repository.

```
pecvd <-
read.csv("https://raw.githubusercontent.com/stricje1/Dat
a/master/pecvd.csv")
structure(pecvd)
```

	No.	A	B	C	D	E	y1	y2
1	1	0.1	40	300	300	10	1.92	1.79
2	2	0.9	40	300	300	10	3.06	10.10
3	3	0.1	220	300	300	10	1.96	3.02
4	4	0.9	220	300	300	10	3.33	15.00
5	5	0.1	40	1200	300	10	1.87	19.70
6	6	0.9	40	1200	300	10	2.62	11.20
7	7	0.1	220	1200	300	10	1.97	35.70
8	8	0.9	220	1200	300	10	2.96	36.20
9	9	0.1	40	300	460	10	1.94	2.31
10	10	0.9	40	300	460	10	3.53	5.58
11	11	0.1	220	300	460	10	2.06	2.75
12	12	0.9	220	300	460	10	3.75	14.50
							.	.
25	25	0.1	40	300	460	60	1.97	5.27
26	26	0.9	40	300	460	60	3.67	12.30
27	27	0.1	220	300	460	60	2.09	6.39
28	28	0.9	220	300	460	60	3.73	30.50
29	29	0.1	40	1200	460	60	1.98	30.10
30	30	0.9	40	1200	460	60	2.99	14.50
31	31	0.1	220	1200	460	60	2.19	50.30
32	32	0.9	220	1200	460	60	3.39	47.10

### 3.11.3. Part 1: Model fitting

Using the data in `pecvd.csv`, we fit the factorial model to the response  $y_1$  including all interactions up to the 5-way.

#### **Solution 1A: Create Factors and Response Dataframes**

We start by loading the necessary R packages: `daewr`.

Now, we extract the variables from the dataframe and using the `as.factor` function, we convert each into factor ( $A \dots B$ ). If we inspect the `pecvd_fac` dataframe, there appears to be no change. We use the `str` function to inspect the structure of `pecvd_fac`.

```
pecvd_fac <- pecvd[2:6]
pecvd_fac$A <- as.factor(pecvd_fac$A)
pecvd_fac$B <- as.factor(pecvd_fac$B)
pecvd_fac$C <- as.factor(pecvd_fac$C)
pecvd_fac$D <- as.factor(pecvd_fac$D)
pecvd_fac$E <- as.factor(pecvd_fac$E)

str(pecvd_fac)
```

```
'data.frame': 32 obs. of 5 variables:
 $ A: Factor w/ 2 levels "0.1","0.9": 1 2 1 2 1 2 1 2 1 2 ...
 $ B: Factor w/ 2 levels "40","220": 1 1 2 2 1 1 2 2 1 1 ...
 $ C: Factor w/ 2 levels "300","1200": 1 1 1 1 2 2 2 2 1 1 ...
 $ D: Factor w/ 2 levels "300","460": 1 1 1 1 1 1 1 1 2 2 ...
 $ E: Factor w/ 2 levels "10","60": 1 1 1 1 1 1 1 1 1 1 ...
```

### **Solution 1B: Merge Response and Factors**

Once we are satisfied that we have the right factors structure, we add the response that we are interested in,  $y_1$ , to the dataframe, for in a new dataframe with five factors and a response.

```
pecvd_df <- data.frame(cbind(pecvd_fac, pecvd$y1))
head(pecvd_df)
```

	A	B	C	D	E	pecvd.y1
1	0.1	40	300	300	10	1.92
2	0.9	40	300	300	10	3.06
3	0.1	220	300	300	10	1.96
4	0.9	220	300	300	10	3.33
5	0.1	40	1200	300	10	1.87
6	0.9	40	1200	300	10	2.62

### **Solution 1C: Fit a Model**

#### **Fit a Linear Model**

Finally, we fit a linear model including all interactions up to the 5-way. The  $R$  for this model is one, indicating a perfect fit

```
fit <- lm(pecvd.y1 ~ .^5, data = pecvd_df)
summary(fit)
```

Call:

`lm.default(formula = pecvd.y1 ~ .^5, data = pecvd_df)`

Residuals:

ALL 32 residuals are 0: no residual degrees of freedom!

Coefficients:

Estimate Std. Error t value

`Pr(>|t|)`

(Intercept)

1.920e+00

NaN

NaN

NaN

A0.9

1.140e+00

NaN

NaN

NaN

B220

4.000e-02

NaN

NaN

NaN

C1200

-5.000e-02

NaN

NaN

NaN

D460	2.000e-02	NaN	NaN	NaN
E60	3.000e-02	NaN	NaN	NaN
A0.9:B220	2.300e-01	NaN	NaN	NaN
A0.9:C1200	-3.900e-01	NaN	NaN	NaN
A0.9:D460	4.500e-01	NaN	NaN	NaN
A0.9:E60	7.000e-02	NaN	NaN	NaN
B220:C1200	6.000e-02	NaN	NaN	NaN
B220:D460	8.000e-02	NaN	NaN	NaN
B220:E60	2.000e-02	NaN	NaN	NaN
C1200:D460	7.000e-02	NaN	NaN	NaN
C1200:E60	-2.000e-02	NaN	NaN	NaN
D460:E60	2.749e-15	NaN	NaN	NaN
A0.9:B220:C1200	1.000e-02	NaN	NaN	NaN
A0.9:B220:D460	-1.300e-01	NaN	NaN	NaN
A0.9:B220:E60	-2.000e-02	NaN	NaN	NaN
A0.9:C1200:D460	-2.000e-02	NaN	NaN	NaN
A0.9:C1200:E60	-5.600e-01	NaN	NaN	NaN
A0.9:D460:E60	4.000e-02	NaN	NaN	NaN
B220:C1200:D460	1.000e-02	NaN	NaN	NaN
B220:C1200:E60	-2.000e-02	NaN	NaN	NaN
B220:D460:E60	-2.000e-02	NaN	NaN	NaN
C1200:D460:E60	1.000e-02	NaN	NaN	NaN
A0.9:B220:C1200:D460	-1.000e-02	NaN	NaN	NaN
A0.9:B220:C1200:E60	3.500e-01	NaN	NaN	NaN
A0.9:B220:D460:E60	-1.400e-01	NaN	NaN	NaN
A0.9:C1200:D460:E60	2.800e-01	NaN	NaN	NaN
B220:C1200:D460:E60	4.000e-02	NaN	NaN	NaN
A0.9:B220:C1200:D460:E60	-1.000e-01	NaN	NaN	NaN

Residual standard error: NaN on 0 degrees of freedom  
 Multiple R-squared: 1, Adjusted R-squared: NaN  
 F-statistic: NaN on 31 and 0 DF, p-value: NA

## Fit an ANOVA Model

Now we'll fit an ANOVA model.

```
aov.fit <- aov(pecvd.y1 ~ ., data = pecvd_df)
summary(aov.fit)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
A	1	11.592	11.592	237.104	1.39e-14 ***
B	1	0.374	0.374	7.652	0.010299 *
C	1	0.525	0.525	10.745	0.002968 **
D	1	0.756	0.756	15.472	0.000556 ***
E	1	0.002	0.002	0.050	0.824618
Residuals	26	1.271	0.049		

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

### Solution 1D: Examine Model Contrasts with Tukey

```
TukeyHSD(aov.fit, c("A", "B"))
```

```
Tukey multiple comparisons of means
95% family-wise confidence level

Fit: aov.default(formula = pecvd.y1 ~ ., data = pecvd_df)
$A
  diff      lwr      upr   p adj
0.9-0.1 1.20375 1.043059 1.364441    0
$B
  diff      lwr      upr   p adj
220-40 0.21625 0.05555941 0.3769406 0.0102988
```

### Solution 1E: Generate Cell Means Table

Now we'll generate model cell means.

```
model.tables( aov.fit, type = "means" )$tables
```

```
$`Grand mean`
[1] 2.594375
$A
  A           B
  0.1       0.9
  1.99250  3.19625
  2.48625  2.70250
$B
  C           D
  300       1200
  2.72250  2.46625
  300       460
  2.440625 2.748125
$C
  E
  10        60
  2.603125 2.585625
```

### 3.11.4. Part 2. Make a Half-Normal Plot

In **Figure 3-27**, we used the R `auditor` package to make a half-normal plot of the effects of regression coefficients to determine which main effects and interactions are significant.

```
library(auditor)
model_glm <- glm(pecvd.y1 ~ ., family = gaussian(),
                  data = pecvd_df)
lm_audit <- audit(model_glm, data = pecvd_df,
                   y = pecvd_df$pecvd.y1)
```

```
Preparation of a new explainer is initiated
  -> model label      : lm ( default )
  -> data              : 32 rows 6 cols
  -> target variable   : 32 values
  -> predict function  : yhat.glm will be used ( default )
  -> predicted values  : No value for predict function target
  -> model_info        : package stats , ver. 4.3.3
A new explainer has been created!
```

```
hn_lm <- model_halfnormal(lm_audit)
plot_halfnormal(hn_lm)
```

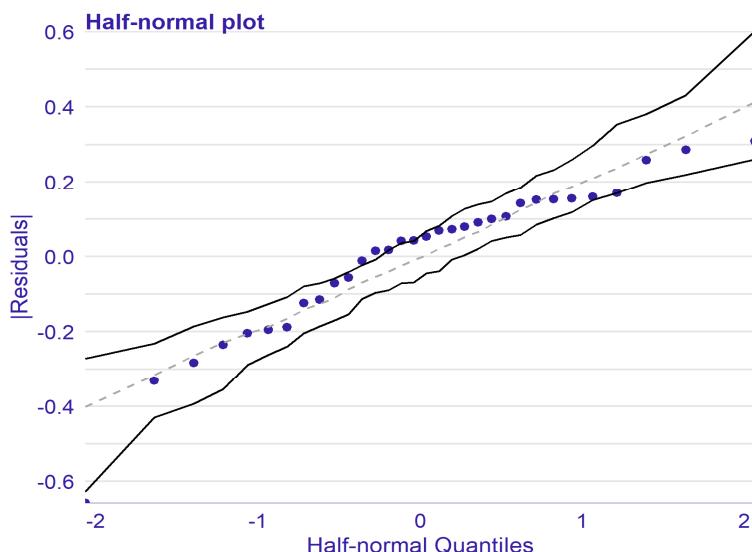
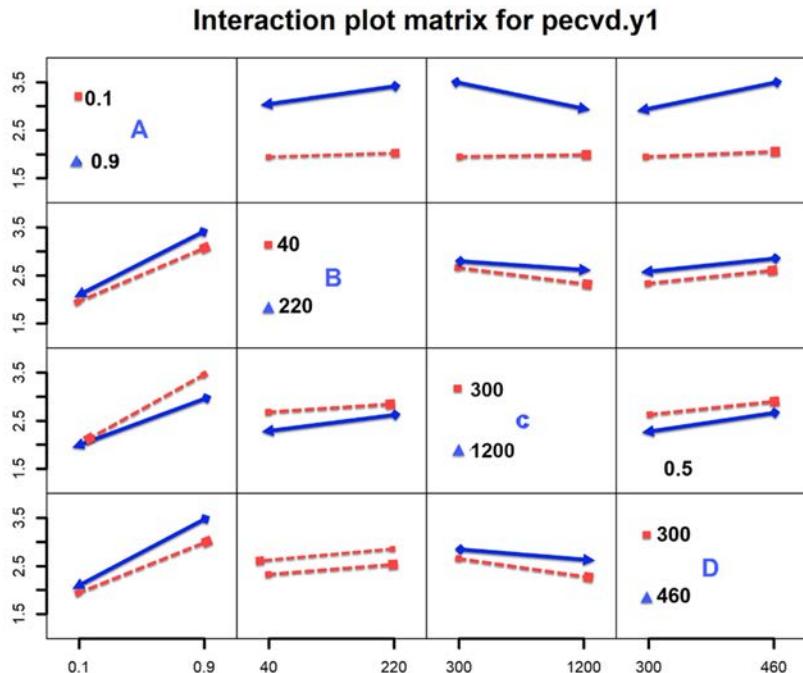


Figure 3-27. Half-normal plot or half-normal quantiles vs. residuals

## Solution 2A: Interaction Plot Matrix

Next, we generate main effects plots and interaction plots in **Figure 3-28**. Interactions appear between factors *A* and *B*, *A* and *C*, and *A* and *D*. These interactions will soon be validated.

```
library(FrF2)
IAPplot(fit, sel = c(1,2,3,4), abbrev = 7, lwd = 2)
```



**Figure 3-28** Interaction plots for the initial model “fit”.

## Solution 2B: Daniel Plots

*Daniel plots* in **Figure 3-29** are normal plot of effects from a two-level factorial experiment. The Normal Plot of Effects seem to validate the model assumptions.

```
model_glm <- glm(pecvd.y1 ~ ., family = gaussian(),
                    data = pecvd_df)
par(mfrow=c(1,3), oma = c(0,0,1,0), pty = "s")
DanielPlot(model_glm, half = TRUE,
```

```

    main = "Half-Normal Plot")
DanielPlot(model_glm, main = "Normal Plot of Effects")
DanielPlot(model_glm, faclab = list(idx = c(12,4,13),
    lab = c("A","B","C","D","E")),
    main = "Active Contrasts")

```

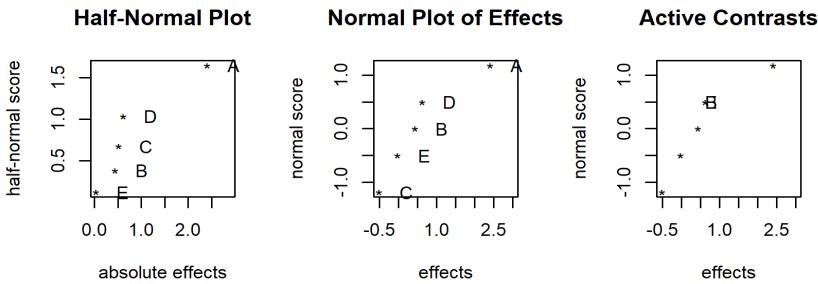


Figure 3-29. DanielPlots : Half-normal, Normal, and Active Contrasts

### Solution 2C: Sample of Interaction Plots

Now we display some model interaction plots as alternatives to **Figure 3-28**.

```

with(pecvd_df, (interaction.plot(A,C, pecvd.y1, type =
"b", pch = c(0,2,6,15), leg.bty = "o", main =
"Interaction for SiH4 to NH3 Flow Rate Ratio by
Pressure",
xlab = "Pressure",
ylab = "average Refract. Index", lwd = 2,
lty = 1:4, col=c("dodgerblue","magenta"))))

```

## Interaction for SiH<sub>4</sub> to NH<sub>3</sub> Flow Rate Ratio by Press

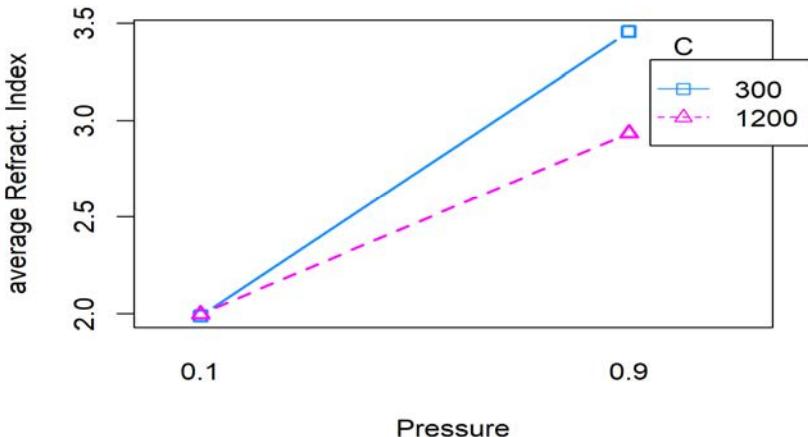


Figure 3-30. Interaction between the SiH<sub>4</sub> to NH<sub>3</sub> Flow Rate Ratio and Pressure relative to the Average Refraction Index of the coating.

```
with(pecvd_df, (interaction.plot(A,E, pecvd.y1,
type = "b", pch = c(0,2,6,15), leg.bty = "o",
main = "Interaction for SiH4 to NH3 Flow Rate Ratio
by Temperature", xlab = "Temperature",
ylab = "average Refract. Index", lwd = 2,
lty = 1:4, col=c("dodgerblue","orange2"))))
```

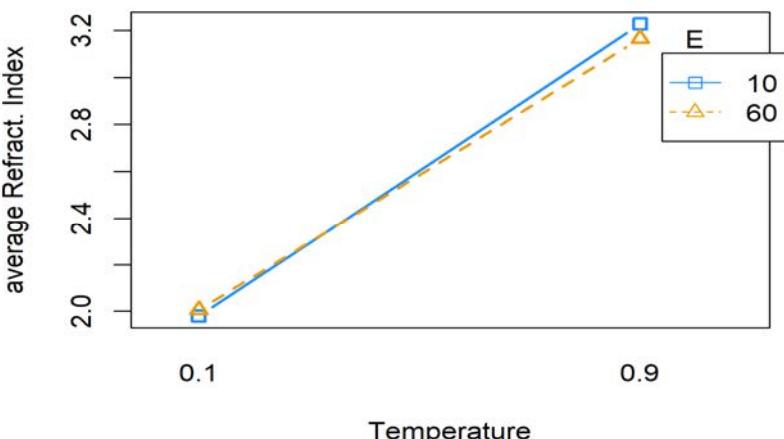


Figure 3-31. Interaction between the SiH<sub>4</sub> to NH<sub>3</sub> Flow Rate Ratio and Temperature relative to the Average Refraction Index of the coating.

### 3.11.5. Problem 3. Drop insignificant terms and Remodel

Next, we drop the insignificant terms from the previous model(s) and fit a new model. We'll also construct residual plots to check the assumptions of the model fit.

#### **Solution 3A: Model with First and Second Order Teams Only**

Let's look examine an ANOVA model and a **Generalized Linear Model (GLM)**. Then we'll use the most significant factors at  $\alpha \leq 0.01$ .

```
model_glm2 <- glm(pecvd.y1 ~ ., family = gaussian(),
                    data = pecvd_df)
model_glm3 <- glm(pecvd.y1 ~ .^2, family = gaussian(),
                    data = pecvd_df)
model_aov2 <- aov(pecvd.y1 ~ ., data = pecvd_df)
model_aov3 <- aov(pecvd.y1 ~ .^2, data = pecvd_df)
print(summary(model_glm2))
```

Call:

```
glm(formula = pecvd.y1 ~ ., family = gaussian(),
     data=pecvd_df)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	1.86750	0.09574	19.505	< 2e-16 ***
A0.9	1.20375	0.07817	15.398	1.39e-14 ***
B220	0.21625	0.07817	2.766	0.010299 *
C1200	-0.25625	0.07817	-3.278	0.002968 **
D460	0.30750	0.07817	3.933	0.000556 ***
E60	-0.01750	0.07817	-0.224	0.824618
---				

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

(Dispersion parameter for gaussian family taken to be  
0.048890)

Null deviance: 14.5216 on 31 degrees of freedom  
Residual deviance: 1.2712 on 26 degrees of freedom  
AIC: 1.586

Number of Fisher Scoring iterations: 2

```
print(summary(model_glm3))
```

```

Call:
glm(formula = pecvd.y1~.^2, family = gaussian(),
data=pecvd_df)

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  1.91125   0.06466 29.557 2.17e-15 ***
A0.9         1.20875   0.07229 16.720 1.49e-11 ***
B220        0.04125   0.07229  0.571  0.57621  
C1200       -0.05375   0.07229 -0.743  0.46797  
D460        0.02375   0.07229  0.329  0.74678  
E60         0.05875   0.07229  0.813  0.42835  
A0.9:B220   0.19750   0.06466  3.054  0.00757 ** 
A0.9:C1200  -0.53250   0.06466 -8.235 3.80e-07 *** 
A0.9:D460   0.41500   0.06466  6.418 8.51e-06 *** 
A0.9:E60    -0.09000   0.06466 -1.392  0.18302  
B220:C1200  0.14250   0.06466  2.204  0.04254 *  
B220:D460   -0.03000   0.06466 -0.464  0.64894  
B220:E60    0.04000   0.06466  0.619  0.54489  
C1200:D460  0.13500   0.06466  2.088  0.05317 .  
C1200:E60   -0.15000   0.06466 -2.320  0.03390 *  
D460:E60    0.04750   0.06466  0.735  0.47322  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 0.00836)

```

Null deviance: 14.5216 on 31 degrees of freedom  
Residual deviance: 0.1338 on 16 degrees of freedom  
AIC: -50.457

Number of Fisher Scoring iterations: 2

From the linear model the factor *A* at level 0.9, as well as the interactions between *A* at 0.9 and *B* at 200, *A* at 0.9 and *C* at 1200, and *A* at 0.9 and *D* at 460.

```
print(summary(model_aov2))
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
<i>A</i>	1	11.592	11.592	237.104	1.39e-14 ***
<i>B</i>	1	0.374	0.374	7.652	0.010299 *
<i>C</i>	1	0.525	0.525	10.745	0.002968 **
<i>D</i>	1	0.756	0.756	15.472	0.000556 ***
<i>E</i>	1	0.002	0.002	0.050	0.824618
Residuals	26	1.271	0.049		

```

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

print(summary(model_aov3))

   Df Sum Sq Mean Sq F value    Pr(>F)
A       1 11.592 11.592 1386.202 < 2e-16 ***
B       1  0.374  0.374  44.737 5.21e-06 ***
C       1  0.525  0.525  62.818 6.26e-07 ***
D       1  0.756  0.756  90.457 5.49e-08 ***
E       1  0.002  0.002  0.293  0.59578
A:B     1  0.078  0.078  9.329  0.00757 **
A:C     1  0.567  0.567  67.816 3.80e-07 ***
A:D     1  0.344  0.344  41.190 8.51e-06 ***
A:E     1  0.016  0.016  1.937  0.18302
B:C     1  0.041  0.041  4.857  0.04254 *
B:D     1  0.002  0.002  0.215  0.64894
B:E     1  0.003  0.003  0.383  0.54489
C:D     1  0.036  0.036  4.359  0.05317 .
C:E     1  0.045  0.045  5.381  0.03390 *
D:E     1  0.005  0.005  0.540  0.47322
Residuals 16  0.134  0.008
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

The ANOVA model validates the results of the linear model with the significant factors being  $A$ ,  $B$ ,  $C$ , and  $D$ , along with the second order interactions between  $A$  and  $B$ ,  $A$  and  $C$ , and  $A$  and  $D$ .

### **Solution 3B: Final Model**

Here we use the ANOVA model with the most significant factors and interactions alone.

```

model_aov4 <- aov(pecvd.y1 ~ A+B+C+D+A:B+A:C+A:D,
                     data = pecvd_df)
print(summary(model_aov4))

```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
A	1	11.592	11.592	979.529	< 2e-16 ***
B	1	0.374	0.374	31.612	8.68e-06 ***
C	1	0.525	0.525	44.389	6.84e-07 ***
D	1	0.756	0.756	63.920	3.19e-08 ***
A:B	1	0.078	0.078	6.592	0.0169 *
A:C	1	0.567	0.567	47.921	3.70e-07 ***
A:D	1	0.344	0.344	29.106	1.53e-05 ***

```
Residuals   24  0.284   0.012
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

### Solution 3C: Final Model Diagnostics

Using the R function `audit` from the `auditor` package, we see that the MSE is 0.00888, which is a good metric.

```
aov_audit <- audit(model_aov4, data = pecvd_df,
y = pecvd_df$pecvd.y1)
```

```
Preparation of a new explainer is initiated
  -> model label      : lm ( default )
  -> data             : 32 rows 6 cols
  -> target variable  : 32 values
  -> predict function : yhat.lm will be used ( default )
  -> predicted values : No value for predict function target
column. ( default )
  -> model_info       : package stats , ver. 4.3.3 , task
regression ( default )
  -> predicted values : numerical, min =  1.87875 , mean =
2.594375 , max =  3.8725
  -> residual function: difference between y and yhat (
default )
  -> residuals        : numerical, min = -0.38 , mean =
-5.759282e-16 , max =  0.125
A new explainer has been created!
```

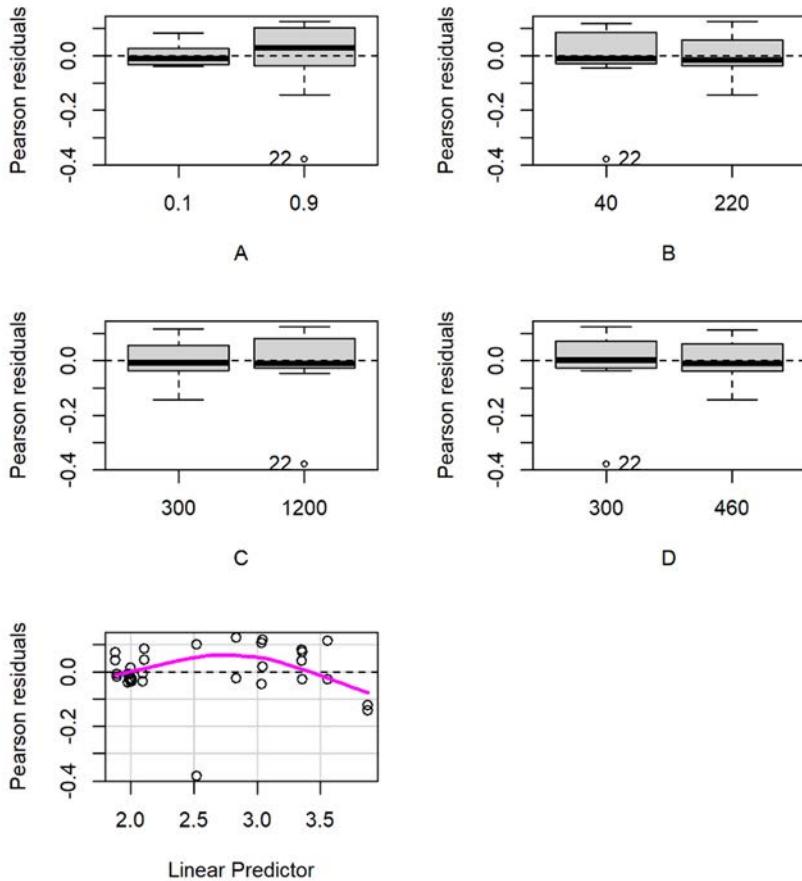
```
score_mse(aov_audit)
```

```
mse: 0.008875781
```

### Residual Plots:

The residual plots provide evidence for validating the model assumptions although there is evidence of nonlinearity present in the linear prediction chart.

```
residualPlots(model_glm)
```



*Figure 3-32. Pearson residual plots for the model main effects*

#### Contrasts using Tukey Output

The Tukey tests show significant differences between the main factors, A, B, C and D.

```
TukeyHSD(model_aov4, c("A", "B", "C", "D"))
```

```
Tukey multiple comparisons of means
95% family-wise confidence level
```

```
Fit: aov.default(formula = pecvd.y1 ~ A + B + C + D + A:B +
A:C + A:D, data = pecvd_df)
```

```
$A
```

	diff	lwr	upr	p adj
0.9-0.1	1.20375	1.124369	1.283131	0
\$B	diff	lwr	upr	p adj
220-40	0.21625	0.1368691	0.2956309	8.7e-06
\$C	diff	lwr	upr	p adj
1200-300	-0.25625	-0.3356309	-0.1768691	7e-07
\$D	diff	lwr	upr	p adj
460-300	0.3075	0.2281191	0.3868809	0

### Autocorrelation Plot using Auditor

The residuals do not show a linear pattern

```
mr_lm <- model_residual(lm_audit)
plot_autocorrelation(mr_lm)
```

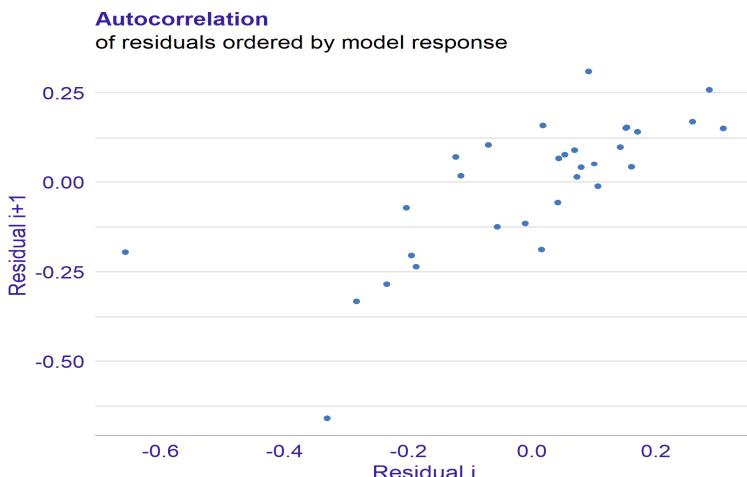


Figure 3-33. ANOVA Model autocorrelation plot of ordered residuals

```
plot_autocorrelation(mr_lm, smooth = TRUE)
```

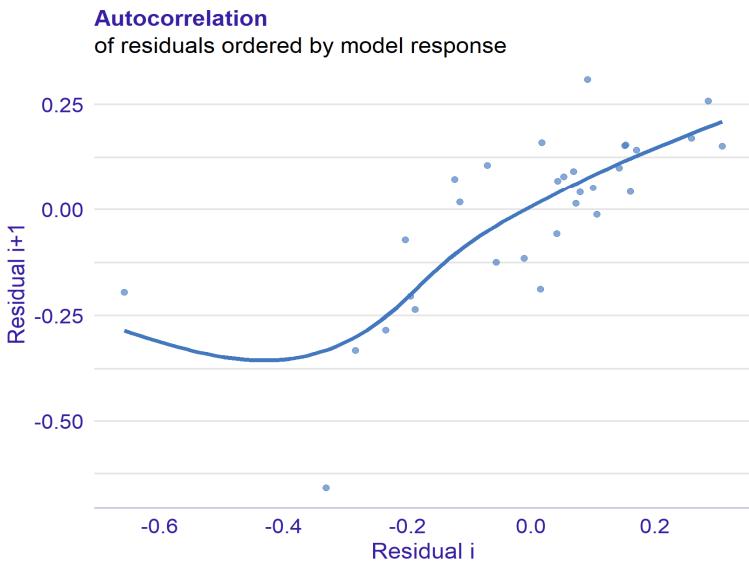


Figure 3-34. ANOVA Model autocorrelation plot of residuals with fitted curve

### Daniel Plots

```
par(mfrow = c(1,3), oma = c(0,0,1,0), pty = "s")
DanielPlot(model_glm, half = TRUE,
main = "Half-Normal Plot")
DanielPlot(model_glm, main = "Normal Plot of Effects")
DanielPlot(model_glm,
faclab = list(idx = c(12,4,13),
lab = c("A","B","C","D","E")),
main = "Active Contrasts")
```

### More Outcomes

Now we'll generate model cell means.

```
model.tables( aov.fit, type = "means" )$tables
```

```
$`Grand mean`
[1] 2.594375

$A
A
 0.1      0.9
1.99250 3.19625
```

```

$B
B
    40      220
2.48625 2.70250

$C
C
    300     1200
2.72250 2.46625

$D
D
    300      460
2.440625 2.748125

$E
E
    10       60
2.603125 2.585625

```

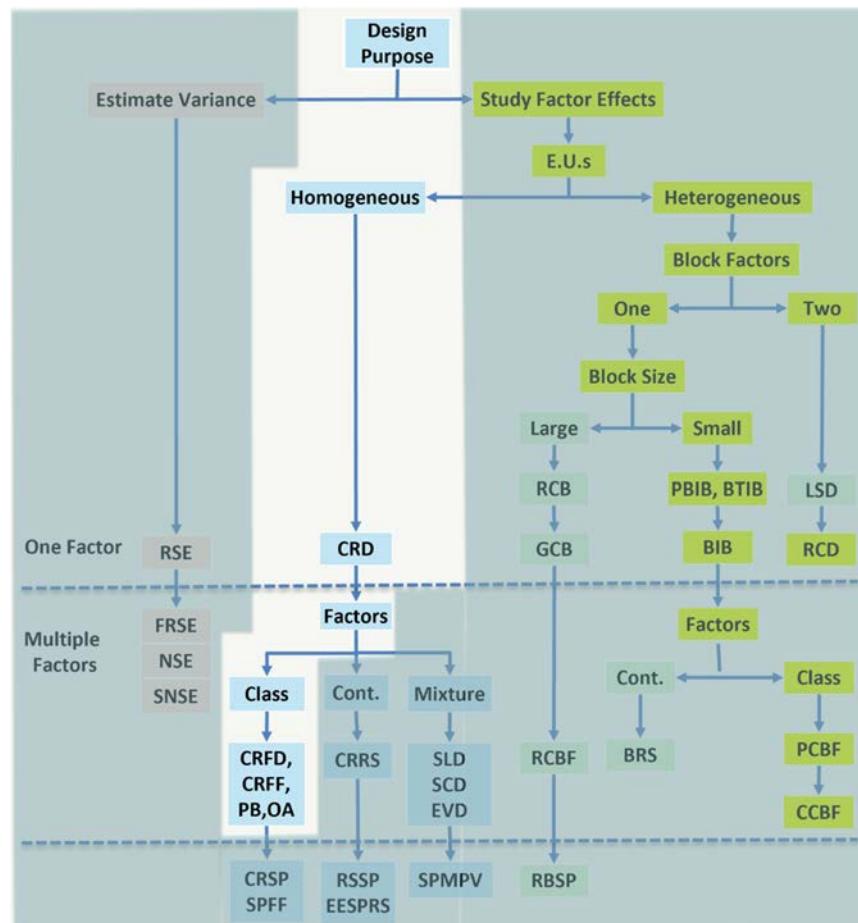
### *3.12. Review of Important Concepts*

When experimenting with more than one treatment factor, factorial designs with all possible treatment combinations randomized to experimental units are much more efficient than separate one factor designs. When there are homogeneous experimental units and multiple factors under study, the CRFD or completely randomized factorial design should be used.

**Figure 3-35** shows when CRD and CRFD should be used. If there is one factor under study with homogeneous experimental units, use a CRD. However, if there are multiple experimental units under study, use the CRFD. In a factorial design the total number of replicates of each level of one factor is the number of replicates per cell multiplied by the product of the number of levels of all other factors in the design. This hidden replication increases the power for detecting factorial effects or reduces the number of experiments needed to obtain the same power as a series of separate one-factor designs. The CRFD should be used when the experimental units are homogeneous, and it is reasonable to run all combinations of levels of the factors.

By studying more than one treatment factor simultaneously in a factorial

design, interaction or joint effects of the factors can be detected. Interactions occur when the effect of one factor is different depending on the level of another factor or on a combination of levels of other factors. Interactions are common in the real world and ignoring them by experimenting with one factor at a time can be very misleading. Examples in this chapter show that it is easy to describe or interpret the meaning of an interaction by comparing or contrasting the effect of one factor over the levels of another factor. Interaction graphs are helpful in describing interactions.



**Figure 3-35. Design Selection Roadmap**

The model for analysis of a factorial with two factors can be written as

$$y_{ijk} = \mu + \alpha_i + \beta_j + \alpha\beta_{ij} + \varepsilon_{ijk}$$

and can be easily extended to multiple factors. The assumptions for the analysis are homogeneity and normality of experimental errors and can be checked using the residual plots described in Chapter 2.

When there is an equal number of replications per cell or treatment combination, the  $F$ -tests in the ANOVA table produced by the R `aov` function or the `anova` summary of a `lm` object will be correct, and the marginal means and cell means produced by the R `model.tables` function will be unbiased estimates that will be useful in revealing the direction of factor effects and interactions. However, if there is an unequal number of replicates per cell, these  $F$ -tests and marginal means may be misleading. In that case, type II or type III sums of squares should be used in the ANOVA table. These are produced by the `Anova` function in the `car` package by selecting the desired option. In addition, the least squares or `lsmeans` produced by the `lsmeans` function in the `lsmeans` package will produce unbiased estimates of cell means and factor level means.

When multiple factors are studied in a factorial experiment, often only two levels of each factor are used in order to reduce the total amount of experimentation required. In that case, the model simplifies to a regression model

$$\begin{aligned} y = & \beta_0 + \beta_A X_A + \beta_B X_B + \beta_{AB} X_A X_B + \beta_C X_C + \beta_{AC} X_A X_C \\ & + \beta_{BC} X_B X_C + \beta_{ABC} X_A X_B X_C + \varepsilon \end{aligned}$$

and the regression coefficients are exactly one half the effect or difference in the average response between the low and high levels of the factors. Interaction effects are also easily defined as differences of averages in this case. There is a shortcut formula for approximating the power in two-level factorials, and when there is only one replicate per treatment combination, the significance of effects and interactions can be determined using normal or half-normal plots.

## 4. Randomized Block Designs

---

### 4.1. Introduction

In order to eliminate as much of the natural variation as possible and increase the sensitivity of experiments, it would be advisable to choose the experimental units for a study to be as homogeneous as possible. In mathematical terms this would reduce the variance,  $\sigma^2$ , of the experimental error and increase the power for detecting treatment factor effects. On the other hand, most experimenters would like the conclusions of their work to have wide applicability. Consider the following example. An experimenter would like to compare several methods of aerobic exercise to see how they affect the stress and anxiety level of experimental subjects. Since there is wide variability in stress and anxiety levels in the general population, as measured by standardized test scores, it would be difficult to see any difference among various methods of exercise unless the subjects recruited to the study were a homogeneous group each similar in their level of stress. However, the experimenter would like to make general conclusions from his study to people of all stress levels in the general population.

Blocking can be used in this situation to achieve both objectives. Blocking is the second technique that falls in the category of error control defined in **Section 1.6**. In a randomized block design (RBD), a group of heterogeneous experimental units is used so that the conclusions can be more general; however, these heterogeneous experimental units are grouped into homogeneous sub-groups before they are randomly assigned to treatment factor levels. The act of grouping the experimental units together in homogeneous groups is called **blocking**. Randomly assigning treatment factor levels to experimental units within the smaller homogeneous subgroups of experimental units, or blocks, has the same effect as using only homogeneous units, yet it allows the conclusions to be generalized to the entire class of heterogeneous experimental units used in the study.

If experimental units represent physical entities, blocking or grouping by similar physical characteristics often results in more homogeneous groups. For example, plots of land in agricultural experiments are usually

blocked by proximity because plots in close proximity normally have similar soil characteristics. When experimental units are animals, the grouping of genetically similar animals, such as littermates, often reduces variability within groups. When experimental units are simply trials, or points in time where treatments will be applied, they are often blocked by time since many lurking variables may change over time and trials in close temporal proximity are more alike.

In a **randomized complete block design**, or **RCBD**, with one treatment factor, when the factor has  $t$  levels there will be  $b$  blocks (or subgroups of homogeneous experimental units) that each contain exactly  $t$  experimental units for a total of  $t \times b$  experimental units. The  $t$  experimental units within each block are as similar as possible, and the groups of experimental units vary enough from block to block to allow general conclusions to be drawn. The randomization of experimental units to treatment factor levels, described in Chapter 2, is performed within each block. Fisher first proposed block designs for agricultural experiments where the experimental units were plots of ground. Blocks represented compact plots in close proximity which were similar. Variability among blocks represented the range of conditions over which conclusions would be generalized.

If there are more than  $t$  experimental units within each block, so that treatment factor levels are replicated  $r$  times within each block, there will be a total of  $r \times t \times b$  experimental units. In this situation we call the design a **general complete block design**. Normally the RCB would be preferred over the general complete block design because smaller blocks of experimental units allow for greater homogeneity within the blocks and thus smaller experimental error and more precise tests and estimates.

## 4.2. Creating an RCB in R

The randomization of experimental units to treatment factor levels in a randomized block design can be accomplished using the base R code or using functions from user written packages. To illustrate how to do this using base R code, consider the following experimental situation. A student wanted to investigate old wives' tales of methods for extending

the life of cut flowers. The treatment factor was the liquid to fill the vase. The levels were:

1. Tap water
2. Tap water with one spoonful of sugar added
3. Tap water with one cup of carbonated water
4. Tap water with one cup of 7-up

The experimental units were single flowers and the response was the time in days until the flower wilted. The student wanted the conclusions of his study to apply to many types of flowers, so she used an RCB design. The blocks were:

1. Rose
2. Carnation
3. Daisy
4. Tulip

In the R code below, a vector  $f$ , of factor levels is created to represent the four treatments. Next, the sample function is used to create a random ordering of the treatment levels. This is repeated four times because a different random ordering is required for each of the four blocks or types of flowers. Finally, these vectors are stacked together, merged with the block indicators in a data frame, and written to a .csv file that can be used as an electronic data collection form like the example in **Section 2.2.1**.

```
f <- factor( c(1,2,3,4) )
b1t <- sample(f,4)
b2t <- sample(f,4)
b3t <- sample(f,4)
b4t <- sample(f,4)
t <- c(b1t, b2t, b3t, b4t)
block <- factor( rep(c("carnation", "daisy", "rose",
    "tulip"), each = 4))
flnum <- rep(f,4)
plan<-data.frame(TypeFlower = block,
    FlowerNumber = flnum, treatment = t)
write.table(plan, file = "RCBPlan.csv",
    sep = ",", row.names = FALSE)
```

This code will produce a different randomized list each time it is run. To use the list it produces, the student would number the flowers in each block from 1 to 4. Then using the list, the carnation number 1 would be placed in a vase with the treatment factor level that is indicated on the first line of the data collection form. Carnation number 2 would be placed in a vase with the treatment level that is indicated on the second line of the data collection form, and so forth.

RCB designs can also be easily created using user written packages. The code below shows an example. In this case, the function `design.rcbd` from the `agricolae` package (de Mendiburu, 2012) is illustrated. By default this function labels the experimental units as “plots,” and uses integers for the block numbers. The next statement renames the levels of the blocks. The `seed` argument in the function call is for the randomization, and running the code with the same seed will allow the user to reproduce the same randomized list multiple times.

```
library(agricolae)
treat <- c(1, 2, 3, 4)
outdesign <- design.rcbd(treat, 4, seed = 11)
rcb <- outdesign$book
levels(rcb$block) <- c("carnation", "daisy", "rose",
"tulip")
```

### 4.3. Model for RCB

The model for the analysis of an RCBD is

$$y_{ij} = \mu + b_i + \tau_j + \varepsilon_{ij} \quad \text{Eq. 4-1}$$

where  $b_i$  represent the block effects,  $\tau_j$  represent the treatment effects. The usual assumptions of normality of experimental error and homogeneity of variance of experimental error across levels of the treatment factors and blocks are required for this model. These assumptions can be checked with residual plots as shown in **Section 2.4**.

Notice that this is an additive model which does not include the interaction between block and treatment. Since there are only  $t \times b$  experimental units, there would be zero degrees of freedom for the error term  $ssE$  if a block by treatment interaction term were included in

the model. However, the block by treatment interaction is in fact the correct error term for testing the treatment effects. The experimenter wants to generalize his conclusions about treatment effects over all the experimental units, so the average treatment effects should be larger than any differences in treatment effects among blocks of experimental units. The difference in treatment effects among blocks is exactly what the interaction measures and is therefore the correct error term. By leaving the interaction out of the model, the  $ssE$  becomes identical to the interaction sums of squares.

The ANOVA table for an RCB design is shown symbolically in **Table 4-1**. Representations for the type I sums of squares for blocks and treatments are shown in the table, similar to what was shown in **Section 3.5.1**, but they will be identical to the type III sums of squares for this design. The error sums of squares is  $ssE = y'y - \hat{\beta}'X'y = y'(I - X(X'X)^{-}X')y$ , where  $\hat{\beta} = (X'X)^{-}X'y$ .

**Table 4-1. Analysis of a Variance Table**

Source	df	Sum of Squares	Mean Squares	F-ratio
Blocks	$b - 1$	$ssBlk$ $R(b \mu)$	$ssBlk$ $\frac{R(b \mu)}{b - 1}$	
Treatments	$t - 1$	$ssT$ $R(\tau b, \mu)$	$ssT$ $\frac{R(\tau b, \mu)}{t - 1}$	$\frac{msT}{msE}$
Error	$(b - 1)(t - 1)$	$ssE$	$ssE$ $\frac{ssE}{(b - 1)(t - 1)}$	

The degrees of freedom for the error  $(b - 1)(t - 1)$  is smaller than it would be in a completely randomized design with  $b$  replicates of each treatment level; however, if the groups of experimental units within the blocks are more homogeneous, the  $msE$  should be smaller and the power for detecting differences among the treatment levels higher.

The estimate of the variance of the homogeneous experimental units within each block is given by

$$\hat{\sigma}_{crb}^2 = \frac{ssE}{(b - 1)(t - 1)} \quad \text{Eq. 4-2}$$

An estimate of the variance of the entire group of heterogenous experimental units can be made from the mean squares in the RCB ANOVA. It is given by the formula

$$\hat{\sigma}_{crb}^2 = \frac{ssBlk + ssE}{t(b - 1)} \quad \text{Eq. 4-3}$$

which is a weighted average of the mean square for blocks and the mean square for error. However, the weights are not simply the degrees of freedom for each mean square. If the  $msBlk$  is zero, it can be seen that  $\hat{\sigma}_{crb}^2 < \hat{\sigma}_{rcb}^2$ . The ratio of  $\hat{\sigma}_{crb}^2$  and  $\hat{\sigma}_{rcb}^2$  is a measure of the efficacy of blocking.

The error degrees of freedom for the RCB is  $v_{rcb} = (b - 1)(t - 1)$ , and the error degrees of freedom for a **completely randomized design (CRD)** with the same number of experimental units would be  $v_{crd} = t(b - 1)$ . The relative efficiency of the RCB is then given by the formula:

$$RE = \frac{(v_{rcb} + 1)(v_{crd} + 3)\hat{\sigma}_{crb}^2}{(v_{rcb} + 3)(v_{crd} + 1)\hat{\sigma}_{rcb}^2} \quad \text{Eq. 4-4}$$

RE can be used to determine the number of observations that would be required in a CRD, with heterogeneous experimental units, in order to have the variances for the treatment means equivalent to that achieved with the RCB. If  $b \times t$  experimental units were used in the RCB design, then  $RE \times (b \times t)$  experimental units would be required in a CRD design, without blocking, to attain equivalent variances of treatment means.

#### **4.4. Example of an RCB – Advanced Combat Helmet**

The Advanced Combat Helmet (ACH) replaces the Personnel Armor System, Ground Troops (PASGT) Helmet for general use by the U.S. Army. The ACH consists of a finished ballistic protective shell, pad suspension system, a retention system (chin strap/neck strap), cover, and eyewear retention strap. The Acquisition Decision Memorandum (ADM) for the ACH was approved on January 8, 2003, with the goal of modifying the PASGT Helmet outer shell geometry, while at the same time applying material technology advancements to reduce the weight and maintain protection against fragmentation of 9mm small arms

munitions.

#### **4.4.1. ACH RTP Performance Requirements**

The ACH is designed to provide ballistic protection or **resistance to penetration (RTP)** from fragments as well as 9mm projectiles. The helmet shell, including any hardware exposed on the outside of the shell, is designed to be resistant to a 9mm Full Metal Jacketed Round Nose (FMJ RN) Remington bullet penetration with a nominal mass of 124 grains. The 9mm FMJ RN ballistic test is an industry standard adopted from the National Institute of Justice Standard, “Ballistic Resistance of Body Armor.” It was also derived from the Operational Requirements Document (ORD) for the Land Warrior, which outlines the requirements and operational capability needs of an integrated soldier protection and equipment system. The ORD states that the greatest threat to the land warrior is fragmentation and the second greatest threat is bullets. For the ACH, the 9mm FMJ RN test does not only represent a capability to be resistant against bullets but also larger sized fragments.

In response to a January 2009 DoD OIG Report, “DoD Testing Requirements for Body Armor,” the Director Operational Test and Evaluation (DOT&E) published the hard body armor **First Article Test (FAT)** protocol on April 27, 2010, and the **Lot Acceptance Testing (LAT)** protocol on July 2, 2010. On December 7, 2010, DOT&E published the Military Combat Helmet Standard for Ballistic Testing FAT applicable to all DoD combat helmet acquisition programs, including the ACH. The DOT&E FAT protocol states that combat helmets must meet a set of statistically based FAT to qualify a design/manufacturing process for full-rate production. The FAT protocol was updated on September 2, 2011, to include both aramid-based helmets, such as the ACH and Lightweight Helmet (LWH), and ultra-high molecular polyethylene-based helmets, such as the ECH, designed to defeat threats more lethal than a 9mm FMJ RN.

#### **4.4.2. FAT RTP Requirements**

The FAT RTP legacy requirements used by ACH bridge contracts were established in the Contract Purchase Description October 30, 2007. It requires a sample size of four helmets tested under different

environmental conditions and shot five times. Under the contract, one penetration of the total 20 shots will result in a failed FAT. The sample size and number of allowable penetrations are not derived from a known statistical model.

In contrast to the legacy FAT RTP requirement, the DOT&E FAT RTP requirement is statistically based. The FAT RTP requirement increases the sample size to 48 helmets at 5 shots per helmet, totaling 240 shots based on an established “90/90 standard.”<sup>4</sup> An increase in sample size increases the statistical confidence of the testing result.

The Clopper-Pearson method is used to calculate the allowable number of penetrations out of a given sample size. When applied against the required parameters (90/90 and 240 shots), the Clopper-Pearson method yields 17 allowable penetrations. **Table 4-2** shows the standard

**Table 4-2. Comparison of FAT RTP Protocols**

9 mm RTP shell	Ambient 70° F	Hot 160° F	Cold -60° F	Seawater, then test at 70° F	Totals	Penetrations Accept/Reject
DOT&E test protocol sample size	60 shots 12 helmets	60 shots 12 helmets	60 shots 12 helmets	60 shots 12 helmets	240 shots 48 helmets	17/18

The sampling plan to determine the accept/reject criteria is based on the American National Standards Institute (ANSI) American Society for Quality (ASQ) Z1.4-2008 with a special inspection level S-3. These translate to the sample size and accept/reject criteria in Table 4.

**Table 4-3. DOT&E LAT RTP**

9 mm RTP shell	Lot Size	Sample Size	Accept	Reject
	91-150	25 shots 5 helmets	0	1
	151-500	40 shots 8 helmet	1	2
	501-1,200	65 shots 13 helmets	1	2
	1,200+	65 shots 13 helmets	1	2

In this case, the helmet is represented by the term  $b_i$  in the model  $y_{ij} = \mu + b_i + \tau_j + \varepsilon_{ij}$ . The experimental error, represented by  $\varepsilon_{ij}$ , is the effect of the state of helmet  $i$  during the run when it received shot  $j$ . If

data were presented without describing the experimental unit and the randomization process, the model could be easily misspecified as  $y_{ij} = \mu + \tau_i + \varepsilon_{ij}$  resulting in the wrong analysis and conclusions.

```
helm <- read.csv("https://raw.githubusercontent.com/
  stricje1/Data/master/helmets.csv" )
helmx <- helm[1:4]
helmx$cond = as.factor(helmx$cond)
helmx$block = as.factor(helmx$size)
helmx$side = as.factor(helmx$side)
helmx$helmet = as.factor(helmx$helmet)
head(helmx)
```

	size	cond	side	helmet	block
1	S	Ambient	L	1	S
2	S	Ambient	R	2	S
3	S	Ambient	F	3	S
4	S	Ambient	B	4	S
5	S	Hot	L	1	S
6	S	Hot	R	2	S

We used the R function `lm` to produce the linear model with second order interactions, since we can expect interaction with the environmental conditions and the parts of the ACH. After eliminating several insignificant factors, we settled on the following model.

```
helm3 <- lm(y ~ block + cond + side + side:cond,
  data = helm_df)
summary(helm3)
```

Call:

```
lm.default(formula = y ~ block + cond + side + side:cond, data
= helm_df)
```

Residuals:

Min	1Q	Median	3Q	Max
-4.8750	-0.6875	0.0000	0.8750	4.1875

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	3.750e-01	1.085e+00	0.346	0.731188
blockM	-2.500e-01	7.039e-01	-0.355	0.724129
blockS	6.250e-02	7.039e-01	0.089	0.929643

```

blockXL           6.875e-01  7.039e-01  0.977  0.333944
condCold          3.750e+00  1.408e+00  2.664  0.010687 *
condHot           5.000e-01  1.408e+00  0.355  0.724129
condSeawater      1.300e+01  1.408e+00  9.234  5.94e-12 ***
sideF             2.000e+00  1.408e+00  1.421  0.162314
sideL             5.000e-01  1.408e+00  0.355  0.724129
sideR             2.987e-15  1.408e+00  0.000  1.000000
condCold:sideF   -5.500e+00  1.991e+00 -2.762  0.008278 **
condHot:sideF   -2.000e+00  1.991e+00 -1.005  0.320487
condSeawater:sideF -8.250e+00  1.991e+00 -4.144  0.000149 ***
condCold:sideL   -3.000e+00  1.991e+00 -1.507  0.138846
condHot:sideL   -7.500e-01  1.991e+00 -0.377  0.708164
condSeawater:sideL -8.750e+00  1.991e+00 -4.395  6.68e-05 ***
condCold:sideR   -3.250e+00  1.991e+00 -1.632  0.109577
condHot:sideR   -2.500e-01  1.991e+00 -0.126  0.900633
condSeawater:sideR -9.250e+00  1.991e+00 -4.646  2.96e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

```

Residual standard error: 1.991 on 45 degrees of freedom

Multiple R-squared: 0.8046, Adjusted R-squared: 0.7265

F-statistic: 10.3 on 18 and 45 DF, p-value: 1.442e-10

In the resulting ANOVA table above, the F-tests show that there is a significant difference in treatment factor levels, especially the Seawater condition.

To interpret the differences in treatment factor levels, we should perform a comparisons of means. Here we will use orthogonal polynomial contrasts as described in **Section 2.8** are useful. We can use the R function `contr.poly` to calculate the linear, quadratic, cubic, and quartic contrasts for dose. The code is shown below. There the split option in the `summary.aov` is used rather than the `summary.lm` function that was used in **Section 2.8**, since we only need to see the single degree of freedom partition for the dose factor in the model.

```

contrasts(helm_df$cond) <- contr.poly(4)
helm3 <- aov(y ~ block + cond + side + side:cond,
data = helm_df)

summary.aov(helm3,split = list(cond = list(
  "Linear" = 1,
  "Quadratic" = 2,

```

```
"Cubic" = 3,
"Quartic" = 4)))
```

The results show that there is a significant linear and quadratic trend in condition and its interaction with side over the number of penetrations in the ACH.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
block	3	7.6	2.54	0.641	0.592505
cond	3	478.6	159.54	40.249	8.51e-13 ***
cond: Linear	1	266.4	266.45	67.219	1.77e-10 ***
cond: Quadratic	1	138.1	138.06	34.830	4.37e-07 ***
cond: Cubic	1	74.1	74.11	18.697	8.39e-05 ***
cond: Quartic	1				
side	3	92.6	30.87	7.789	0.000271 ***
cond:side	9	155.7	17.31	4.366	0.000392 ***
cond:side: Linear	3	83.0	27.67	6.982	0.000590 ***
cond:side: Quadratic	3	25.6	8.52	2.150	0.107207
cond:side: Cubic	3	47.2	15.72	3.966	0.013646 *
cond:side: Quartic	0	0.0			
Residuals	45	178.4	3.96		
---					
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '					

```
TukeyHSD( helm.aov, "cond" )
```

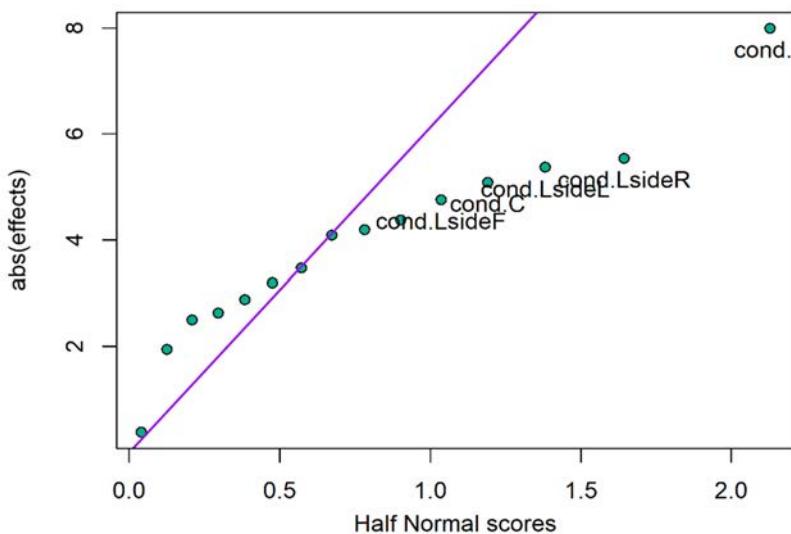
```
Tukey multiple comparisons of means
95% family-wise confidence level
```

```
Fit: aov.default(formula = y ~ .^2, data = helm_df)
```

\$cond	diff	lwr	upr	p	adj
Cold-Ambient	0.8125	-1.098882	2.723882	0.6544884	
Hot-Ambient	-0.2500	-2.161382	1.661382	0.9839501	
Seawater-Ambient	6.4375	4.526118	8.348882	0.0000000	
Hot-Cold	-1.0625	-2.973882	0.848882	0.4392754	
Seawater-Cold	5.6250	3.713618	7.536382	0.0000001	
Seawater-Hot	6.6875	4.776118	8.598882	0.0000000	

Because there were no replicates in the experiment, we used a half-normal plot (see **Figure 4-1**) of the effects to determine the significant effects and interactions.

```
halfnorm(effects, names(effects), alpha = .5)
```



*Figure 4-1. Half-normal plot for ACH test conditions and helmet sides*

#### 4.5. Example of an RCB - ADHD Research

Consider the data in **Table 4-4** from Lim and Wolfe (1997), partially modified from Heffner et al. (1974). The effect of the drug d-amphetamine sulfate on the behavior of rats was the object of the experiment. It is used to treat attention-deficit hyperactivity disorder (**ADHD**) and **narcolepsy** (uncontrollable desire for sleep or sudden attacks of deep sleep).

The behavior under study was the rate at which water-deprived rats pressed a lever to obtain water. The response was the lever press rate defined as the number of lever presses divided by the elapsed time of the session. The treatment factor levels were five different dosages of the drug in milligrams per kilogram of body weight, including a control dosage consisting of saline solution. An experiment, or run, consisted of injecting a rat with a drug dosage, and after one hour an experimental session began where a rat would receive water each time after a second lever was pressed. The experimental unit in these experiments was not a rat, but the state of a single rat during one experiment or run, since an individual rat could be used in many experiments by repeatedly injecting it with different doses of the drug (after an appropriate washout period) and by observing the lever pressing behavior. Because there was wide

variability in the lever pressing rate between rats, an RCB design was used, and a rat represented the blocking factor. Each rat received all five doses in a random order with an appropriate washout period in between.

In this case, the rat is represented by the term  $b_i$  in the model  $y_{ij} = \mu + b_i + \tau_j + \varepsilon_{ij}$ . The experimental error, represented by  $\varepsilon_{ij}$ , is the effect of the state of rat  $i$  during the run when it received dose  $j$ . If data were presented without describing the experimental unit and the randomization process, the model could be easily misspecified as  $y_{ij} = \mu + \tau_i + \varepsilon_{ij}$  resulting in the wrong analysis and conclusions.

**Table 4-4. Rat Behavior Experiment**

Rat	0.0	0.5	1.0	1.5	2.0
1	0.60	0.80	0.82	0.81	0.50
2	0.51	0.61	0.79	0.78	0.77
3	0.62	0.82	0.83	0.80	0.52
4	0.60	0.95	0.91	0.95	0.70
5	0.92	0.82	1.04	1.13	1.03
6	0.63	0.93	1.02	0.96	0.63
7	0.84	0.74	0.98	0.98	1.00
8	0.96	1.24	1.27	1.20	1.06
9	1.01	1.23	1.30	1.25	1.24
10	0.95	1.20	1.18	1.23	1.05

To utilize the R function `aov` to produce the ANOVA using the correct model, the following commands are used.

```
mod1 <- aov( rate ~ rat + dose, data = drug )
summary(mod1)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)			
rat	9	1.668	0.1854	22.2	3.7e-12 ***			
dose	4	0.460	0.1151	13.8	6.5e-07 ***			
Residuals	36	0.301	0.0083					
---								
Signif. codes:	0	'***'	0.001	'**'	0.01'*'	0.05'.'	0.1 ' '	1

In the resulting ANOVA table above, the  $F$ -tests show that there is a significant difference in treatment factor levels.

To interpret the differences in treatment factor levels, comparisons of

means should be made. Since the factor levels are quantitative, orthogonal polynomial contrasts as described in *Section 2.8* are useful. The R function `contr.poly` that was introduced in *Section 2.8* can be used to calculate the linear, quadratic, cubic, and quartic contrasts for dose. The code is shown below. There the split option in the `summary.aov` is used rather than the `summary.lm` function that was used in *Section 2.8*, since we only need to see the single degree of freedom partition for the dose factor in the model.

```
contrasts(drug$dose) <- contr.poly(5)
mod2 <- aov( rate ~ rat + dose, data = drug)

summary.aov(mod2, split = list(dose = list(
  "Linear" = 1,
  "Quadratic" = 2,
  "Cubic" = 3,
  "Quartic" = 4)))
```

The results show that there is a significant linear and quadratic trend in lever press rate over the dose of the drug.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
rat	9	1.668	0.185	22.21	3.7e-12 ***
dose	4	0.460	0.115	13.78	6.5e-07 ***
dose: Linear	1	0.061	0.061	7.31	0.01 *
dose: Quadratic	1	0.394	0.394	47.23	4.8e-08 ***
dose: Cubic	1	0.004	0.004	0.49	0.49
dose: Quartic	1	0.001	0.001	0.09	0.76
Residuals	36	0.301	0.008		
---					
Signif. codes:	0	'***'	0.001	'**'	0.01 '*'
			0.05	'.'	0.1 ' '
					1

The significant linear and quadratic trends over the dose range can be visualized by plotting the means as a function of dose. The R code below produces the graph in *Figure 4-2*. The quadratic trend line on the plot fits the means well and shows that the lever press rate increases as a function of dose until it reaches a maximum somewhere between 1.0 and 1.5 milligrams per kilogram of body weight.

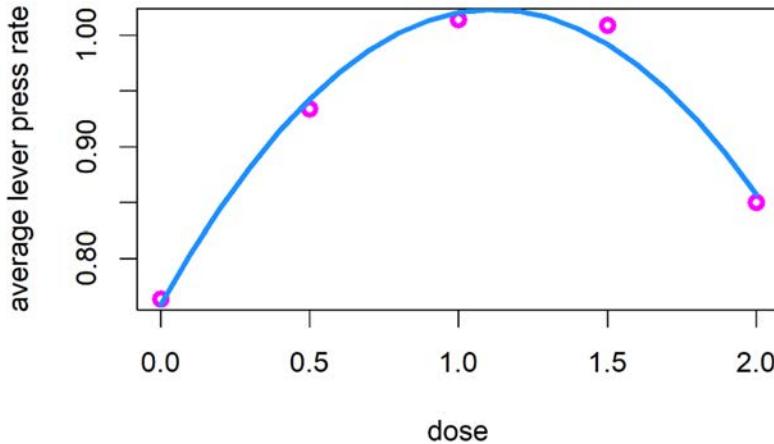
```
R <- do.call("cbind", split(covid$rate,
covid$test_case))
y <- apply(R, 1, mean )
```

```

x <- as.double( levels(covid$dose) )
plot( x, y, xlab = "dose",
      ylab = "average lever press rate" )
xx <- seq( 0.0, 2.0, .1 )
rate.quad <- lm( y ~ poly( x, 2 ) )
lines(xx, predict( rate.quad, data.frame( x = xx) ))

```

The estimated variance of the experimental units (trials) within a block (or rat) is the mean square error  $\hat{\sigma}_{rcb}^2 = 0.00834867$ . The variance of the heterogeneous experimental units is given by



**Figure 4-2. Lever Press Rate Means as Function of Drug Dose**

$$\hat{\sigma}_{crb}^2 = \frac{ssBlk + ssE}{t(b-1)} = \frac{(1.6685 + 0.3006)}{((5)(10-1))} = 5.2413 \quad \text{Eq. 4-5}$$

This is approximately five times larger than the variance within a rat and demonstrates the effectiveness of blocking by rat in the experiment. The relative efficiency is given by

$$\begin{aligned}
RE &= \frac{(\nu_{rcb} + 1)(\nu_{crd} + 3)\hat{\sigma}_{crb}^2}{(\nu_{rcb} + 3)(\nu_{crd} + 1)\hat{\sigma}_{rcb}^2} \\
&= \frac{(37)(48)}{(39)(46)} \cdot \frac{0.043758}{0.0083487} = 5.2413
\end{aligned} \quad \text{Eq. 4-6}$$

This means that blocking has reduced the variance of experimental units approximately  $80\% = 1 - \frac{0.0083487}{0.043758}$ , and that it would take

approximately five times as many trials to have the equivalent variances for treatment means if each rat had been used for only one trial in a CRD design, and the rat-to-rat variability had not been removed from the error term.

#### 4.6. Determining the Number of Blocks

The  $F$ -test for treatment or dose effect in the last example was highly significant ( $P < 0.0001$ ). If the experiment were to be repeated in a situation where the variability of the response (lever press rate within a rat) and differences in treatment means were to remain approximately the same, fewer blocks or rats would be required to detect significant differences in treatments. The noncentrality parameter for the  $F$ -test of treatment effects in the randomized complete block design is  $\lambda = \left(\frac{b}{\sigma^2}\right)\sum_j \tau_j^2$ , and the degrees of freedom are  $v_1 = t - 1$ , and  $v_2 = (b - 1)(t - 1)$ . Therefore, in order to calculate the number of blocks that will result in a power between 0.8 and 0.9 for detecting a difference in treatment means, the R code in Section 3.5.2 can be modified by changing the formula for the denominator degrees of freedom and the noncentrality factor.

The R code below can be used for calculating the power for a randomized block design as a function of the number of blocks. Using the results from the last experiment, the estimate of  $\hat{\sigma}_{rcb}^2 = 0.0083487$  and  $css = \sum_j \tau_j^2$  can be estimated to be

$$(.764 - .9142)^2 + \dots + (0.850 - .9142)^2 = 0.460208.$$

Using these as inputs to a more general  $F$ -power function that takes the degrees of freedom, and the noncentrality parameter as arguments, the results are created that are shown below the code. There it can be seen that a power greater than 0.99 can be achieved with only  $b = 2$  blocks or rats in the experiment.

```
library(daewr)
bmin <- 4
bmax <- 5
alpha <- .05
sigma2 <- 0.0083487
```

```

css <- 0.0460208
nu1 <- 5-1
blocks <- c(bmin:bmax)
nu2 <- (blocks - 1) * nu1
nc <- (blocks * css) / sigma2
Power <- Fpower( alpha, nu1, nu2, nc )
data.frame(blocks, nu1, nu2, nc, Power)

```

	blocks	nu1	nu2	nc	Power
1	4	4	12	22.049	0.88778
2	5	4	16	27.562	0.96711

If an estimate  $\hat{\sigma}_{crb}^2$  were available from previous experiments or pilot studies, Hinkelmann and Kempthorne (1994) have shown the relative efficiency (RE)  $\hat{\sigma}_{crb}^2$  can also be used to get a rough estimate of the number of blocks required for an RCBD. For example, suppose  $\hat{\sigma}_{crb}^2$  were estimated to be 0.040865 from previous experiments and the number of replicates of each treatment required for a CRD design to achieve adequate power for detecting a practical difference in means was  $r = 20$ , determined by the methods of **Section 3.5.2**. If blocking was expected to reduce the variance by 90% (i.e.,  $\hat{\sigma}_{rcb}^2 = 0.10 \times \sigma_{crd}^2$  or  $RE = 10.0$ ). Then the number of blocks required to achieve the same power with an RCB design is  $b = r/RE = 20/10 = 2$ .

## 4.7. Factorial Designs in Blocks

Blocking is even more effective when combined with a factorial design in treatment factors. A **Randomized Complete Block Design (RCBD)** is defined by an experiment whose treatment combinations are assigned randomly to the experimental units within a block. Generally, blocks cannot be randomized as the blocks represent factors with restrictions in randomizations such as location, place, time, gender, ethnicity, breeds, etc. It is not simply possible to randomly assign a particular gender to a person. It is not possible to pick a country and call  $X$  country. However, the presence of these factors (also known as **nuisance factors**) will introduce systematic variation in the study. For example, the crops produced in the northern vs the southern part will get exposed to different climate conditions. Therefore, they should be controlled whenever possible. Controlling these nuisance factors by blocking will

reduce the experimental error, thereby increasing the precision of the experiment and many other benefits. In the **completely randomized design (CRD)**, the experiments can only control the random unknown and uncontrolled factors (also known as lucking nuisance factors). However, the RCBD is used to control/handle some systematic and known sources (nuisance factors) of variations if they exist.

RCBD is arguably the most common design of experiments in many disciplines, including agriculture, engineering, medical, etc. In addition to the experimental error reducing ability, the design widens the generalization of the study findings. For example, if the study contains the place as a blocking factor, the results could be generalized for the places. A fertilizer producer can only claim that it is effective regardless of the climate conditions when it is tested in various climate conditions.

The “**complete block**” part of the name indicates that each treatment combination is applied in all blocks. If a block misses one or more treatment combinations, the experiment would be called **Randomized Incomplete Block Design (RIBD)**. The design would still be called randomized because the treatment combinations are randomly assigned to the experimental units within the blocks. If a block is only missing data points from a couple of observation units, the experiment will still be called RCBD with missing data, but not “incomplete block design.”

#### **4.8. Example of an RCBF – Cancer Research**

As an example of a blocked factorial experiment consider the data in **Table 4-5**. This experiment was conducted by Festing (2003) to determine whether BHA (a common antioxidant used in processed foods) induced activity of the liver enzyme EROD in mice and whether this activity was independent of the strain of mice. It was part of a larger study to determine if antioxidants help protect against cancer.

**Table 4-5. Activity of EROD Liver Enzyme in Control and BHA-Treated Mice**

Strain	Block 1		Block 2	
	Treated	Control	Treated	Control
A/J	18.7	7.7	16.7	6.4
129/Ola	17.9	8.4	14.4	6.7
NIH	19.2	9.8	12.0	8.1

BALB/c	26.3	9.7	19.8	6.0
--------	------	-----	------	-----

The factors in this experiment were *A*, whether a mouse was treated with BHA or not, and *B*, the strain of the mouse. Since you cannot assign a particular mouse to be of a certain strain, the experimental unit is not the mouse but the trial or conditions existing in the lab when a particular experiment was run. One run consisted of selecting a mouse from a specified strain; then either incorporating BHA in the diet (for a 3-week period) or not (depending on what was specified); and finally humanely sacrificing the mouse and performing an autopsy to determine the activity level of the enzyme EROD in the liver. Since the results of in vivo experiments like this can vary substantially from time to time in the same laboratory due to differences in reagents used for the enzyme analysis, calibration of instruments, and environmental factors in the animal husbandry, the experiments were blocked in time. Two mice were selected from each of four strains, one was randomly chosen to receive BHA in the diet, and eight trials or runs were conducted simultaneously. This represented one block. Three months later, the whole process was repeated for a second block of runs. Since there are two levels of the treatment factor (BHA treated or control) and four levels of the factor strain, there were  $2 \times 4 = 8$  experimental units or runs per block.

The model for analysis of a two-factor factorial in a randomized block design, like that shown in **Table 4-5**, is

$$y_{ijk} = \mu + b_i + \alpha_j + \beta_k + \alpha\beta_{jk} + \varepsilon_{ijk} \quad \text{Eq. 4-7}$$

where  $b_i$  represents the block effect,  $\alpha_j$  represents the treatment factor effect, and  $\beta_k$  represents the strain factor effect. This model is easily generalized to multi-factor factorial designs in randomized blocks. Notice that in **Eq. 4-7** there is an interaction  $\alpha\beta_{jk}$  between the two factorial factors, but there is no interaction between the block factor,  $b_i$ , and the factorial factors,  $\alpha_j$  and  $\beta_k$ . The interactions with blocks are the error term for the analysis and if they are included in the model there will be zero degrees of freedom for  $sse$ .

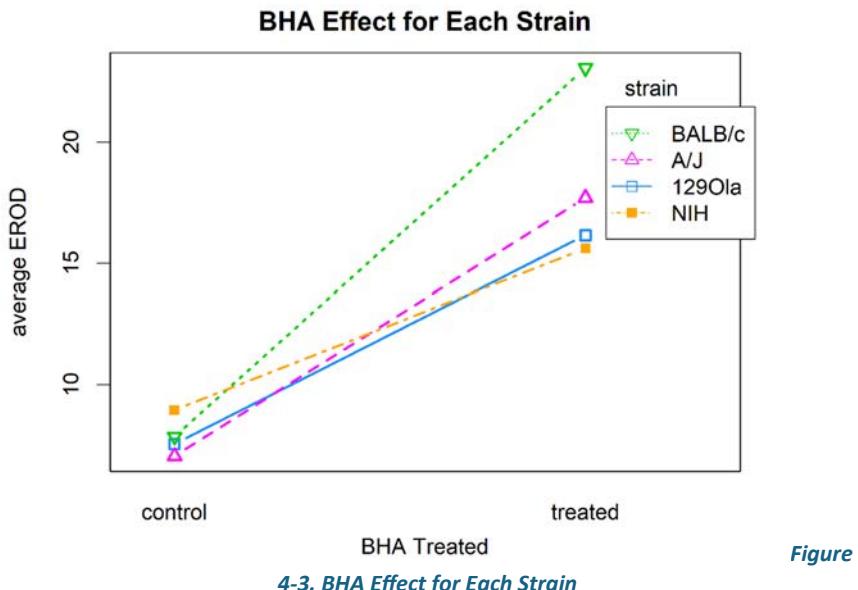
The R code commands to perform the analysis are shown below with the resulting ANOVA table. There it can be seen that BHA treatment,

strain, and the interaction are significant. The block sums of squares is also large resulting in a relative efficiency  $RE = 2.11$ . This means that it would take more than twice as many mice to have the same power or sensitivity if the experiments were not blocked by time.

```
library(daewr)
mod3 <- aov( y ~ block + strain * treat, data = bha)
summary(mod3)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)						
block	1	48	48	18.37	0.0036 **						
strain	3	33	11	4.24	0.0527 .						
treat	1	422	422	162.96	4.2e-06 ***						
strain:treat	3	40	13	5.19	0.0337 *						
Residuals	7	18	3								
---											
Signif. codes:	0	'***'	0.001	'**'	0.01	'*'	0.05	.	0.1	' '	1

Figure 4-3 helps in interpreting the factorial effects and interaction. It can be seen that on the average, BHA added to the diet increases activity of the enzyme EROD in the mouse liver. However, this increase is nearly doubled for mice of strain BALB/c.



Figure

## 4.9. Generalized Complete Block Design

When experimental units represent physical entities, smaller groups or blocks of experimental units usually result in greater homogeneity. The larger the group, the more likely it is to have many experimental units that are widely different than the norm or average. For that reason it is inadvisable to have blocked designs with more than the minimum,  $t$ , experimental units per block, where  $t$  is the number of levels or combination of levels of treatment factors. However, in some cases where experimental units represent trials rather than physical entities and the experimental runs can be made quickly, larger block sizes may not increase the variability of experimental units within a block. In that case, a design with replicates of each treatment level within a block (called a **generalized complete block design** or **GCB**) can be used.

Consider the following example from Golf Magazine (Bastable, 2006, June). An experiment was conducted to determine the ideal tee height for driving a golf ball as far as possible. The purpose was to recommend to all readers of the article what tee height they should use. The treatment factor was the tee height as shown in **Table 4-6**. The data is in **Table 4-7**.

**Table 4-6. Treatment Factor Levels for Golf Experiment**

Level	Tee Height
1	Entire ball below crown
2	Half the ball above the crown
3	Bottom of ball at top of club-face

An experiment consisted of a golfer hitting a golf ball from a specified height, and the response was the distance the ball traveled. To make a general conclusion, a representative group of golfers had to be used rather than one golfer. Since the ability to drive the ball by the golfers used in the study differed, it made sense to group or block the trials by golfer and randomize the order that each golfer hit a ball from each of the tee heights. However, since hitting more than three golf balls would not be likely to fatigue a golfer and cause more variability in his driving distance, there was no need to restrict a golfer to hitting just three balls. Instead, each golfer hit  $r = 5$  golf balls from each of  $t = 3$  tee heights.

The results from this experiment are shown in the Appendix at the end of this chapter. Nine golfers were used in this part of the study; each golfer hit five balls from each tee height (15 balls total) in a random order.

**Table 4-7. Data from the Golf Experiment**

Tee	Golfer ID								
Hgt.	1	2	3	4	5	6	7	8	9
1	142	169.5	142.7	185.4	222.2	133.6	165.2	174.3	229.7
1	141.8	177	136.2	164.8	201.9	132.6	173.2	160.1	220.7
1	153.7	169.1	140.2	173.9	192.5	135	174.2	162.8	240.4
1	130.6	176.5	143.3	191.9	182	147.6	176.9	174.6	219.5
1	147.8	173.8	145.8	164.5	224.8	136.7	166.4	172.6	225.6
2	142.7	185.6	137.8	184.7	197.7	145.5	178.8	184.4	241.6
2	136.2	164.8	159	172.8	229.8	154.5	163.4	181.8	242.1
2	140.2	173.9	151.1	175.8	203.3	150.5	160.2	185	243.4
2	143.3	191.9	154.1	184.7	214.3	137.9	160.6	192.4	240.8
2	145.8	164.5	135	172.2	220.9	154.4	169.3	193.3	240.7
3	137.8	184.7	142	176	221.8	145.9	172.8	180.6	243.3
3	159	183	141.8	177	240	146	183.2	172.5	242.1
3	151.1	195.9	153.7	175.3	221.4	149.2	170.2	181.2	236.1
3	154.1	194.4	130.6	176.5	234.9	145.2	169.6	178.4	248.3
3	135	182.2	147.8	173.8	213.2	147.2	169.9	167.6	240.4

Since there are replicate experimental units for each treatment in each block, it is possible to fit the model

$$y_{ijk} = \mu + b_i + \tau_j + b\tau_{ij} + \varepsilon_{ijk} \quad \text{Eq. 4-8}$$

However, this leaves a dilemma. The *msE* in the traditional ANOVA and the denominator for the *F*-tests for treatment effect and the block by treatment interaction is based on the variability of experimental units within the same treatment and block (in this case golfer). If the interaction between block (golfer) and treatment factor (tee height) were significant, its interpretation would imply that the optimal tee height could be different for different golfers. The golfers in the study were just a sample of golfers, and if the optimal tee height were different for them, there would be no way to recommend an optimal tee height for all readers of the *Golf Magazine* article.

To make a general recommendation, the treatment factor should be tested using the block by treatment interaction mean square as the denominator of the  $F$ -test. If the mean square for tee height is significantly larger than the mean square for the interaction between golfer and tee height, it would be justification for making a general recommendation for the optimal tee height (another justification for using the block by treatment interaction as the denominator of the  $F$ -test for treatment will be given in Chapter 5, **Section 5.9**). An  $F$ -test for treatments constructed in this way is not made automatically by the `aov` function in R but can be specified using the `Error(id/teehgt)` option in the `aov` function as shown in the code below where `id` represents block or golfer id and `teehgt` represents the level of the treatment factor.

```
library(daewr)
data(rcb)
mod4 <- aov(cdistance ~ teehgt + Error(id/teehgt),
             data = rcb)
summary(mod4)
```

```
Error: id
      Df Sum Sq Mean Sq F value Pr(>F)
Residuals 8 124741   15593

Error: id:teehgt
      Df Sum Sq Mean Sq F value Pr(>F)
teehgt     2    1724     862     5.85  0.012 *
Residuals 16    2356     147
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Error: Within
      Df Sum Sq Mean Sq F value Pr(>F)
Residuals 108   7341      68
```

The sum of squares and mean square for block or `golfer id` is shown in the section of output labeled `Error: id` and the test for the treatment effect (or tee height) is shown in the section of output labeled `Error: id:teehgt`. In that section, what is labeled as `Residuals` is actually the block by treatment interaction sum of squares, and that

mean square is used in the denominator of the  $F$ -test. The final section of the output shows the residual sum of squares and mean square, which is not used in testing.

In the results, it can be seen that the tee height is significant at the  $\alpha = 0.0124$  level. This is much different than the incorrect  $P$ -value (1.13e-05) that would result from the command:

```
mod4a <- aov( cdistance ~ id*teehgt, data = rcb)
summary(mod4a)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
id	8	124741	15593	229.41	< 2e-16 ***
teehgt	2	1724	862	12.68	1.1e-05 ***
id:teehgt	16	2356	147	2.17	0.01 *
Residuals	108	7341	68		
---					
Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1					

This model uses the residual mean square with 108 degrees of freedom for the

denominator of the  $F$ -test. In this example, either way of making the  $F$ -test for treatment indicates significance, but in general the  $F$ -test for treatment may sometimes be significant when testing with the error mean square as the denominator of the  $F$ -test while insignificant when testing with the block by treatment mean square as the denominator to the  $F$ -test. In that case, the result of the  $F$ -test using the interaction mean square should be used if you want to generalize conclusions to all potential blocks. An alternate way of accomplishing the correct test is to create a new response by averaging the responses in each block by treatment combination and then fitting the normal RCB model as shown on the next page.

Sometimes there is a more powerful test of treatment effects than can be obtained using the block by treatment interaction term as the denominator of the  $F$ -test. If the block by treatment interaction term itself is insignificant, the additive model  $y_{ijk} = \mu + b_i + \tau_j + \varepsilon_{ijk}$  can be fit to the data and the denominator of the default  $F$ -test for treatments will use the  $msE$  term that is a pooled or weighted average of the interaction and error term from model (Eq. 4-8). If the interaction is

negligible, the interaction mean square is estimating the same experimental error as the error mean square. In that case, pooling these two mean squares increases the degrees of freedom for the error term and increases the power or sensitivity of the *F*-test for treatment effects.

Normally the preliminary *F*-test of the interaction, which is used to decide whether to fit the additive model or not, is conducted at a higher significance level like  $\alpha = 0.25$ . If the interaction is significant at the  $\alpha = 0.25$  level, use the interaction mean square as the denominator of the appropriate *F*-test for treatment effects. If the interaction is not significant at  $\alpha = 0.25$ , fit the additive model and use the default *F*-test for treatments. This procedure is called the “Pool or not to Pool” procedure. For the golf experiment, the interaction sum of squares is significant at the  $\alpha = 0.0102 < 0.25$  level, and thus the additive model should not be fit, and there is no more powerful *F*-test for treatments than the one shown above.

## 4.10. Two Block Factors LSD

It was first shown in agricultural experiments (Fisher, 1958) that the process of grouping experimental plots into homogeneous blocks might profitably be duplicated. For example, on the left side of **Figure 4-4** we see a representation of a randomized block design (RCB) laid out in a field. In this design, one field, roughly square in shape, is divided into four rectangular blocks. Each block is further divided into four plots and the treatment levels (*A*, *B*, *C*, *D*) are randomly assigned to one plot within each block. In the figure the random assignment within the block is represented by each letter only appearing once in each row that represents a block. If there is a fertility gradient running from top to bottom in this field, the plots within a block will be more alike in fertility than plots in different blocks or rows, and the randomized block design would work well to reduce the variance of experimental errors within a block.

If there were no clear gradient in the field, but adjacent plots tended to be more alike than plots at different ends of the same block, a design like the one shown on the right side of **Figure 4-4** assigns each treatment level only once to each row and once to each column. In that way, the

variability from column to column can also be removed from the error sums of squares, further increasing the sensitivity for detecting treatment effects. The design shown on the right side of **Figure 4-4** is called a **Latin-square design** or **LSD**, and it is blocked both horizontally and vertically. The restriction with an LSD is that the number of row blocks equals the number of column blocks, which equals the number of levels of the treatment factor. This restriction will be relaxed in Chapter 7, where more general row-column blocking schemes will be discussed.

Block	RCB				Latin Square			
	B	C	D	A	B	D	C	A
1	B	C	D	A				
2	A	D	C	B				
3	B	D	A	C				
4	C	B	A	D				

**Figure 4-4. Comparison of RCB and Latin-Square Designs**

The model for an LSD is written

$$y_{ijk} = \mu + r_i + c_j + \tau_k + \varepsilon_{ijk} \quad \text{Eq. 4-9}$$

where  $r_i$  represents the row blocking factor,  $c_j$  represents the column blocking factor, and  $\tau_k$  represents the treatment factor. Like **Eq. 4-1** for the RCB design, no interactions are included in the model so that any differences in treatment factor levels can be generalized over rows and columns.

Latin-square designs can be used whenever there are two independent blocking factors that can be used to group experimental units. For example, if an experiment were being conducted to determine the effect of tread design on the wear life of automobile tires, the experimental unit would be a wheel on a car and the treatment factor would be the tread design of the tire mounted on that wheel. It would make sense to block the experimental units by type of automobile, since tires may wear faster on heavier cars than they do on lighter cars. It

would also make sense to block by position of the tire on a car since front right tires wear at a different rate than left rear tires, and so forth. These are independent blocking factors because all four-wheel positions exist on any type of car. To use an LSD, the number of tread types compared, and the number of car types used in the study must be four in order to equal the number of wheel positions on a car. The row blocking factor in the Latin square would represent the type of car, with four alternatives ranging over the class of cars to which the experimenter would like to make inference. The column blocking factor would represent the position of a tire on the car (FL,FR,RL,RR), and the treatment factor would represent the four different ad designs being tested.

#### 4.10.1. Creating and Randomizing Latin-Square Designs

Latin-square designs are easy to create by cyclically rotating the letters or symbols used to represent the treatment factor levels. For example, for a  $5 \times 5$  Latin square, let the letters  $A, B, C, D$ , and  $E$  represent the levels of the treatment factor. Then the design is created as:

A	B	C	D	E
B	C	D	E	A
C	D	E	A	B
D	E	A	B	C
E	A	B	C	D

To prevent biases from unknown lurking variables, randomization should be used in LSDs. However, care must be taken so that after randomization each treatment level still occurs once in each row and once in each column. This can be accomplished by first randomizing the order of the rows (keeping the column positions fixed in each row), then randomizing the columns (keeping row positions fixed within each column), and finally randomizing the order of the treatment labels. This can be easily accomplished with the [design.lsd](#) function in the R [agricolae](#) package.

The code below illustrates the use of the function [design.lsd](#) to create and randomize a plan to study the effect of the number of shelf facings on the sales of toothpaste in drugstores. The treatment factor is the number of shelf facings (1-4), the column blocking factor is the store

(to account for store-to-store differences), and the row blocking factor is the calendar week (to account for seasonal factors). The response would be the weekly sales in dollars. The first six lines of the resulting randomized data frame `lsd` is shown below the function call, and it could be used to make an electronic data collection form as shown in previous chapters.

```
tmts <- c(1, 2, 3, 4)
outdesign <- design.lsd( tmts, seed = 23)
lsd <- outdesign$book
levels(lsd$row) <- c("Week 1", "Week 2", "Week 3",
"Week 4")
levels(lsd$col) <- c("Store 1", "Store 2", "Store 3",
"Store 4")
head(lsd)
```

	plots	row	col	tmts
1	101	Week 1	Store 1	3
2	102	Week 1	Store 2	1
3	103	Week 1	Store 3	2
4	104	Week 1	Store 4	4
5	201	Week 2	Store 1	2
6	202	Week 2	Store 2	4

#### 4.10.2. Analysis of a Latin-Square Design

Latin-square designs are frequently used in steer or dairy cow feeding experiments and in bioequivalence studies to compare different formulations of a drug in phase II clinical trials. In these studies the column blocking factor is time and the row blocking factor is animal or human subject. In some cases, the treatment administered in one time period may have a carryover effect on the response in the next period. However, if there is a sufficient washout period between column blocks, there will be no carryover effects and the data can be analyzed as a traditional Latin square.

To illustrate the analysis of data from a Latin-square, consider the following bioequivalence study. The data is shown in **Table 4-8** (taken from Selwyn and Hall (1984)).

**Table 4-8. Treatment and Resulting AUC for Bioequivalence Study**

Subject	Period		
	1	2	3
1	A 1186	B 642	C 1183
2	B 984	C 1135	A 1305
3	C 1426	A 1540	B 873

The purpose was to test the bioequivalence of three formulations (**A=solution**, **B=tablet**, **C=capsule**) of a drug as measured by the AUC or area under the curve, which relates the concentration of the drug in the blood as a function of the time since dosing. Three volunteer subjects took each formulation in succession with a sufficient washout period between. After dosing, blood samples were obtained every half-hour for four hours and analyzed for drug concentration. AUC was calculated with the resulting data. Since there may be a large variation in metabolism of the drug from subject to subject, the subject was used as a row blocking factor. Since the absorption and metabolism of a drug will vary from time to time for a particular subject, time was used as a column blocking factor.

The R code to open the data frame and fit model with *Eq. 4-9* is shown below.

```
library(daewr)
mod6 <- aov( AUC ~ Subject + Period + Treat,
            data = bioeqv)
summary(mod6)
```

The resulting ANOVA table, shown on the next page, indicates that there is no difference in the three formulations.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Subject	2	114264	57132	0.26	0.79
Period	2	45196	22598	0.10	0.91
Treat	2	15000	7500	0.03	0.97
Residuals	2	442158	221079		

For illustrative purposes, the code and results below show the means for the three treatments, and a Tukey multiple comparison.

```
model.tables( mod6, type = "means" )$tables$Treat
```

```
Treat
  A      B      C
1198.7 1105.7 1120.3
```

```
TukeyHSD( mod6, "Treat")
```

```
Tukey multiple comparisons of means
95% family-wise confidence level
```

```
Fit: aov.default(formula = AUC ~ Subject + Period + Treat, data = bioeqv)
```

```
$Treat
  diff      lwr      upr     p adj
B-A -93.000 -2354.5 2168.5 0.96867
C-A -78.333 -2339.8 2183.2 0.97757
C-B  14.667 -2246.8 2276.2 0.99920
```

#### 4.10.3. Determining the Number of Replicates

The number of replicates of each treatment factor level, in an LSD with  $t$  rows and  $t$  columns must be equal to  $t$ . The only way the power for detecting differences in treatment means can be increased would be to increase the number of rows or columns. In the last example the row blocking factor represented subjects, and the column blocking factor represented periods. One way of increasing the power for detecting differences in treatment means would be to increase the number of subjects. If the number of subjects,  $r$ , were doubled, that is,  $r = 2t$ , it would be essentially the same as replicating the Latin square with  $t$  additional subjects.

In general, if we consider a replicated Latin square to have  $r = nt$  rows (where  $n$  is an integer),  $t$  columns and  $t$  levels of the treatment factor, the model for the data will still be Equation (4.9), but the degrees of freedom for the error term will be  $\nu_2 = (r - 2)(t - 1)$ . The noncentrality factor for the  $F$ -test of no treatment effects is  $\lambda = nt \sum_k \frac{\tau_k^2}{\sigma^2}$  where  $n$  is the number of times the square has been repeated.

Therefore in order to calculate the number of replicates of the square,  $n$ , that will result in a power between 0.8 and 0.9 for detecting a difference in treatment means, the R code in **Section 4.6** can be

modified by changing the formula for the numerator and denominator degrees of freedom, and the noncentrality factor.

## **4.11. Review of Important Concepts**

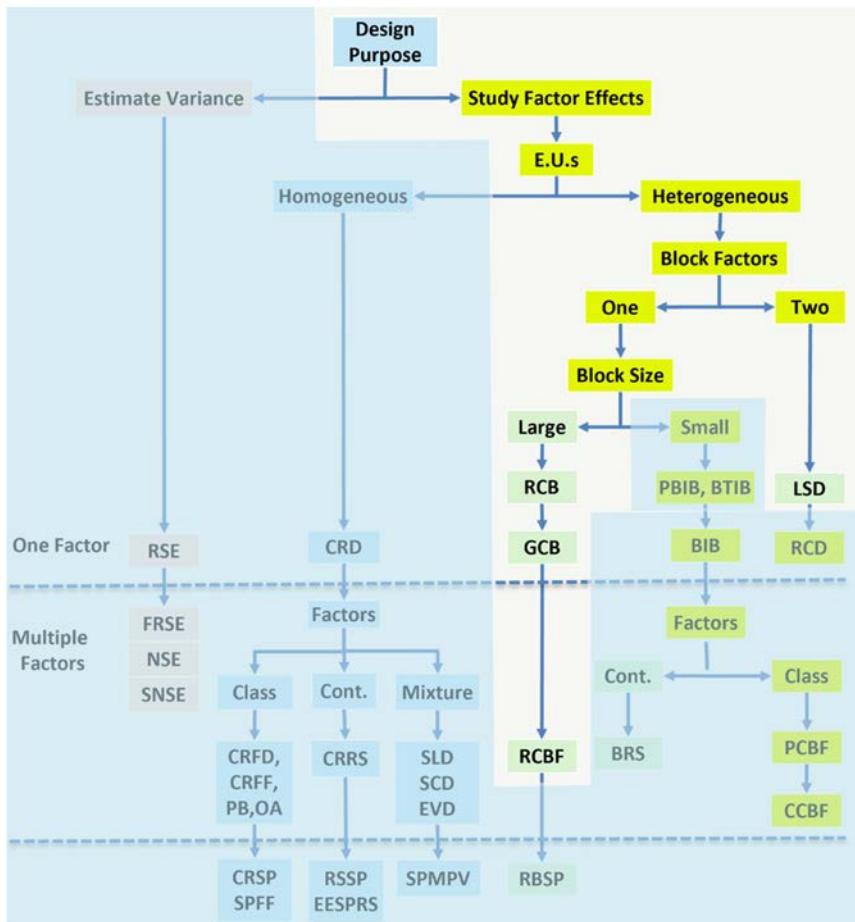
When experimental units are not homogeneous and the experimenter does not want to restrict the conclusions of his study to a homogeneous subset of experimental units, the randomized block design can be used. In the randomized block design, heterogeneous experimental units are grouped into homogeneous subgroups or blocks prior to assigning them to treatment levels. This allows the variation between blocks to be removed from the error sums of squares and increases the power or sensitivity for detecting treatment effects. The **relative efficiency (RE)** is a measure of the efficacy of blocking, and the higher the value for *RE* the more effective the blocking. **Figure 4.4** illustrates when the various types of blocked designs should be used.

When there is one treatment factor, and the heterogeneous experimental units can be grouped into categories based on one blocking factor the randomized complete block or RCB design is used. If the block size, or number of experimental units in a block, can be larger without increasing the variance of experimental units within a block the generalized complete block or GCB design can be used. If there are multiple factors under study, then the randomized complete block factorial or RCBF should be used. If the heterogeneous experimental units can be grouped into classifications based on two independent blocking factors and there is only one treatment factor, the Latin-square design or LSD should be used.

A randomized complete block design or RCB has only one experimental unit per treatment level per block. The model for the RCB design is  $y_{ij} = \mu + b_i + \tau_j + \varepsilon_{ij}$ , which does not include the interaction between block and treatment because this is the correct error for testing treatment effects if the experimenter wants general conclusions.

In a generalized complete block design, replicates of each treatment level within each block are included. Because of the replicates in the GCB, the interaction between block and treatment can be included in

the model,  $y_{ijk} = \mu + b_i + \tau_j + b\tau_{ij} + \varepsilon_{ijk}$ , but it should still be used as the error term for testing the treatment effects and this can be accomplished by using the `+ Error( )` in the `aov` model as shown in [Section 4.9](#).



**Figure 4-5. Design Selection Roadmap**

In RCBF every combination of levels of the factors is randomized to experimental units within a block. The model for analysis is the usual factorial model with the addition of a block term,  $y_{ijk} = \mu + b_i + \alpha_j + \beta_k + \alpha\beta_{jk} + \varepsilon_{ijk}$ . No interaction between blocks and factorial effects or interactions should be included in the model if the experimenter wants general conclusions. Writing a model that does not represent the way

the experiment was conducted can result in using the wrong mean squares in the denominators of the *F*-tests and may cause the wrong conclusions to be drawn.

With two independent blocking factors, LSDs can be used to further decrease the variance of experimental error. The model for the Latin square is  $y_{ijk} = \mu + r_i + c_j + \tau_k + \varepsilon_{ijk}$ , and again no interaction between row blocks, column blocks, or treatment factors is included in the model. When randomizing a list of experiments for an LSD, care must be taken so that after randomization each treatment factor level still occurs exactly once in each row block and each column block.



## **5. Designs to Study Variances**

---

### ***5.1. Introduction***

In the experiments described in Chapters 2 through 4, the purpose was to compare the response between different levels of controllable variables or factors in order to predict what the response might be in the future at specific levels of the factors, or to recommend the best factor level for future use. Another purpose of experimentation is to study sources of variability in the response. For example, cholesterol, blood glucose, and other diagnostic tests made by doctors are known to vary with the procedures and the equipment used to make the measurements. Experiments might be conducted to find out how much of the variability is due to equipment and how much is due to procedure. Symbolically  $\sigma_T^2 = \sigma_p^2 + \sigma_e^2$  where  $\sigma_T^2$  is the total variance,  $\sigma_p^2$  and  $\sigma_e^2$  are the portions of the total due to procedure and equipment, respectively.  $\sigma_p^2$  and  $\sigma_e^2$  are called the components of variance or variance components. An experiment can be conducted to collect data so that the variance components can be estimated. In this type of experiment, there might not be any interest in the difference in average diagnostic readings between specific pieces of equipment because there are (and will continue to be) many in use.

### ***5.2. Reason for Studying Variability***

There are at least three reasons for conducting experiments to study the sources of variability.

1. In some cases the purpose may be descriptive, and the variance components have value in themselves.
2. A second reason for quantifying the sources of variability is to gain insight into how to reduce the variance of the response.
3. A third reason for studying the sources of variability is to stimulate ideas about the causes of variability that could be tested in further experiments.

Two examples of where variance components are useful as descriptive measures are in genetics and in educational and psychological testing (see Searle et al., (1992)). In dairy cow breeding, the variability in milk

production can be partitioned into the amount due to the sire and the daughter, that is,  $\sigma_T^2 = \sigma_s^2 + \sigma_d^2$ . The ratio of  $h = \frac{4\sigma_s^2}{\sigma_s^2 + \sigma_d^2}$  is called the heritability and is highly important to dairy farmers. In psychological and educational testing the variability in test scores can be partitioned into the person-to-person variability and the repeat test scores for the same person, that is,  $\sigma_T^2 = \sigma_p^2 + \sigma_r^2$ . In this case,  $\frac{\sigma_p^2}{\sigma_p^2 + \sigma_r^2}$  is called the intra-class correlation, and high values of it imply reliability of the testing procedure.

In industrial quality control there is a need to reduce variability in process measurements of key product and process characteristics. If one cannot measure accurately, there is no hope of controlling or improving quality. Total measurement variability within a plant can be attributed to the measurement equipment (or gage) and the operator (or inspector) making the measurement, that is,  $\sigma_T^2 = \sigma_g^2 + \sigma_o^2$ . In order to reduce measurement variability, management needs to know where to concentrate its efforts. If the major proportion of variability is  $\sigma_g^2$ , perhaps effort needs to be placed on recalibrating measurement equipment or buying more new and more precise and consistent equipment. On the other hand, if the major source of measurement variability is  $\sigma_o^2$ , perhaps better training of operator-inspectors may solve the problem.

In some cases, a researcher would like to conduct an experiment like those described in Chapters 2–4 to compare the average response caused by different levels of controllable factors; however, he or she has such limited knowledge about the mechanism under study that it is difficult to hypothesize what factors or levels of factors to study. In this situation, determining the sources of variability in the response may prompt ideas about what factors would be most profitable to study. For example, knowing whether the majority of variability in an industrial process is batch-to-batch or within a batch would give insight as to whether factors that could be varied within a batch, or factors that could be varied from one batch to another, should be studied in optimization experiments. In cases where one factor experiment or a factorial experiment has been conducted and nothing was found to be significant,

Leitnaker and Cooper (2005) suggest that a follow-up sampling study (to classify the sources of variability in the response) may explain why no significant factors were found.

### 5.3. Random Factors and Random Sampling Experiments

When the purpose of experimentation is to study differences in the average response caused by differences in factor levels (like the experiments described in Chapters 2–4), the factors in the experiment are called **fixed factors**. The levels of these factors are specifically selected by the experimenter. On the other hand, when the purpose of experimentation is to study the variance caused by changing levels of a factor, the factor is called a **random factor**. For example, the model  $y_{ij} = \mu + \tau_i + \varepsilon_{ij}$  in Chapter 2, the factor  $\tau_i$  would be considered a fixed factor. Although we have not called it a factor before, the term  $\varepsilon_{ij}$  in this model would be considered a random factor.  $\varepsilon_{ij}$  represents the effect of the  $j$ th experimental unit within the  $i$ th level of the treatment factor. In the Aerogel Density experiment described in Chapter 2, the experimental unit was the sheet of aerogel. Replicate sheets (i.e.,  $j = 1, \dots, 4$ ) were used in this experiment so that  $\sigma^2$ , the variance of the experimental units, could be estimated and used to judge the significance of the fixed factor (heating time). Since the purpose of including multiple levels (or sheets) within each heating time was to estimate  $\sigma^2$ ,  $\varepsilon_{ij}$  is considered a random factor.

Whereas the levels of fixed factors are specifically chosen by the experimenter, the levels of random factors are just samples of possible levels that could have been used. For example, in the Aerogel Density experiment, the experimenter chose 35, 40, and 45 minutes to study for the levels of heating time. However, the four replicate sheets used for each heating time represent only a sample of the aerogel sheets that could have been used in the experiment. For this reason, experiments that are used to study variances can be thought of as **random sampling experiments** or **RSE** since the factor levels are just a sample of possible levels. For example, consider the data in **Table 5-1** patterned after the international survey of apolipoproteins conducted in 1984–1985

(Henderson *et al.*, (1987)). Apo A-I is known to help clear cholesterol from arteries. However, the apo genotype yields poor predictive values when screening for clinically defined atherosclerosis. This may be in part due to difficulty in measurement

**Table 5-1. Measured Apo A-I Concentrations by Laboratory Code**

Lab	A	B	C	D
	1.195	1.155	1.021	1.163
	1.144	1.173	1.037	1.171
	1.167	1.171	1.022	1.182
	1.249	1.175	1.064	1.184
	1.177	1.153	1.094	1.175
	1.217	1.139	0.992	1.134
	1.187	1.185	1.072	1.169
		1.144		1.136

The purpose of the study was to examine and quantify the variation among laboratories with respect to their measurement of an international reference material for apo A-I and B. Several measurements of the relative concentrations of preparations of the reference material were made by 28 selected laboratories. **Table 5.1** shows data from four representative labs. The model for the data can be written as:

$$y_{ij} = \mu + t_i + \varepsilon_{ij} \quad \text{Eq. 5-1}$$

where  $y_{ij}$  is the  $j$ th measurement of apo A-I concentration at the  $i$ th laboratory,  $\mu$  is the overall average measured concentration,  $t_i$  is the laboratory effect, and  $\varepsilon_{ij}$  is the effect of the  $j$ th measurement in the  $i$ th lab. There was no interest in comparing measured apo A-I concentrations among the specific laboratories in the study since they can be considered a random sample, or representative sample, of several labs around the world. Since the purpose of including several labs was to estimate the component of variance,  $\sigma^2$ , in measured

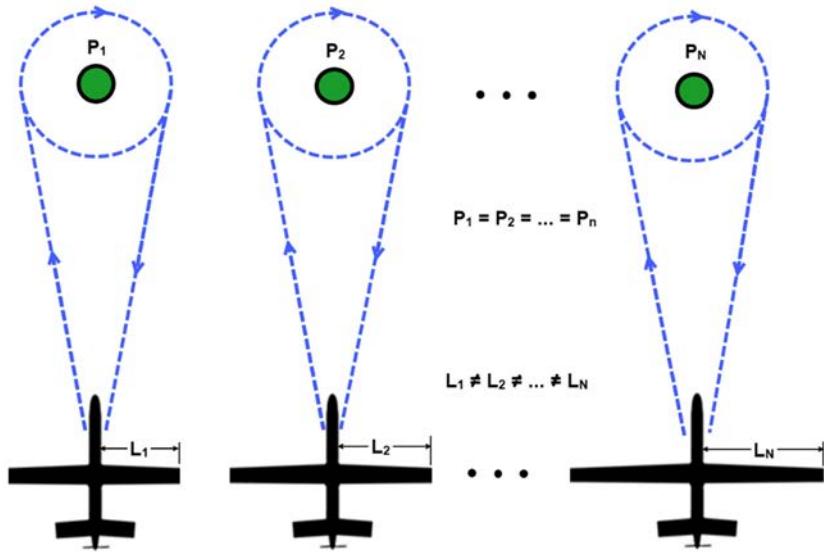
concentration due to lab,  $t_i$  can be considered a random factor and the experiment can be thought of as a sampling experiment. Note that the Roman letter  $t_i$  was used to represent the random region effect, whereas in model (*Eq. 2-2*) in Chapter 2 the Greek letter  $\tau_i$  was used to represent the fixed treatment factor. The replicate measurements of apo A-I made in each lab are only samples of the possible measurements that could be made in that lab. Since multiple measurements were made in each lab in order to estimate  $\sigma^2$ ,  $\varepsilon_{ij}$  can also be considered a random factor. Except for the  $\varepsilon$  used to represent the random experimental unit, the convention in this book will be to use Roman letters to represent random factors and Greek letters to represent fixed factors.

The usual assumptions regarding *Eq. 5-1* are that the random effects,  $t_i$  and  $\varepsilon_{ij}$ , are independent and normally distributed with zero means and variances equal to the variance components  $\sigma^2$  and  $\sigma_t^2$ , respectively. Since the variance of a sum of independent random variables is the sum of the variances,  $\sigma_y^2 = \sigma_t^2 + \sigma^2$ . Data from sampling experiments can be used to partition the variance in the response,  $\sigma_y^2$ , into the two variance components of which it is composed.

## 5.4. One-Factor Sampling Designs

We can use one-factor sampling experiments to partition variability into two sources. As an example of partitioning variance, consider the UAV experiments described in **Section 2.9**. In that example there might have been some differences in the description of the experimental unit among testers. Some might argue that the experimental unit was the material from which a UAV design was cut. Others might argue that it was the trial, or air conditions, at the time a UAV was launched and timed. If the first definition was used, then replicate experiments would consist of:

1. Assembling several UAVs where each one was made with a different wingspan,  $L_1, L_2, \dots, L_N$ .
2. Launching, and timing with the same levels of each of the other factors, i.e., once. This is equivalent to holding one factor constant while varying another (the material).
3. Recording the time on station for each UAV, given the same station parameters,  $P_i$ .



*Figure 5-1. UAV experiment – variable wingspans' effect on time-on-station*

If the second definition was used, replicate experiments would consist of repeatedly launching and measuring time-on-station one UAV. One practical way to decide how to define the experimental unit would be to partition the variability in time-on-station into UAV-to-UAV variability and variability among repeat launching of the same UAV. If a substantial part of the variability was among UAVs of the same design, there would be reason to make multiple UAVs for replicates. If, on the other hand, all of the variability was among time-on-station of the same UAV, there would be no reason to make multiple UAVs of the same design for replicates. In that case, repeat launches to the target  $P_i$  of the same UAV could be considered replicates.

The variability in UAV time-on-station can be partitioned into the UAV-to-UAV variability and within UAV variability using a one-factor sampling experiment. To do this, first randomly select six UAVs with the same levels of several factors, like body width, tail length, and wingspan. Launch and time each of the six UAVs three times each according to a randomized order like that created in **Section 2.2.1**, and record time-on-station.

As a second example of partitioning variability into two sources,

consider the following example presented by Davies (1949). A dye manufacturer wanted to know if there was an appreciable contribution to variability in dyestuff color yields owing to the quality of the intermediate acid batch used. It was a two-step process. Intermediate acid batches were produced in one step, and in a later step the acid was used to produce the dyestuff. The goal was to keep dyestuff color yields consistently high. If the majority of variation was caused by differences among the intermediate acid batches, then improvement efforts should be concentrated on the process that makes the acid batches. If the majority of variation was within preparations of the dyestuff made from the same acid batch, improvement efforts should be focused on the process step of making the dyestuff. A sampling experiment was run wherein six representative samples of *H* acid intermediate were taken from the step manufacturing process that produces it. From each acid sample, five preparations of the dyestuff Naphthalene 12B were made in a laboratory, and these were representative of the preparations that could be made with each sample. The data from the sampling experiment is shown in **Table 5-2**. The yields are given in grams of standard color.

**Table 5-2. Yields of Naphthalene Black 12B**

Sample of H Acid	1	2	3	4	5	6
Individual yields in	1440	1490	1510	1440	1515	1445
grams of standard	1440	1495	1550	1445	1595	1450
color	1520	1540	1560	1465	1625	1455
	1545	1555	1595	1545	1630	1480
	1580	1560	1605	1595	1635	1520

## 5.5. Estimating Variance Components

The model represented by **Eq. 5-1** can be expressed in matrix terms as:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \quad \text{Eq. 5-2}$$

where  $\boldsymbol{\beta}' = (\mu, t')$  and  $t'$  is the vector of random effects, and the

independence and normality assumptions can be expressed as:

$$\begin{pmatrix} \mathbf{t} \\ \boldsymbol{\varepsilon} \end{pmatrix} \sim MVN \left( \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix}, \begin{pmatrix} \sigma^2 I_t & \mathbf{0} \\ \mathbf{0} & \sigma^2 I_n \end{pmatrix} \right) \quad Eq. 5-3$$

where MV N represents the multivariate normal distribution, It is a  $t \times t$  identity matrix, t is the number of levels of the random factor  $t_i$ , and  $n = tr$  is the total number of experiments or runs. With the independence and normality assumptions, there are several methods of estimating variance components from the data obtained in sampling experiments.

### 5.5.1. Method of Moments Estimators

Fisher was the first to show how to estimate the variance components from an analysis of variance. To do this, the method of moments is used. The model for the one-factor sampling experiment is given by Equation (5.1), and the analysis of variance table is identical to **Table 2-2** in Chapter 2. The F-test in the analysis of variance table can be used to test the null hypothesis  $H_0: \sigma_t^2 = 0$  against the alternative  $H_a: \sigma_t^2 > 0$ .

To estimate the variance components from the ANOVA, the mean squares are equated to their expected values and the simultaneous equations are solved. When the treatment factor is a random factor, as in model (**Eq. 5-1**), and there are an equal number of replicates in each level, the  $msT$  term in the ANOVA table follows a distribution that is a multiple of the central chi-square distribution. This is different than the model with a fixed effect treatment factor. In that case, described in **Section 2.3**, the distribution of  $msT$  was a noncentral chi-square. In either case, the  $msT$  from the ANOVA can be represented as the quadratic form  $\mathbf{y}'\mathbf{A}\mathbf{y}$ ; and in the random effects model (**Eq. 5-1**), Hartley (1967) has shown that its expected value can be written as  $\sigma^2 + c\sigma_t^2$ , where  $c = \sum_i \mathbf{x}_i' \mathbf{A} \mathbf{x}_i$ , and  $\mathbf{x}_i$  is an indicator of the  $i$ th level of  $t_i$  (that is the  $(i + 1)$ th column in the  $\mathbf{X}$  matrix as shown in Equation (2.6)). When there is an equal number of replicates,  $r$ , in each level of the random factor, the coefficient in the expected mean square simplifies to  $c = r$ . The method of moments estimators for the variance components can be used when there are an equal number of replicates in each level of the random factor as shown in **Table 5-2**, or an unequal number as shown in

**Table 5-1.** For the equal replicates case, the estimates turn out to be uniformly best unbiased estimators, but for the unequal case, estimators that are uniformly best do not exist (see Searle et al., (1992)).

To illustrate the estimation of the variance components using R, the code to open the data frame containing data from the apo measurement sampling experiment, shown in **Table 5-1**, along with the `aov` function call to produce the ANOVA are shown below.

```
mod1 <- aov( conc ~ lab, data = Apo )
sm1 <- summary(mod1)
sm1
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)						
lab	3	0.09223	0.03074	42.11	4.01e-10 ***						
Residuals	26	0.01898	0.00073								
	---										
Signif. codes:	0	'***'	0.001	'**'	0.01	'*'	0.05	'.'	0.1	' '	1

In this case, the ANOVA table is not only printed, but also stored in the object `sm1`. By doing this, the mean squares for treatment and error can be retrieved for later calculations. The first step in calculating the expected mean square coefficient  $c$  using Hartley's (1967) method is to extract the indicators of each level of the treatment factor,  $x_i$ , from the model matrix. The R code to do that is shown below.

```
X <- as.matrix( model.matrix(mod1) )
labB <- X[,2]
labC <- X[,3]
labD <- X[,4]
labA <- X[,1]-(labB+labC+labD)
```

The next step is to compute an ANOVA using each indicator vector as the response, extract the treatment mean square from each ANOVA, and finally sum the results to get the coefficient  $c$  as shown below.

```
s1 <- summary(aov (labA ~ Apo$lab ))
x1 <- as.matrix( s1[[1]][1,3] )
s2 <- summary(aov( labB ~ Apo$lab ))
x2 <- as.matrix( s2[[1]][1,3] )
s3 <- summary(aov(labC ~ Apo$lab ))
x3 <- as.matrix( s3[[1]][1,3] )
```

```

s4 <- summary(aov(labD ~ Apo$lab ))
x4 <- as.matrix( s4[[1]][1,3] )
c <- x1 + x2 + x3 + x4

```

As shown in **Section 2.4.3** the expected value of the  $msE$  in the ANOVA table is  $\sigma^2$ , therefore two simultaneous equations, in the two unknown variance components  $\sigma^2$  and  $\sigma_t^2$ , can be created by equating the mean squares for model and error to their expected values. The R code to retrieve the mean squares and print the results is shown below.

```

sigma2 <- as.matrix( sm1[[1]][2,3] )
mslab <- as.matrix( sm1[[1]][1,3] )
cat(" Mean Square for Lab = ", mslab, "\n",
    " Mean Square for Error = ", sigma2, "\n",
    "Expected Mean Square for Lab", "\n",
    "Var(error)+", c, "Var(Lab)", "\n")

```

```

Mean Square for Lab =  0.03074443
Mean Square for Error =  0.0007301573
Expected Mean Square for Lab
Var(error)+ 7.488899 Var(Lab)

```

The solution to the equations equating the mean squares to their expected values are shown below.

$$\begin{aligned}
0.03074443 &= \sigma^2 + 7.4889\sigma_t^2 \\
0.00073016 &= \sigma^2 \\
\Rightarrow \hat{\sigma}^2 &= 0.00073016 \\
\text{and } \Rightarrow \hat{\sigma}^2 &= \frac{0.03074443 - 0.00073016}{7.4889} = 0.0040078
\end{aligned}$$

These equations could be solved using R as shown in the code below.

```

sigma2t <- (mslab - sigma2) / c
cat("Method of Moments Variance Component
    Estimates","\n", "Var(error)=",sigma2,
    "\n","Var(Lab)=",sigma2t," \n")

```

```

Method of Moments Variance Component Estimates
Var(error)= 0.0007301573

```

$\text{Var}(\text{Lab}) = 0.00400784$

Since  $\hat{\sigma}^2 < \hat{\sigma}_t^2$ , these estimates show there is much or more variability among labs than within a lab.

### 5.5.2. Interval Estimates

When the normality assumptions apply and there is an equal number of replicates for each level of the random factor in model (5.1), exact interval estimates exist for  $\sigma^2$ ,  $\frac{\sigma_t^2}{\sigma^2 + \sigma_t^2}$ ,  $\frac{\sigma^2}{\sigma^2 + \sigma_t^2}$ , and  $\frac{\sigma_t^2}{\sigma^2}$  based on the distributions of the means squares. The `aov` function in R does not produce these confidence intervals, but they can be easily calculated by hand from the statistics produced in the ANOVA table. If there are  $i = 1, \dots, T$  levels of the random factor and  $j = 1, \dots, r$  replicates in each level of the random factor, then Table 5.3 (that was taken from Searle et al. (1992)) contains exact formulas for confidence intervals on lines 1, 3-5, and an approximate formula on line 2.

As an example of calculating the confidence intervals, consider the data in the sampling experiment shown in Table 5.2 to study the variability in dyestuff color yields. From the ANOVA  $ssT = 56,358$ ,  $ssE = 58,830$ ,  $T = 6$ ,  $r = 5$ , and  $F = \frac{mST}{mse} = 4.59847$ . The upper 0.975 percentile of the chi-square distribution with 24 degrees of freedom is 39.36. This can be obtained using the R `qchisq` function as `qchisq(.975, 24)`. The upper 0.975 percentile of the  $F$ -distribution with 5 and 24 degrees of freedom is 3.15482. This can be obtained using the R `qf` function as `qf(.975, 5, 24)`. The other percentiles of the  $F$  and chi-square can be obtained similarly. The exact 95% and the exact 95% confidence interval on  $\frac{\sigma_t^2}{\sigma^2 + \sigma_t^2}$  is given by

$$\begin{aligned} & \left( \frac{\frac{F}{F_{5,24,0.975}} - 1}{r + \frac{F}{F_{5,24,0.975}} - 1}, \frac{\frac{F}{F_{5,24,0.025}} - 1}{r + \frac{F}{F_{5,24,0.025}} - 1} \right) \\ &= \left( \frac{\frac{4.59847}{3.15482} - 1}{5 + \frac{4.59847}{3.15482} - 1}, \frac{\frac{4.59847}{0.15929} - 1}{5 + \frac{4.59847}{0.15929} - 1} \right) \\ &= (0.0838, 0.8479). \end{aligned}$$

**Table 5-3. Confidence Intervals for Functions of Variance Components in One-Factor Random Model with Equal Replication**

Confidence Interval

Line	Parameter	Lower Limit	Upper Limit	Confidence Coefficient
1	$\sigma^2$	$\frac{ssE}{\chi_{T(r-1)}^2, U}$	$\frac{ssE}{\chi_{T(r-1)}^2, U}$	$1 - \alpha$
2	$\sigma^2$	$\frac{ssT(1 - \frac{F_U}{F})}{r\chi_{T-1,U}^2}$	$\frac{ssT(1 - \frac{F_L}{F})}{r\chi_{T-1,L}^2}$	$1 - 2\alpha$
3	$\frac{\sigma_t^2}{\sigma_t^2 + \sigma^2}$	$\frac{\frac{F}{F_U} - 1}{r + \frac{F}{F_U} - 1}$	$\frac{\frac{F}{F_L} - 1}{r + \frac{F}{F_L} - 1}$	$1 - \alpha$
4	$\frac{\sigma^2}{\sigma_t^2 + \sigma^2}$	$\frac{\frac{F}{F_L} - 1}{r + \frac{F}{F_L} - 1}$	$\frac{\frac{F}{F_U} - 1}{r + \frac{F}{F_U} - 1}$	$1 - \alpha$
5	$\frac{\sigma_t^2}{\sigma^2}$	$\frac{\frac{F}{F_U} - 1}{r}$	$\frac{\frac{F}{F_L} - 1}{r}$	$1 - \alpha$

Notation:

$$F = \frac{msT}{msE}$$

$$Pr\{\chi_{V,L}^2 \leq \chi_V^2 \leq \chi_{V,U}^2\} = 1 - \alpha$$

$$Pr\{F_L \leq F_{v1,v2} \leq F_U\} = 1 - \alpha$$

confidence interval on  $\sigma^2$  is given by:

$$\left( \frac{ssE}{\chi_{24,0.975}^2}, \frac{ssE}{\chi_{24,0.025}^2} \right) = \left( \frac{58830}{39.36}, \frac{58830}{12.4} \right) = (1494.51, 4744.35)$$

The approximate 90% confidence interval on  $\sigma_t^2$  is given by:

$$\begin{aligned} & \left( \frac{ssT\left(1 - \frac{F_{5,24,0.975}}{F}\right)}{r\chi_{5,0.975}^2}, \frac{ssT\left(1 - \frac{F_{5,24,0.025}}{F}\right)}{r\chi_{5,0.025}^2} \right) \\ &= \left( \frac{56358\left(1 - \frac{3.15482}{4.60}\right)}{5(12.8325)}, \frac{56358\left(1 - \frac{0.15929}{4.60}\right)}{5(0.83121)} \right) \\ &= (275.9551, 13,097.168) \end{aligned}$$

Notice the interval estimate of  $\sigma_t^2$  is much wider than the interval estimate of  $\sigma^2$ . This is because  $ssT$  only has 5 degrees of freedom in the ANOVA, while  $ssE$  has 24 degrees of freedom.

### 5.5.3. Maximum Likelihood and REML Estimators

Although the method of moments estimators are uniformly best unbiased estimators, they have one unfortunate property. When  $msT$  is less than  $msE$  in the analysis of variance, the estimator of  $\sigma_t^2$  will be negative. This can happen quite frequently if  $\frac{\sigma_t^2}{\sigma^2} \leq 0.10$ , and there are less than  $T = 10$  levels of the random factor  $t_i$ . **Maximum likelihood (ML)** and reduced or **restricted maximum likelihood (REML)** are preferred methods of estimation that avoid this problem. **REML** is an adaptation of the maximum likelihood technique that maximizes part of the likelihood. The fact that maximum likelihood estimators cannot lie outside their parameter space prevents both the ML and REML methods from obtaining negative estimates of  $\sigma_t^2$ . To understand how maximum likelihood and REML work, we will consider the equal replication case. Given the model and assumptions in **Eq. 5-2** and **Eq. 5-3**, the distribution of  $y$  can be written as:

$$y \sim MVN(\mu \mathbf{1}, V) \quad \text{Eq. 5-4}$$

where  $V$  is a block diagonal matrix with  $T$  blocks of  $(\sigma_{tJ_r}^2 + \sigma_r^{2I})$  along the diagonal. The likelihood function is

$$L(\mu, V | y) = \frac{\exp \left[ -\frac{1}{2} (y - \mu \mathbf{1}_n)' V^{-1} (y - \mu \mathbf{1}_n) \right]}{(2\pi)^{\frac{1}{2}n} |V|^{\frac{1}{2}}} \quad \text{Eq. 5-5}$$

or the equal replication case, this can be simplified to:

$$L(\mu, \sigma^2, \lambda | y) = \frac{\exp \left\{ -\frac{1}{2} \left[ \frac{ssE}{\sigma^2} + \frac{ssT}{\lambda} + \frac{(\bar{y}_.. - \mu)^2}{\frac{\lambda}{n}} \right] \right\}}{(2\pi)^{\frac{1}{2}n} \sigma^2^{\frac{1}{2}n} \lambda^{\frac{1}{2}T}} \quad \text{Eq. 5-6}$$

where  $\lambda = \sigma^2 + r\sigma_T^2$ . The maximum likelihood estimates are obtained

by maximizing the likelihood with respect to  $\mu, \sigma^2$ , and  $\lambda$ .

The REML estimates of  $\sigma^2$  and  $\sigma_t^2$  are obtained by maximizing

$$L(\sigma^2, \sigma_t^2 | ssT, ssE) = \frac{L((\mu, \sigma^2, \lambda | y)}{L(L(\mu | \bar{y}))} \quad \text{Eq. 5-7}$$

can be obtained by factoring  $L(\mu, \sigma^2, \lambda | y)$  using the fact that  $ssE$  and  $ssT$  are independent of  $\bar{y}_{..}$ . The maximization can be done analytically for the equal replication case as shown by Searle et al. (1992) and can be done numerically for the unbalanced case.

A desirable property of the REML estimates is that they are the same as the method of moments (analysis of variance estimates) when there is equal replication in each level of the random factor and  $mST > mSE$ . The maximum likelihood estimators and REML estimators are calculated by the R package `lme4` using a numerical solution for both the unbalanced and balanced cases.

To illustrate the REML estimators, consider the following case. A manufacturer of packaged dry soup mixes was experiencing excessive variability in the package weights of a dry soup mix component called the “intermix.” The intermix is a mixture of flavorful ingredients such as vegetable oil, salt, and so forth. Too much intermix in a soup packet gives it too strong a flavor, and not enough gives too weak a flavor. It was a two-step process to make the packaged soup mix. The first step was to make a large batch of soup and dry it on a rotary dryer. Next, the dried soup batch was placed into a mixer, where the intermix was added through ports as it was mixed. Then it was packaged in sealed bags of uniform weight. There were several factors that could be changed in the first step (production of the dried soup batch), and several factors that could be changed in the second step (adding the intermix and mixing) that could possibly affect the variability of the weight of the intermix in each sealed bag. A factorial experiment was to be planned to find out which factors affected variability in intermix weights. In order to determine which factors to include in a factorial experiment, a reasonable first step would be to partition the variability in intermix weight into the variability from soup batch to soup batch and the variability within a batch caused by the process to mix and add intermix

to the dried soup. If there was little variability from batch to batch, the experiment would only need to consider factors involved in the mixing step.

In order to partition the variability in package weights into batch-to-batch and within batch, a sampling experiment was performed. A random sample of four batches was selected over a month of production runs. From each batch, three 1-lb samples of the finished soup mix were taken from the mixing tank as it was being packaged. The weight of the intermix was determined for each sample. The results of the sampling experiment are shown in **Table 5-4**.

The `lmer` function in the R package `lme4` can compute the REML estimates of variance components. The code to open the data frame for the dry soup mix data and call the `lmer` function to produce the REML estimates of  $\sigma_t^2$  and  $\sigma^2$  are shown on the next page, and the results are shown below the commands.

**Table 5-4. Variability in Dry Soup Intermix Weights**

Batch	Weight
1	0.52, 2.94, 2.03
2	4.59, 1.26, 2.78
3	2.87, 1.77, 2.68
4	1.38, 1.57, 4.10

The model formula (`weight~1 +(1|batch)`) for the function `lmer` is different than the model statement for the `lm` or `aov` functions shown in Chapter 2. The response is on the left of the “`~`” operator as before, but the `1` on the right indicates that a global mean will be fit. The vertical bar “`|`” in the second term of the model formula separates the fixed effects from the random effects. The term to the left of the vertical bar is a fixed effect (the overall mean) and the term to the right (`batch`) is a random effect.

```

library(daewr)
library(lme4)
rmod2 <- lmer( weight ~ 1 + (1|batch), data = soupmx)
summary(rmod2)

```

```

Linear mixed model fit by REML ['lmerMod']
Formula: weight ~ 1 + (1 | batch)
Data: soupmx

REML criterion at convergence: 37.5

Scaled residuals:
    Min      1Q  Median      3Q     Max 
-1.56147 -0.71722 -0.01614  0.43230  1.86604 

Random effects:
Groups   Name        Variance Std.Dev.
batch    (Intercept) 0.00     0.000
Residual           1.41     1.187
Number of obs: 12, groups: batch, 4

Fixed effects:
            Estimate Std. Error t value
(Intercept) 2.3742    0.3428   6.926
optimizer (nloptwrap) convergence code: 0 (OK)
boundary (singular) fit: see help('isSingular')

```

In the output, the familiar ANOVA table is missing because it is not needed to obtain the REML estimates. The output instead shows the resulting numerical estimators. It can be seen that  $\hat{\sigma}_t^2$ , the batch-to-batch variability, is estimated to be zero, while  $\sigma^2$ , the variability caused by the mixing process within a batch, is estimated to be 1.41. If the method of moments estimators is used with this data (left as an exercise), the estimator of  $\sigma_t^2$  turns out to be negative. The conclusions from this sampling experiment indicate that further experiments to identify the causes of variability in the intermix should concentrate on factors that can be varied in the second step of the process where the intermix is added and mixed with the batch. These further experiments were performed and will be shown in the next chapter.

In processes that contain more than two steps, Leitnaker and Cooper (2005) show that **multistage sampling** experiments, to be described in

**Section 5.7**, are very useful for identifying process steps and factors within those steps that would be good candidates for further study with factorial type experiments.

An asymptotic approximate confidence interval for the variance components can be obtained in the R `lme4` package using the likelihood profile method. The `profile` and `confint` functions provide a confidence interval on the square root (i.e., standard deviations) of the variance components. The code below shows how to get the confidence intervals using the data in **Table 5-2**.

```
library(daewr)
library(lme4)
pr1 <- profile( fm1M <- lmer( yield ~ 1 + (1| sample),
    data = Naph, REML = FALSE))
confint(pr1) # 95% confidence interval on sigma
```

	2.5 %	97.5 %
.sig01	12.19854	84.06305
.sigma	38.22998	67.65770
(Intercept)	1486.45150	1568.54849

Squaring the left and right end-point of the interval for `.sigma` gives the 95%confidence interval (1461.53–4577.56) for  $\sigma^2$ , the variance component of the within sample yields of napthalene black 12B. This is very close to the exact confidence interval in **Table 5-3**.

A similar interval estimate can be obtained for  $\sigma_t^2$ . The asymptotic confidence intervals will be reasonably close to the exact confidence intervals produced using the formulas in **Table 5-3**, when the degrees of freedom for the term corresponding to the variance component being estimated is greater than 45.

#### 5.5.4. Determining the Sample Size For One-Factor Sampling Studies

Since there are two variance components being estimated in a one-factor sampling design, there are two things to consider when determining the sample size. In order to accurately estimate the replicate variance,  $\sigma^2$ , the important thing to consider is the number of degrees of freedom for error  $v_2 = t(r - 1)$ . The accuracy of the

estimate of the random factor,  $\sigma_t^2$ , will always be relative to the accuracy in estimating  $\sigma^2$ .

The accuracy for estimating  $\sigma^2$  can be expressed in terms of the width of the confidence interval given on line 1 of **Table 5-3**. Since  $E(ssE) = t(r - 1)\sigma^2$ , the expected width of the 95% confidence interval can be written as:

$$\sigma^2 \left[ \frac{t(r - 1) \times (\chi_{t(r-1), 0.975}^2 - \chi_{t(r-1), 0.025}^2)}{\chi_{t(r-1), 0.975}^2 \times \chi_{t(r-1), 0.025}^2} \right] \quad \text{Eq. 5-8}$$

Therefore, if you would like the half-width of the confidence interval to be 50% of  $\sigma^2$ , search for the number of levels of the random factor,  $t$ , and the number of replicates,  $r$ , such that the multiplier of  $\sigma^2$  in **Eq. 5-8** is 1.0. This can be done easily using the `qchisq` function in R by enumerating various cases. The example below shows the calculation of the multiplier of  $\sigma^2$  that determines the expected width of the confidence interval. The resulting output below the code shows that any combination of  $t$  and  $r$  that result in  $\nu_2 = t(r - 1)$  in the range of 36 to 38 will give the desired accuracy in estimating  $\sigma^2$ .

```
nu2 <- 36:44
chiu <- qchisq(.975, nu2)
chil <- qchisq(.025, nu2)
width <- nu2 * (chiu - chil) / (chil * chiu)
halfw <- width/2
data.frame(nu2, width, halfw)
```

	nu2	width	halfw
1	36	1.0259871	0.5129936
2	37	1.0091269	0.5045635
3	38	0.9930584	0.4965292
4	39	0.9777224	0.4888612
5	40	0.9630653	0.4815327
6	41	0.9490392	0.4745196
7	42	0.9356004	0.4678002
8	43	0.9227095	0.4613548
9	44	0.9103307	0.4551654

A simple rule of thumb can be used to get an idea as to how many levels

of the random factor,  $t$ , to include in the sampling experiment. When  $\sigma_t^2$  is expected to be larger than  $\sigma^2$ ,  $t$  should be as large as possible, so  $t = \nu_2$ , and  $r = 2$  would be reasonable.

Another way of determining both  $t$  and  $r$  would be to consider the power of the  $F$ -test for testing the hypothesis  $H_0: \sigma_t^2 = 0$ . Under the alternative hypothesis  $H_a: \sigma_t^2 > 0$ , the statistic  $F = \frac{mST}{mse}$  follows a multiple of the central  $F$ -distribution and the power or probability of exceeding the critical limit can be expressed by

$$1 - \beta = Pr\left(F_{t-1,t(r-1)} > \frac{1}{1 + r \times \rho} F_{t-1,t(r-1),\alpha}\right) \quad Eq. 5-9$$

where  $\rho = \frac{\sigma_t^2}{\sigma^2}$ . Again the sample sizes  $t$  and  $r$  that give adequate power for specified alternatives can be determined by enumerating several cases in R. In this case, the use of the R functions `qf` and `pf` make it easy. For example, if you wanted to have a power greater than  $1 - \beta = 0.90$  for rejecting  $H_0: \sigma_t^2 = 0$ , when  $\rho = \frac{\sigma_t^2}{\sigma^2} \geq 3.0$  the R code below will find some alternatives. The resulting output below the code shows several combinations of  $t$  and  $r$  that result in power greater than 0.90.

```
alpha <- .05
rho <- 3.0
t <- rep(5:7, each = 3)
r <- rep(2:4, 3)
nu_1 <- t-1
nu_2 <- t * (r - 1)
fcrit <- qf( 1 - alpha, nu_1, nu_2 )
factor <- 1 / ( 1 + r * rho )
plimit <- factor * fcrit
power <- 1 - pf( plimit, nu_1, nu_2 )
data.frame( t, r, power)
```

t	r	power	
1	5	2	0.6025330
2	5	3	0.8397523
3	5	4	0.9142402
4	6	2	0.6876308

5 6 3 0.8972133  
 6 6 4 0.9523702  
 7 7 2 0.7565926  
 8 7 3 0.9346005  
 9 7 4 0.9737459

Of course this method does not consider the accuracy of the estimate of  $\sigma^2$ . One might consider using the first method shown above to determine  $v_2 = t(r - 1)$  to have the desired accuracy in estimating  $\sigma^2$ ; and then with  $v_2 = t(r - 1)$  fixed at the number determined, use the second method shown above to determine how large  $t$  should be for adequate power in rejecting  $H_0: \sigma_t^2 = 0$ , at a specified significance level  $\alpha$ , in favor of  $H_a: \sigma_t^2 > 0$  when  $\frac{\sigma_t^2}{\sigma^2} \geq \rho$ .

## 5.6. Two-Factor Sampling Designs

When the purpose of experimentation is to study the variance in the response caused by varying the levels of two independent factors, the design is similar to the two-factor factorial design presented in Chapter 3. However, in the two-factor factorial designs presented in Chapter 3, the levels of the factors would be specifically selected by the experimenter because he would be interested in comparing the average response between these levels. In the two-factor sampling experiment, on the other hand, the levels of the factors are just a random or representative sample of possible levels, and the purpose is to determine how much of the variance in the response can be attributed to varying levels of the factors. In general these designs are called factorial random sampling experiments or FRSE.

Consider the example data presented in *Eq. 5-5* taken from Sower et al. (1999). These are data from a Gage R&R study commonly performed in industrial quality assurance departments.

*Table 5-5. Data from Gage R & R Study*

Operator			
Part	1	2	3
1	0.71	0.56	0.52
	0.69	0.57	0.54
2	0.98	1.03	1.04

	1.00	0.96	1.01
3	0.77	0.76	0.81
	0.77	0.76	0.81
4	0.86	0.82	0.82
	0.94	0.78	0.82
5	0.51	0.42	0.46
	0.51	0.42	0.49
6	0.71	1.00	1.04
	0.59	1.04	1.00
7	0.96	0.94	0.97
	0.96	0.91	0.95
8	0.86	0.72	0.78
	0.86	0.74	0.78
9	0.96	0.97	0.84
	0.96	0.94	0.81
10	0.64	0.56	1.01
	0.72	0.52	1.01

In these studies the purpose is to classify the variability in measured features of manufactured products or product components. Assuming the gage or measuring instrument is properly calibrated, a measured value determined during a quality control inspection can be considered to be a function of the true feature dimension, the gage repeatability, and the gage reproducibility. **Gage repeatability** is the ability of a single operator to obtain the same measurement value multiple times using the same measuring instrument (or gage) on the same feature of a single manufactured component (or part). **Gage reproducibility** is the ability of different operators to obtain the same measured value multiple times using the same gage on the same part. If the variability in measurements caused by the gage repeatability plus the gage reproducibility is more than 10% of the tolerance range, the measurements may not be accurate enough to be used in monitoring product quality.

The **Gage R&R sampling** experiment consists of selecting a set of manufactured parts or components that are representative of the part-to-part variability in normal manufacturing. In **Table 5-5**, a sample of 10 parts was selected and these parts represent the levels of the first factor in the sampling experiment. Next, a random or representative sample of inspectors is selected. The inspectors or operators represent the levels

of the second factor in the sampling experiment. Finally, each inspector measures each part twice in a random order and the results are assembled in a table like **Table 5-5**. The replicate measurements represent the replicates in each cell.

Since each operator or inspector measured each part, the model for the data in **Table 5-5** can be written in the form of a factorial model

$$y_{ijk} = \mu + a_i + b_j + ab_{ij} + \varepsilon_{ijk}, \quad \text{Eq. 5-10}$$

where  $y_{ijk}$  is the  $k$ th measurement ( $k = 1, \dots, r$ ) made by the  $j$ th operator ( $j = 1, \dots, b$ ) on the  $i$ th part ( $i = 1, \dots, a$ ),  $a_i$  is the part effect,  $b_j$  is the operator or inspector effect, and  $ab_{ij}$  is the interaction effect.

The difference between this model and **Eq. 3-2** in Chapter 3 is the fact that the effects  $a_i$ ,  $b_j$ , and  $ab_{ij}$  are now assumed to be independent, normally distributed random variables with zero means and variances  $\sigma_a^2$ ,  $\sigma_b^2$ , and  $\sigma_{ab}^2$ . Since the variance of a sum of independent random variables is the sum of the variances, the total variance in the response  $Var(y) = \sigma_y^2 = \sigma_a^2 + \sigma_b^2 + \sigma_{ab}^2 + \sigma^2$ .  $\sigma_a^2$  represents the portion of the total variance due to actual differences in part features,  $\sigma_b^2$  is the portion of the variance caused by differences among operators,  $\sigma_{ab}^2$  is the portion of the variance caused by the interaction of operator and part, and  $\sigma^2$  is the portion of the variance caused by replicate measurements or Gage repeatability. The sum of  $\sigma_b^2 + \sigma_{ab}^2$  is the gage reproducibility. The repeatability plus reproducibility  $\sigma_b^2 + \sigma_{ab}^2 + \sigma^2$  is a measure of the variance attributable to measurement error.

### 5.6.1. Estimating Variance Components

For the case with an equal number of replicates per subclass, like the example in **Table 5-5**, it is convenient to use either the method of moments or REML estimators of the variance components. When using the method of moments, it is necessary to know the coefficients of the variance components in the expected mean squares. For a balanced two-factor sampling design, the expected value of the mean squares can be found using the tabular method of Bennett and Franklin (1954) and are shown in **Table 5-6**. The estimates of the variance components can

then be obtained by equating the mean squares in the ANOVA to the expected mean squares and solving for the variance components.

**Table 5-6. Expected Mean Squares in Two-Factor Sampling Design**

Source	df	EMS
A	$a - 1$	$\sigma^2 + r\sigma_{AB}^2 + rb\sigma_A^2$
B	$b - 1$	$\sigma^2 + r\sigma_{AB}^2 + rb\sigma_B^2$
AB	$(a - 1)(b - 1)$	$\sigma^2 + r\sigma_{AB}^2$
Error	$(r - 1)ab$	$\sigma^2$

The R code to produce the ANOVA table is shown below.

```
library(daewr)
modr1 <- aov( y ~ part + oper + part:oper,
               data = gagerr)
summary(modr1)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)						
part	9	1.4489	0.16099	214.18	< 2e-16 ***						
oper	2	0.0297	0.01485	19.76	3.35e-06 ***						
part:oper	18	0.4839	0.02688	35.77	1.87e-15 ***						
Residuals	30	0.0225	0.00075								
	---										
Signif. codes:	0	'***'	0.001	'**'	0.01	'*'	0.05	.	0.1	' '	1

Using the mean squares in the above table, the R code below calculates the variance component estimates using the method of moments.

```
sigma2 <- .000752
sigma2po <- (.026885 - sigma2) / 2
sigma2o <- (.014852 - sigma2 - 2 * sigma2po ) / 20
sigma2p <- (.160991 - sigma2 - 2 * sigma2po ) / 6
cat("Method of Moments Variance Component Estimates",
    "\n",
    "Var(error)=",sigma2," \n",
    "Var(part x oper)=",sigma2po," \n",
    "Var(oper)=", sigma2o," \n",
    "Var(part)=",sigma2p," \n")
```

Method of Moments Variance Component Estimates

Var(error)= 0.000752  
 Var(part x oper)= 0.0130665  
 Var(oper)= -0.00060165  
 Var(part)= 0.022351

For this example the method of moments estimator of  $\sigma_b^2$  is negative. One or two atypical observations can result in an estimated variance component that is too large, too small, or even negative. Graphical techniques, like those described in **Section 5.10**, can often reveal atypical values, and increase the chance of a useful interpretation of the data.

In cases where atypical values are not the cause of negative variance component estimates, the REML estimators avoid the negative values. The R code to produce the REML estimates is shown below. From these results it can be seen that most  $94.3\% = 100 \times \left[ \frac{0.01247}{0.01247 + 0.0007517} \right]$  of the measurement error is due to **reproducibility**. Therefore to reduce measurement error, efforts should be concentrated on better training of operator inspectors rather than investing in more precise gages.

```
library(lme4)
modr2 <- lmer(y ~ 1 + (1|part) + (1|oper) +
(1|part:oper), data = gagerr)
summary(modr2)

Linear mixed model fit by REML ['lmerMod']
Formula: y ~ 1 + (1 | part) + (1 | oper) + (1 | part:oper)
Data: gagerr

REML criterion at convergence: -133.9

Scaled residuals:
    Min      1Q  Median      3Q     Max 
-2.43502 -0.36558 -0.01169  0.38978  1.94190 

Random effects:
Groups   Name        Variance Std.Dev.
part:oper (Intercept) 0.0124650 0.11165
part       (Intercept) 0.0225515 0.15017
oper       (Intercept) 0.0000000 0.00000
Residual             0.0007517 0.02742
Number of obs: 60, groups: part:oper, 30; part, 10; oper, 3

Fixed effects:
            Estimate Std. Error t value
(Intercept)  0.7982    0.0518 15.41
optimizer (nloptwrap) convergence code: 0 (OK)
```

```
boundary (singular) fit: see help('isSingular')
```

### 5.6.2. Confidence Intervals on Variance Components in Two-Factor Designs

When there is an equal number of replicates per subclass, and the normality assumptions hold, the ANOVA mean squares are independently distributed as multiples of chi-square random variables. Therefore confidence intervals on any expected mean square can be obtained similar to line 1 in **Table 5-3**. However, except for  $E(msE) = \sigma^2$ , the expected value of all other mean squares are linear combinations of two or more variance components. Although exact confidence intervals can be obtained on individual variance components in the balanced two-factor design following the formula on line 2 of **Table 5-3**, they are not applicable to all designs or unbalanced data. Burdick and Graybill (1992) show a method of computing approximate confidence intervals that is more generally applicable.

Whenever a variance component can be expressed in the form  $\delta = c_{1E(ms1)} - c_{2E(ms2)}$  where  $ms_1$  and  $ms_2$  are mean squares in the ANOVA table and  $c_1$  and  $c_2$  are positive, the approximate confidence interval shown by Burdick and Graybill is applicable. For example, in the ANOVA table shown above for the gage R&R study,

$$E(msPartOperator) = \sigma^2 + 2\sigma_{ab}^2$$

and

$$E(msOperator) = \sigma^2 + 2\sigma_{ab}^2 + 20\sigma_{ab}^2$$

therefore

$$\delta = 0.05 \times E(msOperator) - 0.05 \times E(msPartOperator) = \sigma_b^2.$$

An approximate  $1 - \alpha$  confidence interval on  $\delta$  is given by:

$$(\delta - \sqrt{V_L}, \delta + \sqrt{V_U}) \quad \text{Eq. 5-11}$$

where  $\delta = c_{1ms1} - c_{2ms2}$ ,  $v_1$  is the degrees of freedom for  $ms1$ ,  $v_2$  is the degrees of freedom for  $ms2$ ,

$$V_L = G_1^2 c_1^2 ms1^2 + H_2^2 c_2^2 ms2^2 + G_{12} c_1 c_2 (ms1)(ms2)$$

$$V_U = H_1^2 c_1^2 ms1^2 + G_2^2 c_2^2 ms2^2 + H_{12} c_1 c_2 (ms1)(ms2),$$

$$G_1 = 1 - \frac{1}{F_{\alpha, v1, \infty}}$$

$$H_2 = \frac{1}{F_{1-\alpha, v2, \infty}} - 1$$

$$G_{12} = \frac{(F_{\alpha, v1, v2} - 1)^2 - G_1^2 F_{\alpha, v1, v2}^2 - H_2^2}{F_{\alpha, v1, v2}}$$

$$H_1 = \frac{1}{F_{1-\alpha, v1, \infty}} - 1$$

$$G_2 = 1 - \frac{1}{F_{\alpha, v2, \infty}}$$

and

$$H_{12} = \frac{(1 - F_{1-\alpha, v1, v2})^2 - H_1^2 F_{1-\alpha, v1, v2}^2 - G_2^2}{F_{1-\alpha, v1, v2}}$$

Although these formulas look formidable, they can be easily evaluated using R. A function `vci`, in the package `daewr`, will compute confidence intervals using these formulas. The inputs to this function are `confl = 1 - alpha`, `c1 = c1`, `ms1 = ms1`, `nu1 = nu1`, `c2 = c2`, `ms2 = ms2`, and `nu2 = nu2`. As an illustration of the use of this function, consider making confidence intervals on the variance component for operator, and the variance component for the part by operator interaction in the gage R&R study data.

To get a confidence interval on the variance component for operator,  $\sigma_b^2$ ,  $ms1 = 0.01485$  with 2 degrees of freedom,  $ms2 = 0.02689$  with 18 degrees of freedom. The function call and output is shown below.

```
library(daewr)
options(digits = 3)
vci(confl = .90, c1 = .05, ms1 = .01485, nu1 = 2,
     c2 = .05, ms2 = .02689, nu2 = 18)
```

`delta=-0.000602 Lower Limit=-0.00158 Upper Limit= 0.00572`

This shows that even though the method of moments estimator for  $\sigma_b^2 = -0.000602$  is negative the upper 90% confidence bound is

positive. The 90% confidence interval for  $\sigma_a^2 = (0.008936 - 0.021895)$  using the same approximation formula.

### 5.6.3. Determining Sample Sizes for Two-Factor Sampling Experiments

In a two-factor sampling experiment, there are three sample sizes to be considered. First the number of levels of factor  $A$ ,  $a$ , second the number of levels of factor  $B$ ,  $b$ , and finally the number of replicates within each cell,  $r$ . The degrees of freedom for the replicate mean square is  $ab(r - 1)$ . By replacing  $t(r - 1)$  by  $ab(r - 1)$  in formula (5.8), it can be used to determine the value of  $ab(r - 1)$  that will result in the desired width of the confidence interval for  $\sigma^2$ . Next, using a rule of thumb like that expressed in **Section 5.5.4** the levels of factor  $A$  and factor  $B$  can be determined.

### 5.6.4. Two-Factor Studies with Unequal Replication

When there are an unequal number of replicates in each subclass, the method of moments or analysis of variance estimators of the variance components are more difficult to compute. The coefficients of the variance components in the expected mean squares are no longer integer functions of the number of levels of the factors and number of replicates as shown in **Table 5-6**. Instead, they must be computed using Hartley's method as described in Section. Additionally, the variance component estimates obtained solving the equations that equate the mean squares to their expected values are not unique. The estimates will be different depending on whether the sequential or type III sums of squares are used in the equations. On the other hand, the maximum likelihood estimators and REML estimators are unique and may be preferred in this situation.

As an example of computing the variance components using the REML method, consider the data in **Table 5-7** taken from a sampling study to estimate the sources of variability in an inter-laboratory assay of calcium in blood serum that was shown by Rao and Rao (1997).

**Table 5-7. Calcium in Blood Serum Solutions with Unknown Concentrations**

---

Standard Solution

---

Laboratory	1	2	3	4
A	87	92	179	177
	84	83	173	
	80	76	166	
B	80	69	138	151
	70	46	138	
	132			
C	70	67	173	176
	60	63	166	
	44	48		

The code to open the data frame and compute the variance components using the `lmer` function in the `lme4` package are similar to those shown in the last section for balanced data. The code and results of the analysis are shown on the next page. The estimated variance component for the lab by solution interaction is near zero, but the method of moments estimator of the same component would be negative. It can be seen that the majority of the variability in the assay for calcium in blood serum is due to differences among the standard solutions and among the repeat analyses using the same solution in the same lab. A very small proportion of the variability is caused by differences in labs.

```
library(daewr)
library(lme4)
rmod3 <- lmer( calcium ~ 1 + (1|lab) + (1|sol) +
    (1|lab:sol),
  data = blood)
summary(rmod3)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: calcium ~ 1 + (1 | lab) + (1 | sol) + (1 | lab:sol)
Data: blood
```

```
REML criterion at convergence: 265
```

```
Scaled residuals:
```

Min	1Q	Median	3Q	Max
-1.6405	-0.5553	-0.0612	0.4173	2.3327

```
Random effects:
```

```

Groups   Name      Variance Std.Dev.
lab:sol (Intercept) 1.27e-07 3.56e-04
sol      (Intercept) 1.49e+03 3.86e+01
lab      (Intercept) 2.80e+01 5.29e+00
Residual                     1.05e+03 3.24e+01
Number of obs: 27, groups: lab:sol, 12; sol, 4; lab, 3

```

**Fixed effects:**

	Estimate	Std. Error	t value
(Intercept)	103.2	20.7	4.99
optimizer (nloptwrap) convergence code: 0 (OK)			
boundary (singular) fit: see help('isSingular')			

Since there are only three labs, any asymptotic confidence interval for  $\sigma_a^2$  would not be accurate. With unequal replication the ANOVA mean squares are no longer distributed as multiples of chi-square random variables, so the confidence intervals using line 2 in Table 5.3 are also not appropriate.

Approximate confidence intervals can still be obtained if unweighted mean squares are substituted for the mean squares in Eq. 5-11. The unweighted mean squares for factor A, B, and the interaction can be obtained by performing an ANOVA on the cell means. Table 5.8 is a symbolic representation of the ANOVA with **unweighted mean squares** (and their expected values) from a two-factor sampling design.

**Table 5-8. Symbolic ANOVA with Unweighted Mean Squares**

Source	df	MS	EMS
Factor A	$a - 1$	$msA_U$	$\sigma^2 + c\sigma_{ab}^2 + bc\sigma_a^2$
Factor B	$b - 1$	$msB_U$	$\sigma^2 + c\sigma_{ab}^2 + ac\sigma_b^2$
Interaction AB	$(a - 1)(b - 1)$	$msAB_U$	$\sigma^2 + c\sigma_{ab}^2$
Error	$\sum_i \sum_j r_{ij} - ab$	$msE$	$\sigma^2$

The symbol  $\bar{c} = ab / \sum_i \sum_j (1/r_{ij})$  and  $r_{ij}$  is the number of replicates in the  $ij$ th cell. From this table it can be seen that  $\hat{\sigma}_a^2 = (1/(b\bar{c}))maA_U - (1/bc)msAB_U$ , which is of the form  $\delta = c_{1E(ms1)} - c_{2E(ms2)}$  shown in Section 5.5.2.

For the data in Table 5.7, the unweighted mean squares are obtained by the R code below. In that code, the first statement uses the [tapply](#)

function to calculate the cell means of the response `calcium` in the data frame `blood`. Next, factors are created that are the same length as the vector of cell means, and the `aov` function is used to make the unweighted means ANOVA table.

```
cellmeans <- tapply( blood$calcium, list(blood$lab,
    blood$sol), mean)
dim(cellmeans) <- NULL
Lab <- factor(rep(c("A", "B", "C"), 4))
Solution <- factor(rep(c(1, 2, 3, 4), each = 3))
mod2 <- aov( cellmeans ~ Lab + Solution + Lab:Solution)
summary(mod2)
```

	Df	Sum Sq	Mean Sq
Lab	2	826	413
Solution	3	15035	5012
Lab:Solution	6	625	104

In the results, the unweighted means square for factor  $A$ ,  $maA_U = 413$ , and the unweighted mean square for factor  $B$ ,  $msAB_U = 104$ . The factor

$$\bar{c} = \frac{ab}{\sum_i \sum_j \left(\frac{1}{r_{ij}}\right)} \\ = \frac{(3)(4)}{\left(\frac{1}{1} + \frac{1}{3} + \frac{1}{3} + \frac{1}{3} + \frac{1}{1} + \frac{1}{2} + \frac{1}{2} + \frac{1}{3} + \frac{1}{1} + \frac{1}{3} + \frac{1}{3} + \frac{1}{2}\right)} \\ = 1.846$$

and  $c_1 = c_2 = \frac{1}{b\bar{c}} = \frac{1}{4(1.846)} = 0.13541$ . Thus the inputs needed for the `vci` function to calculate a 90% confidence interval for  $\sigma_a^2$  is shown in the code below.

```
library(daewr)
vci(confl = .90, c1 = .1354166, ms1 = 413, nu1 = 2,
c2 = .1354166, ms2 = 104, nu2 = 6)
```

`delta= 41.8 Lower Limit= 4.14 Upper Limit= 517`

and the resulting confidence interval is  $(4.138, 516.772)$  is very wide, again due to the fact that there were only three labs in the data

## **5.7. Nested Sampling Experiments (NSE)**

Many sampling experiments with more than one factor use nested factors or a hierarchical design. The levels of a nested factor are physically different depending on the level of factor it is nested within. That was not the case for the factors described in the last section. For example, in the gage R&R study, each operator measured each part; therefore, the operator number was uniquely defined and referred to the same operator regardless of which part he measured. We would call the operator and the part in the gage R&R study crossed factors. To change the design so that the operator was a nested factor, consider an experiment where  $n$  parts were selected and each part was measured by two operators, although it does not have to be the same two operators measuring each part. This might be a more convenient way to conduct the sampling experiment if parts were selected over a long period of time and the same operators were not always present to make the measurements. The fact that the operators differ depending upon the part number being measured makes the operator a nested factor (nested within part). The first operator measuring the first part is not physically the same person as the first operator measuring a subsequent part. Another common example where a nested design occurs is when the measurements are destructive. In that case, each operator must measure a different set of parts, and the part becomes the nested factor (nested within operator) because the first part measured by the first operator is not physically the same as the first part measured by subsequent operators. This type of sampling design is called a **nested sampling experiment or NSE**.

One example where we have already seen nested factors is in the term  $\varepsilon_{ij}$  in the models we have used thus far. It represents the effect of the  $j$ th replicate experimental unit, and since different experimental units are used for each factor level or combination of factor levels, the experimental unit is always nested within another factor level or cell in the design.

When two factors are crossed factors we can include their interaction in the model to represent the extra variability caused by changes in the level of one factor between levels of the other factor. However, if a

factor is nested within another factor we cannot include an interaction between them because the nested factor includes the degrees of freedom that could be taken by the interaction. The model for a two-stage nested design with factor B nested within factor A is written as:

$$y_{ijk} = \mu + a_i + b_{(i)j} + \varepsilon_{ijk} \quad \text{Eq. 5-12}$$

and if there is an equal number of replicates,  $r$ , per cell, the ANOVA table for the nested model can be represented as:

**Table 5-9. Symbolic ANOVA for Two-Factor Nested Design**

Source	df	MS	EMS
Factor A	$a - 1$	$msA$	$\sigma^2 + r\sigma_b^2 + \sigma_{ab}^2$
Factor B	$a(b - 1)$	$msB$	$\sigma^2 + r\sigma_b^2$
Error	$ab((r - 1))$	$msE$	$\sigma^2$

Here it can be seen that the degrees of freedom for factor B,

$$a(b - 1) = (b - 1) + (a - 1)(b - 1)$$

is equal to the degrees of freedom for a crossed factor plus the degrees for the interaction AB. The expected mean squares in the table can again be obtained using the tabular method of Bennett and Franklin (1954).

Nested or hierarchical designs can easily be extended to include several stages or factors. For example, Table 5.10 shows the results of a four-stage nested sampling study on the variability of properties of crude rubber, taken from Bennett and Franklin (1954). In this study a sample of four batches of rubber was taken from each of four suppliers. Since the first batch obtained from the first supplier is not physically the same as the first batch taken from the second supplier, batch is nested within supplier. Next, two sample mixes were made from each batch, and since the two sample mixes for one batch are physically different than the sample mixes for any other batch, the sample mix is nested within the batch. Finally, three replicate tests were performed on each sample mix to determine the elasticity.

The model for the data can be written as:

$$y_{ijkl} = \mu + a_i + b_{(i)j} + c_{(ij)k} + \varepsilon_{ijkl}$$

where  $y_{ijkl}$  is the  $l$ th elasticity determination made from the  $k$ th sample mix, taken from the  $j$ th batch from the  $i$ th supplier,  $a_i$  is the random supplier effect,  $b_{(i)j}$  is the random batch effect,  $c_{(ij)k}$  is the random sample mix effect, and  $\varepsilon_{ijkl}$  is the random replicate determination effect,  $i = 1, \dots, 4$ ,  $j = 1, \dots, 4$ ,  $k = 1, \dots, 2$ , and  $l = 1, \dots, 3$ .

```
# This model can be written in the notation of R
function aov as
mod2 <- aov( elasticity ~ supplier + supplier:batch +
    supplier:batch:sample, data = rubber)
# or in the notation of the R function lmer in the
package lme4 as
library(lme4)
modr3 <- lmer( elasticity ~ 1 + (1|supplier) +
    (1|supplier:batch) + (1|supplier:batch:sample),
    data = rubber)
```

The ‘:’ notation indicates that the batch is nested in the supplier, and so forth. The variance components  $\sigma_a^2$ ,  $\sigma_b^2$ ,  $\sigma_c^2$ , and  $\sigma^2$  can be estimated using the method of moments or REML.

In order to increase confidence in estimates of variance components, the number of levels of a random factor should be increased. However, in hierarchical designs with several stages, increasing the number of levels of the topmost factor greatly increases the overall sample size, even if all the other nested factors have only two levels. For example, in the design shown in **Table 5-10**, if the number of suppliers was increased from 4 to 20 in order to get a more precise estimate of  $\sigma_a^2$ , the number of determinations that would have to be made would increase from the 96 to 480 (see **Table 5-10**). Even if the number of batches per supplier and the number of sample mixes per batch and determinations per mix were reduced to 2 each, there would still be  $20 \times 2 \times 2 \times 2 = 160$  determinations. If the sampling study had been done in this way there would be  $a - 1 = 19$  degrees of freedom for the supplier effect,  $a(b - 1) = 20(2 - 1) = 20$  degrees of freedom for the batch effect,  $ab(c - 1) = 20 \times 2(2 - 1) = 40$  degrees of freedom for the sample effect, and  $abc(r - 1) = 20(2)(2)(2 - 1) = 80$  degrees of freedom for the random replicate effect. Therefore the majority of the 160 observations are used to increase the precision of the bottom two

variance components  $\sigma^2$  and  $\sigma_c^2$ . For this reason, balanced hierarchical designs are usually not recommended if there are more than three stages or sources of variability being studied. Staggered nested designs presented in the next section allow the convenience of nested factors in sampling studies but allow the various variance components to be estimated with more uniform precision.

**Table 5-10. Modulus of Elasticity at 700% Elongation of 96 Prepared Specimens of Smoked Sheet Rubber**

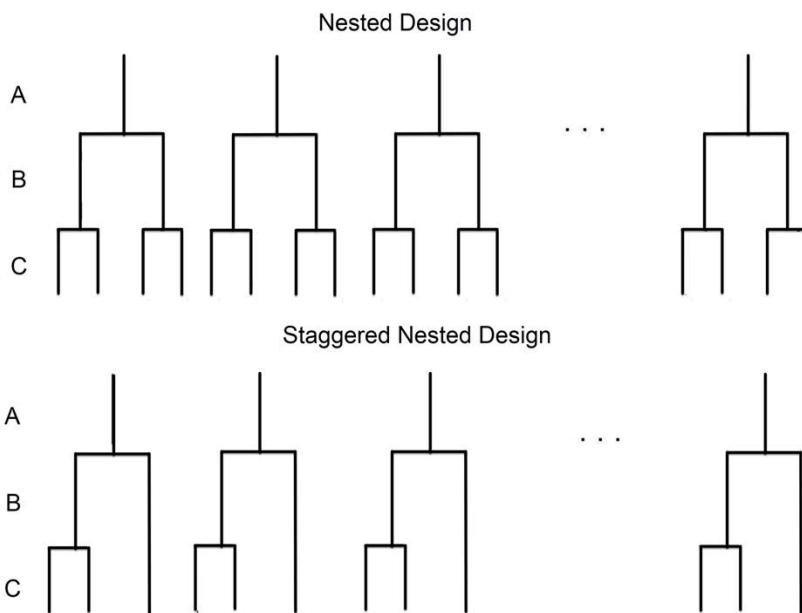
Supplier	A		B		C		D	
	Sample	Mix	Sample	Mix	Sample	Mix	Sample	Mix
	1	2	1	2	1	2	1	2
Batch I	211	171	196	196	200	240	323	262
	215	198	186	210	221	229	279	234
	197	268	190	156	198	217	251	249
	229	234	209	200	191	196	255	249
Batch II	196	210	193	186	189	198	235	247
	200	226	204	196	186	175	223	239
	204	225	204	174	211	196	228	262
Batch III	221	215	165	172	197	184	250	227
	238	196	194	171	210	190	260	272
	229	248	198	202	196	180	273	273
Batch IV	250	249	209	211	197	166	241	256
	238	249	221	204	186	172	221	230

## 5.8. Staggered Nested Designs

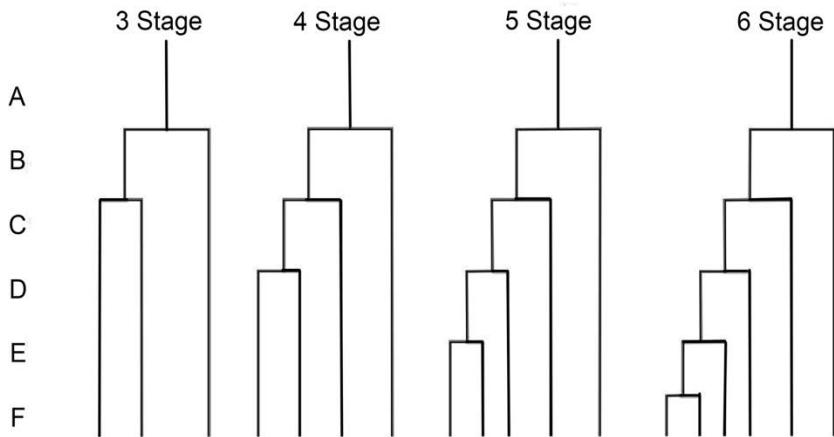
Staggered nested sampling experiments, or **SNSE**, were developed independently by Bainbridge (1965) and Prairie and Anderson (1962). In a completely nested design as discussed in the last section, each level of the topmost factor leads down into two (or more) levels of each succeeding factor or stage. In a staggered nested design, on the other hand, only one of the two levels of the succeeding factor leads to the next two-level stage. **Figure 5-2** illustrates the difference between a nested and staggered nested design for three stages. If there are  $a$  levels of the topmost factor, the nested design requires  $4a$  total observations while the staggered nested design only requires  $3a$  observations. The

savings in observations are multiplied as the number of stages in a nested design increases. **Figure 5-3** shows the schematic for staggered nested designs from three through six stages.

While the information or degrees of freedom available for estimating variance components in a completely nested design is concentrated in the lower tier factors, the information is balanced in a staggered nested design. **Table 5-11** compares the degrees of freedom distribution between a staggered nested design and a completely nested design where each factor except the topmost has only two levels.



**Figure 5-2. nested design increases. Figure 5.2 shows the schematic for staggered nested designs from three through six stages.**



**Figure 5-3. Staggered Nested Designs for 3 to 6 Stages**

Although the expected mean square coefficients for the staggered nested designs can be determined using **Hartley's method** (Hartley, 1967), his method is not available in any R function. Fortunately, the EMS coefficients have been tabulated by Nelson (1983) and are shown in **Table 5-11**. They are needed when calculating the variance components estimators by the method of moments. They are needed when calculating the variance components estimators by the method of moments.

**Table 5-11. Comparison of Degrees of Freedom between Staggered Nested and Nested Designs**

Source	Staggered	
	Nested df	Nested df
A	$a - 1$	$a - 1$
B in A	$a$	$a$
C in B	$a$	$2a$
D in C	$a$	$4a$
E in D	$a$	$8a$
F in E	$a$	$16a$

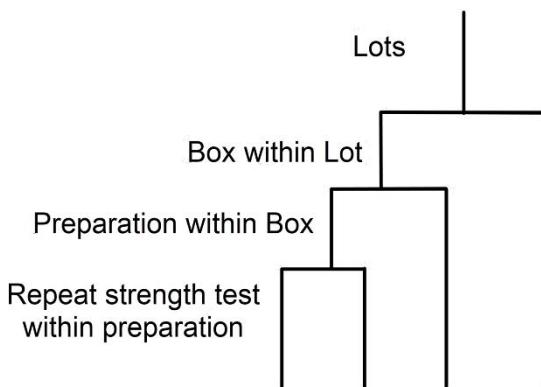
**Table 5-12. Expected Mean Square Coefficients for Staggered Nested Designs**

Stage	Term	EMS
3	A	$\sigma_C^2 + \left(\frac{5}{3}\right)\sigma_B^2 + 3\sigma_A^2$
	B	$\sigma_C^2 + \left(\frac{4}{3}\right)\sigma_B^2$
	C	$\sigma_C^2$
4	A	$\sigma_D^2 + \left(\frac{3}{2}\right)\sigma_C^2 + \left(\frac{5}{2}\right)\sigma_B^2 + 4\sigma_A^2$
	B	$\sigma_D^2 + \left(\frac{7}{6}\right)\sigma_C^2 + \left(\frac{3}{2}\right)\sigma_B^2$
	C	$\sigma_D^2 + \left(\frac{4}{3}\right)\sigma_C^2$
	D	$\sigma_D^2$
5	A	$\sigma_E^2 + \left(\frac{7}{5}\right)\sigma_D^2 + \left(\frac{17}{5}\right)\sigma_B^2 + 5\sigma_A^2$
	B	$\sigma_E^2 + \left(\frac{11}{10}\right)\sigma_D^2 + \left(\frac{13}{10}\right)\sigma_C^2 + \left(\frac{8}{5}\right)\sigma_B^2$
	C	$\sigma_E^2 + \left(\frac{7}{6}\right)\sigma_D^2 + \left(\frac{3}{2}\right)\sigma_C^2$
	D	$\sigma_E^2 + \left(\frac{4}{3}\right)\sigma_D^2$
	E	$\sigma_E^2$
6	A	$\sigma_F^2 + \left(\frac{4}{3}\right)\sigma_E^2 + 2\sigma_B^2 + 3\sigma_C^2 + \left(\frac{13}{12}\right)\sigma_B^2 + 6\sigma_A^2$
	B	$\sigma_F^2 + \left(\frac{14}{5}\right)\sigma_E^2 + \left(\frac{6}{5}\right)\sigma_D^2 + \left(\frac{7}{5}\right)\sigma_C^2 + \left(\frac{5}{3}\right)\sigma_B^2$
	C	$\sigma_F^2 + \left(\frac{11}{10}\right)\sigma_E^2 + \left(\frac{13}{10}\right)\sigma_D^2 + \left(\frac{8}{5}\right)\sigma_C^2$
	D	$\sigma_F^2 + \left(\frac{7}{6}\right)\sigma_E^2 + \left(\frac{3}{2}\right)\sigma_D^2$
	E	$\sigma_F^2 + \left(\frac{4}{3}\right)\sigma_E^2$
	F	$\sigma_F^2$

Mason et al. (1989) described a study where a staggered nested design was used to estimate the sources of variability in a continuous polymerization process. In this process polyethylene pellets are produced in lots of one hundred thousand pounds. A four-stage design was used to partition the source of variability in tensile strength between lots, within lots and due to the measurement process. Thirty lots were sampled at random. Lot represented the topmost factor or

source of variability *A*. From each lot two boxes of pellets were randomly selected. This represented the second stage or source of variability *B*. From the first box selected from each lot, two preparations were made for strength testing, but from the second box selected from each lot only one preparation was made. This represented the third stage or source of variability *C*. Finally, two repeat strength tests were made from the first preparation from box one, while only one strength test was made from the other three preparations.

This sampling scheme is diagramed in **Table 5-4**, and the data is shown in **Table 5-13**.



**Figure 5-4. Diagram of Sampling Scheme for Polymerization Study**

The R code to open the data frame and compute the ANOVA table for use in estimating the variance components by the method of moments is shown below.

```

library(daewr)
mod2 <- aov(strength ~ lot + lot:box + lot:box:prep,
            data = polymer)
summary(mod2)
  
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)					
lot	29	856	29.52	45.55 < 2e-16	***					
lot:box	30	50	1.67	2.58	0.00577 **					
lot:box:prep	30	68	2.28	3.52	0.00046 ***					
Residuals	30	19	0.65							
---										
Signif. codes:	0	'***'	0.001	'**'	0.01	'*' 0.05	'.'	0.1	'.'	1

The method of moments estimators of the variance components are found by equating the mean squares in the ANOVA table above to the expected mean squares in **Table 5-12** and solving. The results are shown on the next page. Once again the method of moments procedure produces a negative estimate for one of the sources.

**Table 5-13. Data from Polymerization Strength Variability Study**

Lot	Box 1 Preparation		Box 2 Preparation	
	test 1	test 2	test 1	test 1
1	9.76	9.24	11.91	9.02
2	10.65	7.77	10.00	13.69
3	6.50	6.26	8.02	7.95
4	8.08	5.28	9.15	7.46
5	7.84	5.91	7.43	6.11
6	9.00	8.38	7.01	8.58
7	12.81	13.58	11.13	10.00
8	10.62	11.71	14.07	14.56
9	4.88	4.96	4.08	4.76
10	9.38	8.02	6.73	6.99
11	5.91	5.79	6.59	6.55
12	7.19	7.22	5.77	8.33
13	7.93	6.48	8.12	7.43
14	3.70	2.86	3.95	5.92
15	4.64	5.70	5.96	5.88
16	5.94	6.28	4.18	5.24
17	9.50	8.00	11.25	11.14
18	10.93	12.16	9.51	12.71
19	11.95	10.58	16.79	13.08
20	4.34	5.45	7.51	5.21
21	7.60	6.72	6.51	6.35
22	5.12	5.85	6.31	8.74
23	5.28	5.73	4.53	5.07
24	5.44	5.38	4.35	7.04
25	3.50	3.88	2.57	3.76
26	4.80	4.46	3.48	3.18
27	5.35	6.39	4.38	5.50
28	3.09	3.19	3.79	2.59
29	5.30	4.72	4.39	6.13
30	7.09	7.82	5.96	7.14

$$\sigma_R^2 = 0.648$$

$$\sigma_P^2 = \frac{(2.281 - 0.648)}{\binom{4}{3}} = 1.22475$$

$$\sigma_B^2 = \frac{\left(1.670 - \left[0.648 + \left(\frac{7}{6}\right) 1.22475\right]\right)}{\binom{3}{2}} = -0.27125$$

$$\sigma_L^2 = \frac{\left(29.516 - \left[0.648 + \left(\frac{3}{2}\right) (1.22475) + \left(\frac{5}{2}\right) (-0.27125)\right]\right)}{4} = 6.92725$$

To estimate the variance components using the REML method, the `lmer` function from the `lme4` package is used as shown below. The results follow the code. The REML estimator of the variance component for box within lot is essentially zero rather than the negative value obtained from the method of moments.

```
library(lme4)
modr3 <- lmer( strength ~ 1 + (1|lot) + (1|lot:box) +
  (1|lot:box:prep), data = polymer)
summary(modr3)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: strength ~ 1 + (1 | lot) + (1 | lot:box) + (1 |
lot:box:prep)
Data: polymer
```

```
REML criterion at convergence: 469
```

```
Scaled residuals:
```

Min	1Q	Median	3Q	Max
-2.1896	-0.4119	-0.0206	0.3826	1.7703

```
Random effects:
```

Groups	Name	Variance	Std.Dev.
lot:box:prep	(Intercept)	1.03e+00	1.014671
lot:box	(Intercept)	2.37e-08	0.000154
lot	(Intercept)	7.24e+00	2.691221
Residual		6.57e-01	0.810433

```
Number of obs: 120, groups: lot:box:prep, 90; lot:box, 60;
lot, 30
```

**Fixed effects:**

	Estimate	Std. Error	t value
(Intercept)	7.221	0.509	14.2

From these results,

$$81\% = \left( 100 \times \frac{7.2427}{7.2427 + 1.0296 + 0.6568} \right)$$

we see that 81% of the total variation is due to variability among lots, while the within lot or box-to-box variability is negligible. Therefore, if the manufacturers would like to decrease the variability in tensile strength, they should focus on reducing the variation on influential factors that change between lots.

While no exact closed form confidence intervals have been developed for variance components estimated from data in staggered nested designs, when the number of levels of the topmost factor is greater than 30, approximate asymptotic estimates can be created using the likelihood profile method with the R package [lme4](#) as discussed in **Section 5.5.3**. Bayesian interval estimates of variance components are also useful for these designs (see Lawson, (2008)).

To determine the sample size for a staggered nested design, follow the procedure outlined in **Section 5.5.4** for determining the sample size for estimating the replicate variance in a one-factor sampling design. Since the degrees of freedom for all factors or stages in the staggered nested design are near equal, following this procedure will give you approximately the same precision on all variance components.

## 5.9. Designs with Fixed and Random Factors

In some cases where fixed treatment factors are being studied in an experimental design like those discussed in Chapters 2 through 4, random factors are also introduced in the model by the way the experiment is conducted. For example, consider the data in **Table 5-14** that resulted from an experiment comparing different formulations and methods of applying a pesticide to the leaves of cotton plants. The goal was to increase the amount of active pesticide remaining on cotton plant leaves one week after application. The pesticide being studied degrades

in sunlight and a certain additive to the formulation retards this process. Different application techniques may differ in the amount of pesticide delivered to the plant leaves. The treatment factors in this experiment were two different formulations of the pesticide and two different application methods, resulting in a  $2^2$  factorial experiment.

The experimental unit was a 20' row of cotton plants called a plot because this was a convenient area within which the application of pesticide could be controlled. Eight plots were selected and two were randomly assigned to each of the four treatment combinations, resulting in two replicates per treatment combination. One week after application, the experimenters were ready to determine the pesticide residue remaining on the plant leaves. However, there was too much plant material in an entire plot to send to the lab for analysis. Therefore, two samples of leaves in an amount convenient for laboratory analysis of pesticide residues were selected from each plot. Each sample was sent to the lab resulting in the data shown in **Table 5-14**.

Formulation, application technique, and their interaction are fixed factors because the experimenters were interested in comparing the average response between levels of these factors. The plot, on the other hand, is a random factor that represents differences in experimental units. It is nested within the combinations of formulation by application technique. There is no interest in comparing experimental units within each combination of formulation and application. Instead, multiple plots per treatment combination were included in the design so that the variance caused by differing plots could be estimated and used to judge significance of formulation and application effects. The replicate samples taken from each plot were for convenience in conducting the experiment. They would be classified as sub-samples or observational units defined in Chapter 1.

The simplest way of analyzing the data would be to average the two sub-samples and proceed as illustrated in **Section 3.5**. However, if the sub-samples can be assumed independent and it is desirable to include all the data (shown in **Table 5-14**) in the analysis, then an additional term for sample must be included in the model. Sample is another random effect since there is no specific interest in comparing the response

between the two samples from each plot.

**Table 5-14. Pesticide Residue on Cotton Plants**

Formulation	Technique	Plot	Sample	
			1	2
A	1	1	0.237	0.252
A	1	2	0.281	0.274
B	1	1	0.247	0.294
B	1	2	0.321	0.267
A	2	1	0.392	0.378
A	2	2	0.381	0.346
B	2	1	0.351	0.362
B	2	2	0.334	0.348

The model for the data can be written in the form

$$y_{ijkl} = \mu + \alpha_i + \beta_j + \alpha\beta_{ij} + p_{(ij)k} + \varepsilon_{ijkl} \quad \text{Eq. 5-13}$$

here  $y_{ijkl}$  is the pesticide residue found on the  $l$ th sample taken from the  $k$ th plot, treated with formulation level  $i$  and application technique  $j$ . In general  $i = 1, \dots, a$ ,  $j = 1, \dots, b$ ,  $k = 1, \dots, r$ , and  $l = 1, \dots, s$ . In this specific example,  $a = 2$ ,  $b = 2$ ,  $r = 2$ , and  $s = 2$ ,  $\alpha_i$  is the formulation effect,  $\beta_j$  is the application effect,  $\alpha\beta_{ij}$  is the interaction effect,  $p_{(ij)k}$  is the random plot effect, and  $\varepsilon_{ijkl}$  is the random sample effect.

The model can be written in matrix notation as:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\varepsilon} \quad \text{Eq. 5-14}$$

where

$$y = \begin{pmatrix} y_{1111} \\ y_{1112} \\ y_{1121} \\ y_{1122} \\ y_{2111} \\ y_{2112} \\ y_{2121} \\ y_{2122} \\ y_{1211} \\ y_{1212} \\ y_{1221} \\ y_{1222} \\ y_{2211} \\ y_{2212} \\ y_{2221} \\ y_{2222} \end{pmatrix}, X = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}, \beta = \begin{pmatrix} \mu \\ \alpha_1 \\ \alpha_2 \\ \beta_1 \\ \beta_2 \\ \alpha\beta_{11} \\ \alpha\beta_{21} \\ \alpha\beta_{12} \\ \alpha\beta_{22} \end{pmatrix}$$

$$Z = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \gamma = \begin{pmatrix} p_{(11)1} \\ p_{(11)2} \\ p_{(21)1} \\ p_{(21)2} \\ p_{(12)1} \\ p_{(12)2} \\ p_{(22)1} \\ p_{(22)2} \end{pmatrix}, \varepsilon = \begin{pmatrix} \varepsilon_{1111} \\ \varepsilon_{1112} \\ \varepsilon_{1121} \\ \varepsilon_{1122} \\ \varepsilon_{2111} \\ \varepsilon_{2112} \\ \varepsilon_{2121} \\ \varepsilon_{2122} \\ \varepsilon_{1211} \\ \varepsilon_{1212} \\ \varepsilon_{1221} \\ \varepsilon_{1222} \\ \varepsilon_{2211} \\ \varepsilon_{2212} \\ \varepsilon_{2221} \\ \varepsilon_{2222} \end{pmatrix}$$

$\beta$  represents the vector of fixed effects, while  $\gamma$  and  $\varepsilon$  represent the vectors of random effects. The assumption of independent and normally distributed random effects can be expressed by  $\gamma \sim MVN(\mathbf{0}, \sigma_p^2 I_{abrs})$ , and  $\varepsilon \sim MVN(\mathbf{0}, \sigma^2 I_{abrs})$ ; therefore,  $y \sim MVN(X\beta, V)$ , where  $V = Z(\sigma_p^2 I_{abrs})Z' + \sigma^2 I_{abrs}$ . The least squares estimator of  $\beta$  would be  $\hat{\beta} = (X'V^{-1}X)^{-1}X'V^{-1}y$ , where  $^{-1}$  refers to a generalized inverse; however,

$\sigma_p^2$ ,  $\sigma^2$  are unknown, and therefore  $V$  is unknown.

R function `aov` solves the problem by treating both  $\beta$  and  $\gamma$  as fixed. The ANOVA sums of squares for each term in the model are computed like those shown in **Table 3-2**, with an additional term called `Residuals`, which represents the sub-samples.

The commands to open the data frame with the data from **Table 5-14** and create the ANOVA table are shown below.

```
library(daewr)
mod4 <- aov( residue ~ form + tech + form:tech +
plot:form:tech, data = pesticide)
summary(mod4)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)						
form	1	0.00002	0.00002	0.040	0.8455						
tech	1	0.03231	0.03231	72.434	2.79e-05 ***						
form:tech	1	0.00219	0.00219	4.900	0.0578 .						
form:tech:plot	4	0.00234	0.00059	1.314	0.3432						
Residuals	8	0.00357	0.00045								
	---										
Signif. codes:	0	'***'	0.001	'**'	0.01	'*'	0.05	'.'	0.1	' '	1

The  $F$ - and  $P$ -values are incorrect in this table because they use the sub-sample or Residuals term as the denominator of the  $F$ -tests. The expected mean squares for this balanced design can be found using Bennett and Franklin's (1954) tabular method and are shown in **Table 5-15**. In this table  $\sigma^2$  and  $\sigma_P^2$  are the variances of the random effects  $\varepsilon_{ijkl}$  and  $p_{(ij)k}$  in **Eq. 5-13**. The  $\tau^2$  terms represent quadratic functions of the fixed effects ( $\alpha_i$ ,  $\beta_j$ , and  $\alpha\beta_{ij}$ ).

**Table 5-15. Expected Mean Squares for Two-Factor Design with Sub-Samples**

Source	Df	EMS
$A$	$a - 1$	$\sigma^2 + r\sigma_P^2 + srb\tau_A^2$
$B$	$b - 1$	$\sigma^2 + r\sigma_P^2 + sra\tau_B^2$
$AB$	$(a - 1)(b - 1)$	$\sigma^2 + r\sigma_P^2 + srt_{AB}^2$
Plot	$(r - 1)ab$	$\sigma^2 + r\sigma_P^2$
Sub-Sample	$(s - 1)rab$	$\sigma^2$

Since the expected mean squares for formulation, application, and their interaction all contain  $\sigma^2 + 2\sigma_p^2$  in addition to a quadratic form involving the fixed effects, the correct mean square to use for the denominator of the  $F$ -ratio for testing these fixed effects is the plot term or `form:tech:plot` whose expectation is  $\sigma^2 + 2\sigma_p^2$ . The expected value of the error or `Residuals` mean square is  $\sigma^2$ , and it is too small for use as a denominator in the  $F$ -ratio for testing the fixed effects. However, this is what the `aov` function ANOVA table uses.

The correct F-values and P-values for the ANOVA are:

$$\text{form: } F_{1,4} = 0.00002 \sim 0.00059 = 0.03, P = 0.8709,$$

$$\text{tech: } F_{1,4} = 0.03231 \sim 0.00059 = 54.76, P = 0.0018,$$

$$\text{form:tech } F_{1,4} = 0.00219 \sim 0.00059 = 3.71, P = 0.1264,$$

where the  $P$ -values can be obtained using the R probability functions (i.e.,  $1 - pf(0.03, 1, 4)$ ). Now it can be seen that the only significant term is the application technique.

In many cases where observational units are different than the experimental units, the observational units will not be independent. In this example, common application of the pesticide to each plot might induce a correlation,  $E(\varepsilon_{ijkl} \times \varepsilon_{ijkl'}) = \rho$ , between sub-samples from the same plot. Even though the independence assumption is violated, Casella (2008) shows the  $F$ -test on fixed effects using the plot mean square as the denominator is still valid.

In this example where the application technique was found to be significant, we should look at the marginal means for application (since there is equal replication) in order to determine which application technique was best. The marginal means can be expressed as  $\bar{y}_{1..}$  and  $\bar{y}_{2..}$ . The expected value  $E(\bar{y}_{j..}) = \mu + \beta_j + \bar{\alpha}\bar{\beta}_j$ . The variance  $Var(\bar{y}_{j..}) = \frac{\sigma_p^2}{ar} + \frac{\sigma^2}{ars}$ , and the variance of the difference in two marginal means would be  $2\left(\frac{\sigma_p^2}{ar} + \frac{\sigma^2}{ars}\right)$ . However, the standard error of the difference in means reported by the estimable function in the R package `gmodels` is  $\sqrt{\frac{2\sigma^2}{ars}} = 0.01056$ , as shown below.

```

c1 <- c(-.5, .5)
mod4 <- aov( residue ~ form + tech + form:tech +
    plot:form:tech,
    contrasts = list( form = c1, tech = c1,
    plot = c1 ), data = pesticide)
c <- ('application effect' = c(0,0,1,0,0,0,0,0))
library(gmodels)
estimable(mod4,c)

```

	Estimate	Std. Error	t value	DF	Pr(> t )
(0 0 1 0 0 0 0 0)	0.089875	0.0105601	8.510813	8	2.789427e-05

The R `aov` function does not estimate the variance component for  $\sigma_p^2$ , since `plot` is treated as a fixed effect in the calculations. A similar problem occurs with all the standard errors of estimable functions of fixed effects in the model when the estimable function in `gmodels` is operating on an object created by `aov`.

When there are both fixed and random effects in the model, due to the way the experiment was conducted, we call it a mixed model and the `lmer` function in the R package `lme4` is a better option for analysis. By default, `lmer` uses the REML method to estimate the variance components for the random effects in the model, and then estimates the fixed effects using the formula  $\hat{\beta} = (\mathbf{X}'\hat{\mathbf{V}}^{(-1)}\mathbf{X})^{-1}\mathbf{X}'\hat{\mathbf{V}}^{(-1)}\mathbf{y}$ . The correct standard errors for estimable functions of fixed effects are produced by `lmer`.

The commands to run `lmer` with the data in **Table 5-14** are shown below.

```

library(lme4)
c1 <- c( -.5, .5 )
mod5 <- lmer( residue ~ 1 + form + tech + form:tech +
    (1|plot:form:tech),
    contrasts = list( form = c1, tech = c1 ),
    data = pesticide)
summary(mod5)

```

Linear mixed model fit by REML ['lmerMod']

```
Formula: residue ~ 1 + form + tech + form:tech + (1 |  
plot:form:tech)  
Data: pesticide
```

```
REML criterion at convergence: -51.9
```

```
Scaled residuals:
```

Min	1Q	Median	3Q	Max
-1.53621	-0.67181	0.05407	0.57711	1.70193

```
Random effects:
```

Groups	Name	Variance	Std.Dev.
plot:form:tech	(Intercept)	6.994e-05	0.008363
Residual		4.461e-04	0.021120

Number of obs: 16, groups: plot:form:tech, 8

```
Fixed effects:
```

	Estimate	Std. Error	t value
(Intercept)	0.316562	0.006052	52.311
form1	-0.002125	0.012103	-0.176
tech1	0.089875	0.012103	7.426
form1:tech1	-0.046750	0.024206	-1.931

```
Correlation of Fixed Effects:
```

	(Intr)	form1	tech1
form1	0.000		
tech1	0.000	0.000	
form1:tech1	0.000	0.000	0.000

In these results, the estimates of the variance components  $\hat{\sigma}_p^2 = 0.00006994$  and  $\hat{\sigma}^2 = 0.000446$  are found, and it can be seen that the correct standard error for the difference of the two application technique means is shown as,  $2 \left( \frac{\sigma_p^2}{ar} + \frac{\sigma^2}{ars} \right) = 0.012103$ . For designs with more than two levels for the fixed factors, the estimable function in `gmodels` package, as well as the `lsmeans` function in the `lsmeans` package, described in Chapters 3 and 4, produce the correct standard errors of the means when operating on an object created by `lmer` rather than `aov`. The `lsmeans` package can also compute Tukey's adjusted pairwise comparisons of the means using the correct standard errors as shown in the code below. Contrasts between `tech1` and `tech2` are significant.

```
library(lsmeans)
lsmeans(mod5, pairwise ~ tech, adjust = c("tukey"))
```

```
$lsmeans
tech lsmean      SE df lower.CL upper.CL
1     0.272 0.00856 4   0.248   0.295
2     0.361 0.00856 4   0.338   0.385
```

Results are averaged over the levels of: form  
Degrees-of-freedom method: kenward-roger  
Confidence level used: 0.95

```
$contrasts
contrast   estimate   SE df t.ratio p.value
tech1 - tech2 -0.0899 0.0121 4  -7.426  0.0018
```

Results are averaged over the levels of: form  
Degrees-of-freedom method: kenward-roger

`lmer` also produces the correct  $F$ -tests for the fixed factors as shown below.

```
anova(mod5)
```

```
Analysis of Variance Table
            npar   Sum Sq  Mean Sq F value
form        1 0.0000138 0.0000138  0.0308
tech        1 0.0245970 0.0245970 55.1425
form:tech   1 0.0016638 0.0016638  3.7300
```

Another design whose model contains both fixed and random effects is the randomized block designs discussed in the last chapter. For example, consider the example from *Golf Magazine* discussed in **Section 5.8**. In that experiment the treatment factor (tee height) can be considered to be a fixed effect, since the object of the experiment was to determine if there is any difference in driving distance caused by different tee heights. However, there was no interest in comparing the specific golfers used in the experiment since they just denoted a sample of representative golfers. Therefore the block factor, or golfer, can be considered to be a random effect. The interaction between a random and fixed effect is also defined to be a random effect. Therefore the interaction of golfer and tee height should also be considered a random effect, and the model for the golf experiment can be written as:

$$y_{ijk} = \mu + b_i + \tau_j + b\tau_{ij} + \varepsilon_{ijk}, \quad \text{Eq. 5-15}$$

where  $b_i$  represents the random golfer effect,  $\tau_j$  represents the fixed tee height effect,  $b\tau_{ij}$  represents the random interaction effect, and  $\varepsilon_{ijk}$  represents the random repeat hit effect,  $i = 1, \dots, 3$ ,  $j = 1, \dots, 9$ , and  $k = 1, \dots, 5$ . By assuming the random golfer, interaction, and repeat hit effects to be independent and normally distributed with variances  $\sigma_b^2$ ,  $\sigma_{b\tau}^2$  and  $\sigma^2$ , respectively, the expected value and variance of  $y_{ijk}$  are given by:

$$E(y_{ijk}) = \mu + \tau_i$$

$$\text{Var}(y_{ijk}) = \sigma_b^2 + \sigma_{b\tau}^2 + \sigma^2$$

The expected values for the ANOVA mean squares are given in **Table 5-16**, where  $Q(\tau) = 8 \sum_{i=1}^3 \frac{(\tau_i - \bar{\tau}_.)^2}{3-1}$ .

These expected mean squares show that to test the hypothesis of no difference in tee height effect, i.e.,  $H_0: \tau_1 = \tau_2 = \tau_3$ , the correct denominator for the  $F$ -ratio would be the interaction Golfer  $\times$  Tee Height, as used in **Section 4.9**, and not the error mean square that is used by default in the ANOVA tables produced by the `aov` function. Also, the correct variance of a difference in marginal or least squares means for different tee heights  $\text{Var}(\bar{y}_{i..} - \bar{y}_{i'..}) = 2 \left( \frac{\sigma_{b\tau}^2}{45} + \frac{\sigma^2}{45} \right)$ .

**Table 5-16. Expected Mean Squares for Randomized Block Design with Replicates within a Block**

Source	df	Expected Mean Square
Golfer ( $b_i$ )	$(9 - 1) = 8$	$\sigma^2 + 5\sigma_{b\tau}^2 + 3\sigma_b^2$
Tee Height ( $\tau_j$ )	$(3 - 1) = 2$	$\sigma^2 + 5\sigma_{b\tau}^2 + Q(\tau)$
Golfer $\times$ Tee Height ( $b_{ij}$ )	$8 \times 2 = 16$	$\sigma^2 + 5\sigma_{b\tau}^2$
Repeat ( $\varepsilon_{ijk} = \text{Error}$ )	$(5 - 1) \times 16 = 64$	$\sigma^2$

By default the R function `aov` does not estimate  $\sigma_b^2$ . Therefore in Chapter 4, the `Error(id/teehgt)` option was used in the `aov` function call in order to get the correct  $F$ -test for tee height. Another way of

getting the correct  $F$ -test for tee height is to use the `lmer` function in the package `lme4`. In addition to producing the correct  $F$ -statistic, this function will calculate the correct standard error for the differences in tee height means.

## 5.10. Graphical Methods to Check Model Assumptions

Graphical analysis of data from sampling experiments to study variances are useful for checking assumptions of the statistical models and identifying atypical observations which may have a heavy influence on the resulting variance component estimates. Snee (1983) explains that another important advantage of graphical analysis is that it forces the analyst to become more familiar with the data and think critically about the mechanisms to explain the observed variation. He suggests simple graphical tools such as half-normal plots, gamma plots in the analysis of nested sampling designs. Normal plots can also be used to check the normality assumption for random effects.

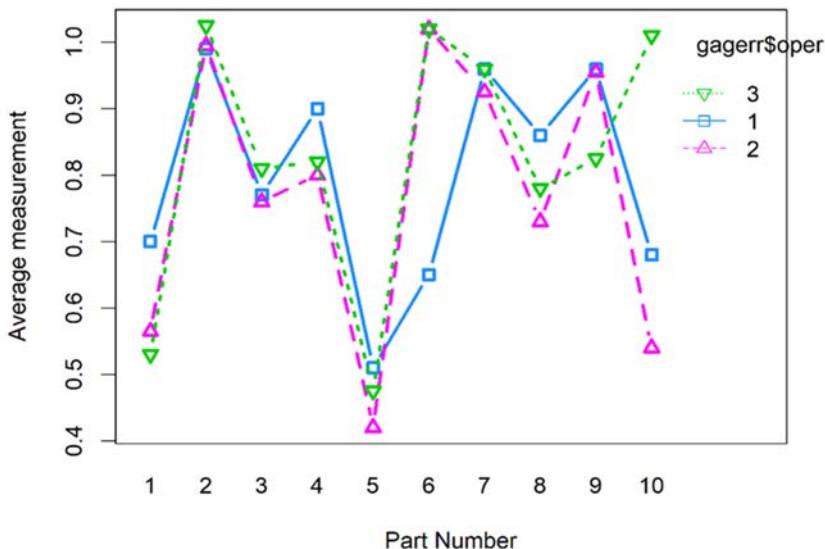
### 5.10.1. Simple Plots

Consider again the data in **Table 5-2**. **Eq. 5-1** is appropriate for the data from this experiment, where the random sample effects,  $t_i$ , and the random experimental errors,  $\varepsilon_{ij}$ , are assumed to be normally distributed. The assumption of constant variance of  $\varepsilon_{ij}$ 's across the various samples is also implied by  $\varepsilon_{ij} \sim N(0, \sigma^2)$ . A simple way to verify these assumptions is to make a simple boxplot of the data like that shown in **Figure 2-3**. There, nonconstant variances or outlier observations can be quickly detected. The normality of the  $\varepsilon_{ij}$  can also be checked with a normal probability plot like that shown in **Figure 5-4**. For the Gage R&R two-factor sampling experiment presented in **Section 5.6** using the method of moments resulted in a negative variance component estimate for  $\sigma_b^2$  the variance due to operators. This estimate was negative because the mean square for interaction of part by operator was larger than the mean square for operator. Simple interaction plots like **Figure 3-4** to **Figure 3-6** can facilitate an explanation of this result. **Figure 5-5** shows the interaction plot for part by operator. An interaction is characterized by the fact that the trace of the average response across the levels of one factor plotted separately

for each level of the other factor will not be parallel. In **Figure 5-5**, it can be seen that the line segments are close to parallel, except for the segments joining parts 5 through 7 and 9 through 10. This is due to the fact that operator 1 had a much lower average measurement on part number 6 than the other two operators, and operator 3 had a much higher measurement on part number 10 than the other two operators.

```
library(daewr)
data(gagerr)
interaction.plot(gagerr$part, gagerr$oper, gagerr$y,
  type = "b",
  main = "Interaction Plot for part by Operator",
  xlab = "Part Number", ylab = "Average measurement",
  pch = c(0,2,6),
  lwd = 2.5, lty = 1:3,
  col = c("dodgerblue","magenta","green3"))
```

**Interaction Plot for part by Operator**



**Figure 5-5. Interaction Plot for Part by Operator**

When the interaction mean square is largely due to one or two observations, as it is in this case, it might be wise to question the results. **Table 5-17** shows the variance component estimates from the data from

the Gage R&R study after eliminating parts 6 and 10. With these parts eliminated, the method of moments does not produce a negative variance component estimate, and the results obtained from the method of moments and REML are quite similar.

**Table 5-17. Comparison of Method of Moments and REML Estimates on Gage R&R Study after Eliminating Parts 6 and 10**

Component	Method of Moments Estimator	REML Estimator
<i>part</i> ( $\sigma^2$ )	0.03191	0.02808
<i>oper</i> ( $\sigma^2$ )	0.0008601	0.0008089
<i>part * oper</i> ( $\sigma^2$ )	0.0020045	0.0020082
<i>Error</i> ( $\sigma^2$ )	0.0004062	0.0004063

In a situation like this, if the parts are still available, it might be wise to have the operators remeasure parts 6 and 10 to see if there were mistakes.

### 5.10.2. Gamma and Half-Normal Plots to Check Constant Variance Assumption

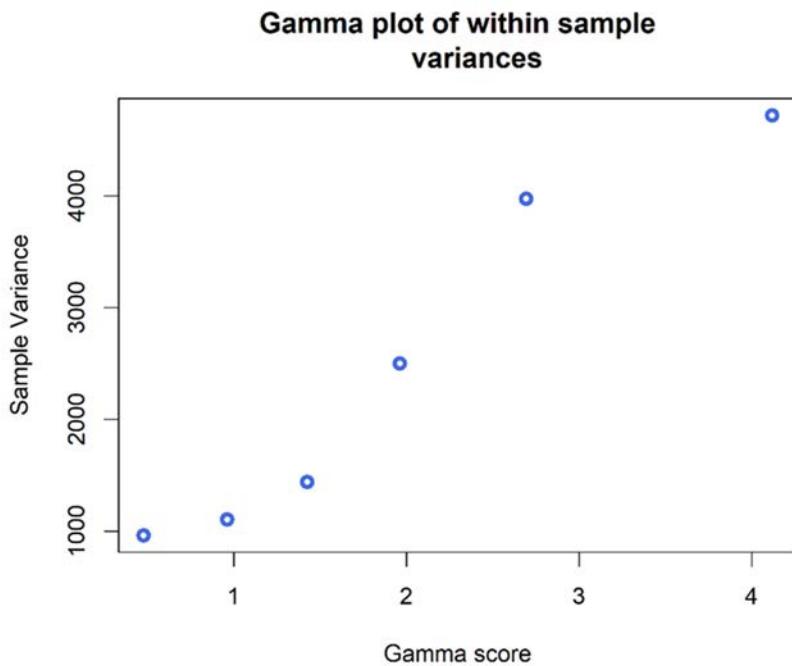
Wilk et al. (1962) proposed gamma probability plots as effective tools for investigating homogeneity of groups of variances. Sample of variances, with  $v$  degrees of freedom, follow the gamma distribution with shape parameter  $v/2$ , and thus the points on a gamma probability plot sample variances that are estimating the same constant  $\sigma^2$  should appear as a straight line. Many nested sampling designs and staggered nested sampling designs involve only samples of size two. In this case, half-normal plots of standard deviations can be used in the same way. To illustrate this, consider again the data in **Table 5-2**. The R code below computes the variances within each sample using the `tapply` function, stores them in the data frame `s2`, then uses `qgamma` function to compute the gamma quantiles. The plot of the gamma scores versus the sample variances is shown in **Figure 5-6**.

```
library(daewr)
data(Naph)
s2 <- tapply( Naph$yield, Naph$sample, var )
os2 <- sort(s2)
r <- c( 1:length(s2) )
```

```

gscore <- qgamma( (r - .5) / length (s2), 2)
plot(gscore, os2, main = "Gamma plot of within sample
variances", xlab = "Gamma score",
ylab = "Sample Variance",
lwd = 3 , col = "royalblue")

```



*Figure 5-6. Gamma Plot of within Sample Variances from Data in Table 5.2*

If the variances are homogeneous, the points on the gamma probability plot will lie roughly along a straight line. Individual points above the line to the far right would indicate variances that are larger than the norm. In *Figure 5-6* with only 6 points it appears that the points fall fairly close to a straight line indicating constant variances.

For an example where there are more variances to plot, consider the polymerization study shown in *Table 5-13*. The methods of moments estimator for the box within lot variance component for this staggered nested sampling study was negative. As with the gage R&R study, one or two atypical values may be the cause of this result, and a graphical analysis of the data may reveal that. The mean squares in the ANOVA presented in *Section 5.8* are pooled estimates of the variances for each

source in the design. If the four observations for lot number  $i$  are denoted by  $Y_{1i}$ ,  $Y_{2i}$ ,  $Y_{3i}$ , and  $Y_{4i}$  as shown below,

Lot	Box 1		Box 2	
	Preparation		Preparation	
	1	2	1	test 1
$i$	test 1 $Y_{1i}$	test 2 $Y_{2i}$	$Y_{3i}$	$Y_{4i}$

Snee (1983) shows that the variances to be pooled from each source to create the ANOVA mean squares are given by:

Source	Variance $s_i^2$
Error or test(prep)	$\frac{(Y_{2i} - Y_{1i})^2}{2}$
prep(box)	$\frac{2}{3} \left( Y_{3i} - \frac{(Y_{1i} + Y_{2i})}{2} \right)^2$
box	$\frac{3}{4} \left( Y_{4i} - \frac{(Y_{1i} + Y_{2i} + Y_{3i})}{3} \right)^2$

The R code to compute the standard deviations within each source is shown below.

```
library(daewr)
data(polymer)
y <- array( polymer$strength, c(4,30) )
sd1 <- sqrt( (y[2,] - y[1,])^**2 / 2 )
sd2 <- sqrt( (2/3) *
  ( y[3,] - (y[1,] + y[2,]) / 2 )^**2 )
sd3 <- sqrt( (3/4) *
  (y[4,] - (y[1,] + y[2,] + y[3,] )/3 )^**2 )
```

The method of moments estimator of the variance component for box within lot was negative because the mean square for prep within box and lot was larger than the mean square for box. This would suggest an investigation of the variances that are pooled to form the mean square for prep within box. The code to make a half-normal plot, **Figure 5-7**, of the standard deviations for prep within box (**sd2**) is shown below.

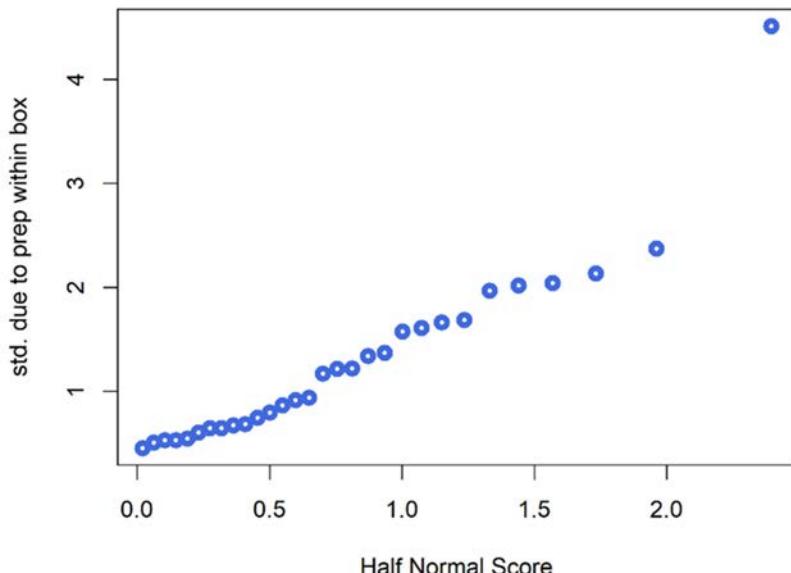
```
osd2 <- sort(sd2)
r <- c( 1: length(sd2))
zscore <- qnorm( ( ( r - .5 ) / length(sd2) +1 )/ 2 )
```

```

plot( zscore, osd2, main = "Half-normal plot of
  prep(box) standard deviations",
  xlab = "Half Normal Score",
  ylab = "std. due to prep within box")
osd2 <- sort(sd2)
r <- c( 1: length(sd2))
zscore <- qnorm( ( ( r - .5 ) / length(sd2) + 1 )/ 2)
plot( zscore, osd2, main = "Half-normal plot of
  prep(box) standard deviations",
  xlab = "Half Normal Score",
  ylab = "std. due to prep within box",
  lwd = 3 , col = "royalblue"))

```

**Half-normal plot of prep(box) standard deviations**



**Figure 5-7. Half-Normal Plot of Standard Deviations of Prep(Box)**

In the plot, shown in **Figure 5-7**, we see a relatively straight line of points extending from the lower left of the plot, except there is one standard deviation that sticks out above the line at the right side. **Table 5-18** shows the standard deviations, that were calculated with the R code above, within each source for each lot. In this table the standard

deviation for prep within box is labeled  $s_2$ , and it can be seen that there is a large value for  $s_2$  in lot 19, which is primarily due to the high result for  $Y_3 = 16.79$ , the first test of the second preparation for box 1.

**Table 5-18. Raw Data for Each Lot and Calculated Standard Deviations**

lot	$Y_1$	$Y_2$	$Y_3$	$Y_4$	$s_1$	$s_2$	$s_3$
1	9.76	9.24	11.91	9.02	0.368	1.968	1.111
2	10.65	7.77	10.00	13.69	2.036	0.645	3.652
3	6.50	6.26	8.02	7.95	0.170	1.339	0.886
4	8.08	5.28	9.15	7.46	1.980	2.017	0.038
5	7.84	5.91	7.43	6.11	1.365	0.453	0.823
6	9.00	8.38	7.01	8.58	0.438	1.372	0.390
7	12.81	13.58	11.13	10.00	0.544	1.686	2.171
8	10.62	11.71	14.07	14.56	0.771	2.372	2.102
9	4.88	4.96	4.08	4.76	0.057	0.686	0.104
10	9.38	8.02	6.73	6.99	0.962	1.608	0.912
11	5.91	5.79	6.59	6.55	0.085	0.604	0.393
12	7.19	7.22	5.77	8.33	0.021	1.172	1.389
13	7.93	6.48	8.12	7.43	1.025	0.747	0.069
14	3.70	2.86	3.95	5.92	0.594	0.547	2.093
15	4.64	5.70	5.96	5.88	0.750	0.645	0.387
16	5.94	6.28	4.18	5.24	0.240	1.576	0.196
17	9.50	8.00	11.25	11.14	1.061	2.041	1.348
18	10.93	12.16	9.51	12.71	0.870	1.662	1.596
19	11.95	10.58	16.79	13.08	0.969	4.511	0.023
20	4.34	5.45	7.51	5.21	0.785	2.135	0.482
21	7.60	6.72	6.51	6.35	0.622	0.531	0.514
22	5.12	5.85	6.31	8.74	0.516	0.674	2.581
23	5.28	5.73	4.53	5.07	0.318	0.796	0.095
24	5.44	5.38	4.35	7.04	0.042	0.865	1.718
25	3.50	3.88	2.57	3.76	0.269	0.914	0.384
26	4.80	4.46	3.48	3.18	0.240	0.939	0.924
27	5.35	6.39	4.38	5.50	0.735	1.217	0.110
28	3.09	3.19	3.79	2.59	0.071	0.531	0.664
29	5.30	4.72	4.39	6.13	0.410	0.506	1.149
30	7.09	7.82	5.96	7.14	0.516	1.221	0.159

If lot number 19 is removed from the data, the method of moments estimate of the variance component for box within lot is no longer negative and method of moments and REML estimators are quite consistent, as can be seen in the results shown in **Table 5-19** (on the next

page). This might lead one to question the result of test 1 of preparation 2 of box from lot 19. If the material was still available, a repeat test would be in order.

**Table 5-19. Comparison of Method of Moments and REML Estimates for Polymerization Study after Removing Lot 19**

Component	Method of Moments Estimator	REML Estimator
Lot ( $\sigma_a^2$ )	5.81864	6.09918
Box(Lot) ( $\sigma_b^2$ )	0.13116	0.04279
Prep(Box) ( $\sigma_c^2$ )	0.76517	0.79604
Error ( $\sigma^2$ )	0.63794	0.64364

### 5.10.3. Probability Plots of Empirical Best Linear Unbiased Estimates of Random Effects

In **Section 2.4**, to verify the assumption that the experimental errors follow a normal distribution, a normal probability plot of the residuals was made. In matrix notation, the vector of residuals can be written as:

$$\hat{\varepsilon} = \mathbf{y} - \mathbf{X}\hat{\beta} \quad \text{Eq. 5-16}$$

When there are random terms in the model in addition to the experimental error term,  $\varepsilon$ , we can check the normality of these random effects by making normal probability plots of the estimated random effects. A straight line of points in this plot would justify the assumption of normality for the random effect in question. This graphical technique will be useful in detecting atypical observations or departures from normality when there are at least 12 to 15 points on the plot. In the model

$$\mathbf{y} = \mathbf{X}\beta + \mathbf{Z}\gamma + \varepsilon \quad \text{Eq. 5-17}$$

the empirical best linear unbiased predictors (EBLUPs) of the random effects  $\gamma$  are given by the equation

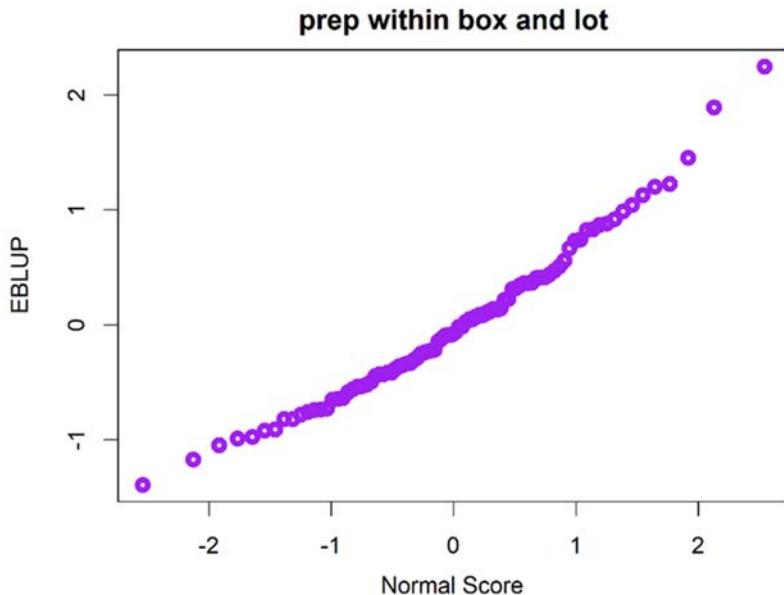
$$\hat{\gamma} = \hat{\mathbf{G}}\mathbf{Z}'\hat{\mathbf{V}}^{-1}(\mathbf{y} - \mathbf{X}\hat{\beta}) \quad \text{Eq. 5-18}$$

where  $\hat{\mathbf{G}}$  is the estimated variance covariance matrix of  $\gamma$ . The `lmer` function in R package `lme4` calculates the EBLUPs, and it can be used for models with all random effects as well as for models with both fixed

and random effects. On the next page is the R code for fitting the nested random effects model for the polymerization data using `lmer`. The EBLUPs can be retrieved from the object created by `lmer` with the `ranef` command.

The points on the normal plot of the EBLUPs for prep within box shown in **Figure 5-8** nearly follow a straight line. However, there are two apparent outliers on the upper right tail of the plot. When comparing the values of these two EBLUPs to a list of the data, it can be seen that the largest EBLUP (2.2488) corresponds to the first test of the second preparation for box 1 in lot 19.

```
library(lme4)
modr3 <- lmer( strength ~ 1 + (1|lot) + (1|lot:box) +
  (1|lot:box:prep), data = polymer)
qqnorm(ranef(modr3)$"lot:box:prep"[[1]],
  main = "prep within box and lot", ylab="EBLUP",
  xlab = "Normal Score" ,
  lwd = 4 , col ="royalblue")
```



**Figure 5-8. Normal Plot of Estimated Random Effects for Prep(Box)**

This is the same point identified by the half-normal plot in the last section and warrants further investigation. The second largest EBLUP

(1.8905) is associated with the test on preparation 1 in box 2 of lot 2. This value (13.69) is higher than any other value for lot 2 and again might warrant further investigation.

## **5.11. Review of Important Concepts**

Sampling experiments are conducted to estimate variances and to partition variances into various sources called variance components. Sometimes estimating variances or partitioning variances into various sources may be the end goal. In other cases partitioning the variance may be an intermediate step in a study trying to reduce the variance or an intermediate step in a study seeking to determine the causes of variability. The factors in a sampling experiment are called random factors as opposed to fixed factors described in Chapters 1–4. The purpose for including random factors in a design is to estimate the variability caused by varying levels of the random factor. Levels of the random factor are a representative sample of possible levels.

When the goal is to partition variance into two sources, one-factor random sampling experiments, or RSEs, are useful. Two-factor sampling experiments or factorial random sampling experiments (FRSEs), like those used in classical gage R&R studies, are used to partition variance into three independent sources. Nested sampling experiments, or NSEs, are used when it is more convenient to use a different set of levels of one factor for each level of another factor. Several stages of nesting can be utilized in one design to partition variance into several sources. Staggered nested sampling experiments, or SNSEs, are a variation of nested designs that balance the information collected over the various sources. **Figure 5-9** illustrates where these designs should be used in relation to the designs described in earlier chapters.

Variance components can be estimated using the analysis of variance method of moments or the maximum likelihood or REML methods. This can be accomplished using the R function `aov`, and the function `lmer` from the R package `lme4`. Formulas were presented in the chapter for exact confidence intervals on variance components estimated from one factor random sampling experiments RSE. Formulas were presented for approximate confidence limits on variance components from two-factor

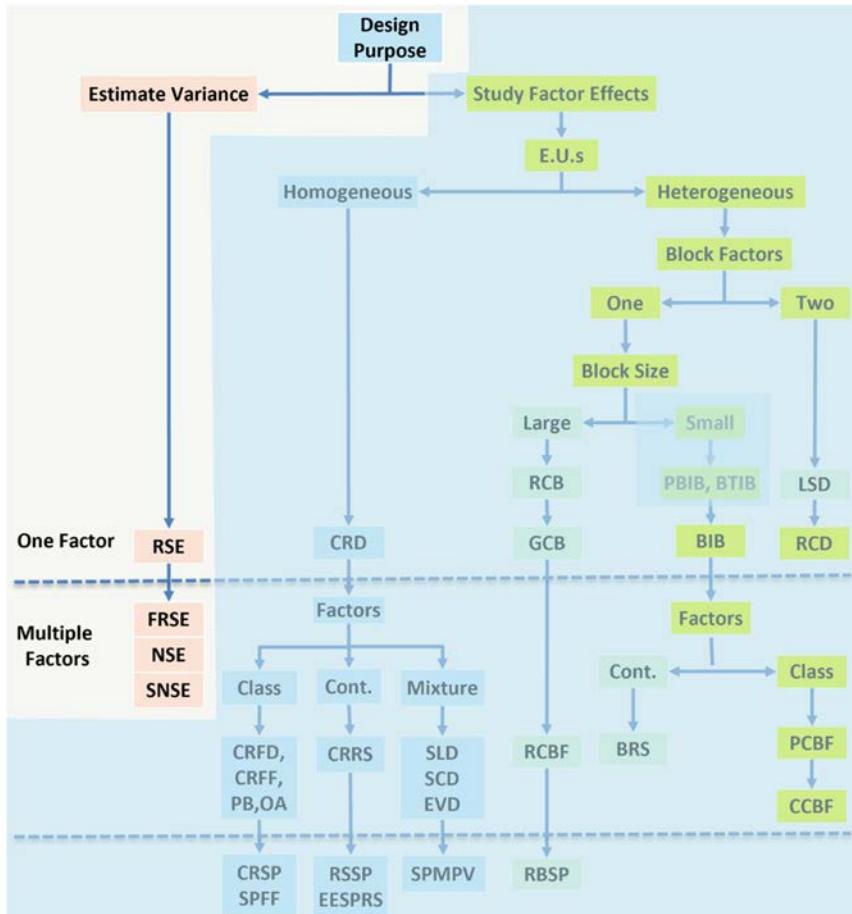
FRSE, or nested design NSE, and a function `vci` was introduced for evaluating these formulas. Asymptotic confidence intervals can be calculated for variance components using the likelihood profile method. These intervals will be reasonably accurate when there are 30–40 degrees of freedom associated with the term in question.

Sample sizes for sampling experiments can be determined to make the width of the confidence interval for  $\sigma^2$  the desired multiple of  $\sigma^2$ . The number of levels for other random factors in the design should be chosen to partition the degrees of freedom among the various sources in the design according to their perceived importance relative to the replicate variance  $\sigma^2$ .

In designs to study the effects of fixed factors, random factors can be introduced into the model by the way the experiment is conducted. For example, when response measurements are made on sub-samples of experimental units an additional nested error term must be included in the model for analysis. Block terms and block by treatment interactions in randomized block experiments would normally be considered to be random factors. When both fixed and random effects are included in the model, the expected values of the ANOVA mean squares (EMS) show which mean square should be used in the denominator of the  $F$ -test for each fixed effect or interaction in the model. The function `lmer` in the R `lme4` package is most useful for analyzing data from designs that include both fixed and random factors. It uses the REML method for estimating the variance components of the random effects, and it automatically computes the correct standard error for estimable functions of fixed effects.

Atypical response values can have a heavy influence on variance component estimates from sampling experiments. Simple plots, such as plots of the response versus the level in one-factor designs or interaction plots for two-factor random sampling experiments, can help to identify atypical response values. For nested designs (or staggered nested designs), gamma plots of sample variances (or half-normal plots of standard deviations) within each source are useful for checking homogeneity of variances and identifying cells where atypical values may occur. Normal probability plots of empirical best linear unbiased

estimators of random effects (EBLUPs) are useful for checking normality assumptions and again for identifying cells where atypical response values may lie.



*Figure 5-9. Design Selection Roadmap*

## 6. Fractional Factorial Designs

---

### 6.1. Introduction

There are two benefits to studying several treatment factors simultaneously in a factorial design. First, the interaction or joint effects of the factors can be detected. Second, the experiments are more efficient. In other words, the same precision of effects can be achieved with fewer experiments than would be required if each of the factors was studied one-at-a-time in separate experiments. The more factors included in a factorial design, the greater the efficiency and the greater the number of interactions that may be detected. However, the more factors included in a factorial experiment, the greater the number of runs that must be performed. When many factors are included in a factorial experiment, one way to reduce the number of runs is to use only two levels of each factor and run only one experiment per cell or treatment combination. These ideas were discussed in **Sections 3.7** and **3.7.5**.

In the preliminary stage of experimentation, where the objective may be to determine which factors are important from a long list of candidates, a factorial design may require too many experiments to perform even when there are only two levels of each factor and only one replicate per cell. Table 6.1 shows the number of experiments required for a  $2^k$  factorial design as a function of the number of factors,  $k$ . With  $k = 7$  or more factors, the large number of runs required for a  $2^k$  is usually impractical.

**Table 6-1. Number of Experiments Required for  $2^k$  Design**

Number of Factors (k)	Number of Experiments ( $2^k$ )
4	16
5	32
6	64
7	128
8	256
9	512

When the number of factors under study is large, researchers will often

abandon the efficiency of factorial experiments altogether and revert to a “seat of the pants” approach or vary one-factor-at-a-time plan. Others will run a factorial experiment with a subset of the factors, chosen from the longer list by guessing which ones may be more important. However, these approaches are less than optimal and do not retain the benefits of factorial experiments with a large number of factors. A better solution to this problem is to use a fraction of the experiments, or runs, required for a full factorial experiment. To be effective, the fraction of runs used must be carefully selected in order to preserve some of the benefits of a full factorial experiment.

One of the desirable properties of a  $2^k$  factorial plan is that factor effects are not obscured by (or correlated with) planned changes in other factors. This property was called orthogonality in Section 3.7.

## 6.2. Half-Fractions of $2^k$ Designs

Consider first choosing a **half-fraction** of a  $2^k$  factorial experiment. Careless choice of half the  $n = 2^k$  runs may not retain the desirable orthogonality property of a  $2^k$  design. One way to preserve this property, when selecting a one-half fraction of a  $2^k$  factorial experiment, is to choose the runs where the coded factor levels for an interaction term (preferably the highest order interaction) are constant.

**Table 6-2** illustrates this procedure. On the left side of the table is a representation of the coded factor levels for a  $2^4$  design. On the right side of the table are the runs that have a constant (+) value for  $X_A \times X_B \times X_C \times X_D$ . These runs represent the half-fraction. The order of the runs on the right side of the table have been reordered so that it can be easily seen that the standard factorial pattern is present in the first three columns. Therefore it can be seen that the orthogonality property is preserved for the first three columns. By further inspection, it can be seen that the fourth factor is also orthogonal to the other three.

With 16 runs in a full  $2^4$  factorial, 15 effects can be estimated in addition to the grand mean. The 15 effects consist of the four main effects, six two-factor interaction effects, four three-factor interactions, and one four-factor interaction. In a half-fraction of a  $2^4$  experiment, however,

there are only 8 runs. Thus only 7 effects can be estimated in addition to the grand mean. By choosing the runs from a full factorial that have a constant value for an interaction, we automatically lose the ability to estimate that interaction effect. By studying the right half of **Table 6-2**, it can also be seen that the coded factor levels or column of signs for  $X_D$  is exactly the product of signs in the first three columns, that is,  $X_D = X_A \times X_B \times X_C$ . This means that the effect we can estimate for  $X_D$  will be completely obscured or confused by the three-factor interaction  $X_A X_B X_C$ . This is not all. As will be shown later, each main effect and interaction in the design is obscured or confounded with one other interaction. This is the price we pay for running one-half the total number of experiments. However, in preliminary experiments where a large number of factors are included in order to find out which ones are truly important, this may not be a serious price to pay.

**Table 6-2. Creating a Half-Fraction by Choosing the Runs in a Full Fraction with Constant Values for an Interaction**

run	Full Factorial				$X_A X_B X_C X_D$	run	Half-Fraction			
	$X_A$	$X_B$	$X_C$	$X_D$			$X_A$	$X_B$	$X_C$	$X_D$
1	-	-	-	-	+	1	-	-	-	-
2	+	-	-	-	-	10	+	-	-	+
3	-	+	-	-	-	11	-	+	-	+
4	+	+	-	-	+	4	+	+	-	-
5	-	-	+	-	-	13	-	-	+	+
6	+	-	+	-	+	6	+	-	+	-
7	-	+	+	-	+	7	-	+	+	-
8	+	+	+	-	-	16	+	+	+	+
9	-	-	-	+	-					
10	+	-	-	+	+					
11	-	+	-	+	+					
12	+	+	-	+	-					
13	-	-	+	+	+					
14	+	-	+	+	-					
15	-	+	+	+	-					
16	+	+	+	+	+					

In preliminary experiments involving a large number of factors, usually

only a small proportion of the factors will have significant effects. This fact has been called the effect sparsity principle by Box and Meyer (1986a). Just as two planets will line up with the moon in the night sky more frequently than three planets will, main effects are more likely to be important than two-factor interactions, and two-factor interactions are more likely to be important than three-factor interactions, and so forth. This general rule has been called the hierarchical ordering principle by Wu and Hamada (2000). Therefore, if fractions of factorial experiments can be planned in a way that main effects are confounded with three-factor and higher-order interactions, the amount of information lost by fractionating the number of runs will be small in comparison to the benefit of a reduced number of runs.

The way a fractional factorial of a  $2^k$  is created in practice is actually the opposite order of what was shown above. Instead of starting with a full factorial and eliminating runs to get the desired fraction, start with a full factorial containing the desired number of runs and add additional factors to the design. For example, to construct a one-half fraction of a  $2^k$  design, denoted by  $\frac{1}{2} 2^k$  or  $2^{k-1}$ , the procedure is as follows:

- a. Write down the base design, a full factorial plan in  $k - 1$  factors using the coded factor levels (-) and (+).
- b. Add the  $k$ th factor to the design by making its coded factor levels equal to the product of the other factor levels (i.e., the highest order interaction).
- c. Use these  $k$  columns to define the design.

A complete list of interactions confounded with each main effect and interaction in a half-fractional factorial is called the **confounding pattern** or **alias structure** of the design. This list is easy to construct based on the assignment of the  $k$ th factor in item 2 of the list above. For example, in the  $2^{4-1}$  design, if the levels of the fourth factor are equal to the product of the levels of the first three factors in the design, we write symbolically  $D = ABC$ . This is called the **generator** of the design. Multiplying on both sides of the generator we get:

$$D^2 = ABCD$$

or

$$I = ABCD$$

where  $I$  represents a column of plus signs and is the multiplicative identity for elementwise products of columns of coded factor levels. The equation,  $I = ABCD$ , is called the **defining relation** for the fractional factorial design, and by multiplying on both sides of this equation, the interaction confounded with any main effect, or interaction can be determined. For example, multiplying by the first factor on both sides of the defining relation we see:

$$A(I) = A(ABCD)$$

or

$$A = BCD$$

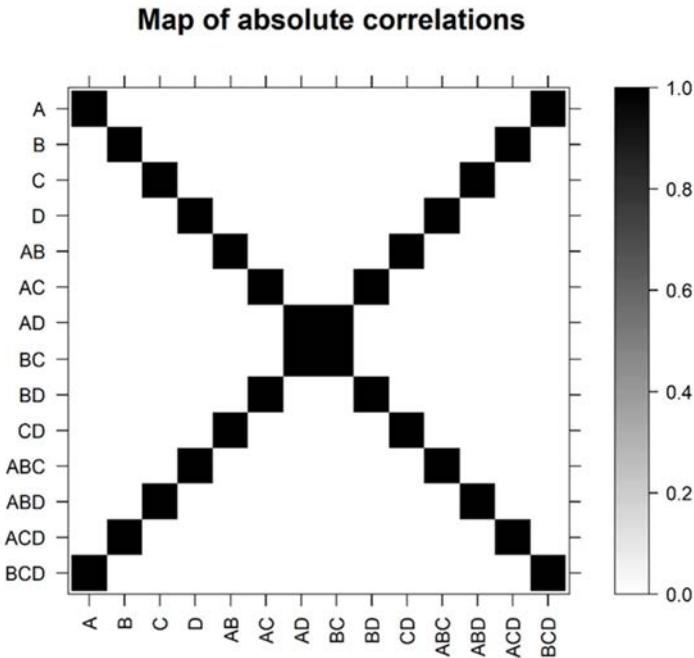
This means that the effect of the first factor  $A$  is confounded with the three-factor interaction  $BCD$ . When data is collected, the effect of factor  $A$  is estimated as the difference in the average response at the high and low levels of factor  $A$ . However, that effect really estimates the sum of the effects of factor  $A$  and the three-factor interaction. Therefore we write it as  $A + BCD$ .

The complete alias pattern for this  $2^{4-1}$  design can be determined by multiplying the defining relation by each main effect and interaction resulting in:

$$\begin{aligned} & I + ABCD \\ & A + BCD \\ & B + ACD \\ & C + ABD \\ & D + ABC \\ & AB + CD \\ & AC + BD \\ & AD + BC \end{aligned}$$

There are only eight unique results in this alias pattern, and they represent the eight effects ( $I$  is the overall mean) that can be estimated from the 8-run fractional factorial design. The alias pattern can also be represented graphically as a color map of the correlation matrix computed from the columns of the design matrix as can be seen in

**Figure 6-1.**



**Figure 6-1. Color Map of Correlations**

In this graph the color intensity in each grid square represents the correlation between the columns in the design matrix. For example, it can be seen that factor *A* has a correlation of 1 with itself, and the *BCD* interaction, and zero correlation with all other factors and interactions. This gives the same information as the alias pattern. When the correlation between a factor and an interaction is 1, they both cannot be included in the model fit by least squares, or they will cause singularities in the  $\mathbf{X}'\mathbf{X}$  matrix.

Replicates are not included in a one-half fractional factorial design because replicates would take as many additional experiments as it would to complete the full factorial. Therefore, there is no estimate of  $\sigma^2$ , the variance of experimental error, when running a fractional factorial. In order to judge the significance of the effects in a fractional factorial design, graphical methods like those described in **Section 3.10.2** should be used. The effects found to be significant should be interpreted

with the hierarchical ordering principle in mind. For example, if the effect for  $B + ACD$  were found to be significant, it will be assumed to represent the effect of factor  $B$  rather than the three-way interaction.

The easiest way to create a  $2^{k-1}$  factorial in R is to use the function `FrF2` in the R package `FrF2`. The example below creates the 16-run  $2^{5-1}$  design with generator  $E = ABCD$ . If the generator is left off, `FrF2` finds one that is optimal in the sense that will be described in [Section 6.4](#).

```
library(FrF2)
design <- FrF2( 16, 5, generators = "ABCD",
randomize = FALSE)
design
```

	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
9	-1	-1	-1	1	-1
10	1	-1	-1	1	1
11	-1	1	-1	1	1
12	1	1	-1	1	-1
13	-1	-1	1	1	1
14	1	-1	1	1	-1
15	-1	1	1	1	-1
16	1	1	1	1	1

```
class=design, type= FrF2.generators
```

In practice the treatment combinations should be randomized to the experimental units, and a randomized list like the one shown in [Section 3.4](#) should be used. This design is then called a completely randomized fractional factorial design of CRFF. The design resulting from the code above is in standard order (not randomized), remove the option `randomize=FALSE` to get a randomized list. The `FrF2` package function `design.info(design)` will print out information about a design previously created by the `FrF2` function such as the generator, the factor names, the alias pattern, whether or not there are replicates, and

whether the order has been randomized. The `FrF2` function alias will just print the alias structure for a design previously constructed. This function requires a response vector  $y$ , and a vector of random uniform numbers was used in the example below.

```
library(FrF2)
y <- runif(16, 0, 1)
aliases(lm(y ~ (.)^4, data = design))
```

```
A = B:C:D:E
B = A:C:D:E
C = A:B:D:E
D = A:B:C:E
E = A:B:C:D
A:B = C:D:E
A:C = B:D:E
A:D = B:C:E
A:E = B:C:D
B:C = A:D:E
B:D = A:C:E
B:E = A:C:D
C:D = A:B:E
C:E = A:B:D
D:E = A:B:C
```

In this alias pattern for the  $2^{5-1}$  design, it can be seen that main effects are confounded with four-way interactions, and two-factor interactions are confounded with three-factor interactions. Therefore, if three and four-factor interactions could be assumed negligible, estimates of all main effects and two-factor interactions could be made.

To illustrate the analysis of a  $2^{k-1}$  design, consider a continuation of the dry soup mix example presented in [Section 5.4.3](#). In that example, the majority of variability in the soup “intermix” was found to be within a batch rather than between batches. The researchers responsible for the project made a list of factors that they thought might influence the variability within a batch (see (Hare, 1988)). These were options that could be changed on the mixer where a batch was mixed and the intermix was added through ports. The list consisted of: (1) the number of ports where intermix was added, (2) the temperature of the mixer (that could be controlled by adding cooling water to the jacket surrounding it), (3) the mixing time, (4) weight of the batch, and (5) the

delay time between mixing and packaging. The response or variability in fill weights of the intermix was obtained by taking five consecutive samples of the soup mix every 15 minutes as the batch was being packaged. The factor labels and levels are shown in **Table 6-3**.

**Table 6-3. Factors and Levels for Soup Mix  $2^{5-1}$  Experiment**

Factor Label	Name	Low Level	High Level
A	Number of Ports	1	3
B	Temperature	Cooling Water	Ambient
C	Mixing Time	60 sec.	80 sec.
D	Batch Weight	1500 lb	2000 lb
E	Delay Days	7	1

The normal batch size was 2000 lbs and the normal mixing time was 60 seconds. Since this experiment was to be run in a production facility, the research and production staff both had to agree to the plan. The plan agreed upon was the  $2^{5-1}$  created on page 198, and **Table 6-4** shows a list of the experiments in actual factor levels. The list is in standard order, with the random run orders listed in the first column.

**Table 6-4.  $2^{5-1}$  Experiment to Determine Which Factors Are Associated with Fill Variation in Random Order**

Random Run Order	(A) Number of Ports	(B) Temperature	(C) Mixing Time (sec)	(D) Batch Weight (lb)	(E) Delay (days)	Response $\hat{o}_p$
12	1	Cool Water	60	1500	1	1.13
13	3	Cool Water	60	1500	7	1.25
5	1	Ambient	60	1500	7	0.97
3	3	Ambient	60	1500	1	1.70
6	1	Cool Water	80	1500	7	1.47
4	3	Cool Water	80	1500	1	1.28
16	1	Ambient	80	1500	1	1.18
14	3	Ambient	80	1500	7	0.98
1	1	Cool Water	60	2000	7	0.78
15	3	Cool Water	60	2000	1	1.36
7	1	Ambient	60	2000	1	1.85
10	3	Ambient	60	2000	7	0.62
11	1	Cool Water	80	2000	1	1.09
2	3	Cool Water	80	2000	7	1.10
9	1	Ambient	80	2000	7	0.76
8	3	Ambient	80	2000	1	2.10

All factor levels could be changed between batches with relatively little

effort, and randomization was not a problem. The requested reduction in batch weight and the increase in mixing time for some batches in the planned list of experiments would not seriously interfere with the production schedule if the list was short, but it would slow production if a 32-run design were used. For this reason the 16-run half fractional factorial was agreed upon. The experimental unit for this experiment was the batch of dried soup mix put in the mixer. The response  $\hat{\sigma}_p$  was an estimate of the standard deviation of fill weight within a batch, computed from a sampling study made during packaging of each batch.

The same design can be created with actual factor names and levels. In the R code below this is demonstrated along with the use of the `add.response` function from the `DoE.base` package to include the response. After adding the response, the model (`mod1`) was fit to the soup experiment data using the R function `lm`, and part of the summary of the `lm object (mod1)` includes the actual factor names.

```
library(FrF2)
soup <- FrF2(16, 5, generators = "ABCD", factor.names =
  list(Ports = c(1,3), Temp = c("Cool", "Ambient"),
    MixTime = c(60,80), BatchWt = c(1500,2000),
    Delay = c(7,1)), randomize = FALSE)
y <- c(1.13, 1.25, .97, 1.70, 1.47, 1.28, 1.18, .98,
  .78, 1.36, 1.85, .62, 1.09, 1.10, .76, 2.10 )
library(DoE.base)
soup <- add.response( soup , y )
mod1 <- lm( y ~ (.)^2, data = soup)
summary(mod1)
```

Call:

```
lm.default(formula = y ~ (.)^2, data = soup)
```

Residuals:

```
ALL 16 residuals are 0: no residual degrees of freedom!
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	1.22625	NaN	NaN	NaN
Ports1	0.07250	NaN	NaN	NaN
Temp1	0.04375	NaN	NaN	NaN

MixTime1	0.01875	NaN	NaN	NaN
BatchWt1	-0.01875	NaN	NaN	NaN
delay1	0.23500	NaN	NaN	NaN
Ports1:Temp1	0.00750	NaN	NaN	NaN
Ports1:MixTime1	0.04750	NaN	NaN	NaN
Ports1:BatchWt1	0.01500	NaN	NaN	NaN
Ports1:delay1	0.07625	NaN	NaN	NaN
Temp1:MixTime1	-0.03375	NaN	NaN	NaN
Temp1:BatchWt1	0.08125	NaN	NaN	NaN
Temp1:delay1	0.20250	NaN	NaN	NaN
MixTime1:BatchWt1	0.03625	NaN	NaN	NaN
MixTime1:delay1	-0.06750	NaN	NaN	NaN
BatchWt1:delay1	0.15750	NaN	NaN	NaN

Residual standard error: NaN on 0 degrees of freedom  
 Multiple R-squared: 1, Adjusted R-squared: NaN  
 F-statistic: NaN on 15 and 0 DF, p-value: NA

The expression  $formula = y(.)^2$  in the call to the `lm` function causes it to fit a **saturated model** in the main effects and two-factor interactions. Since there are no replicates, a normal probability plot of the regression coefficients was used to aid in judging which effects are significant. This is the same thing that was done in **Section 3.10.2**, except each effect in this model is confounded with one other interaction as shown in the alias pattern in **Section 6.2**. Therefore only 15 effects can be estimated. In the code below, the data frame is recreated using coded factor labels and the LGB function is used to make a half-normal plot.

```
library(daewr)
data(gagerr)
interaction.plot(gagerr$part, gagerr$oper, gagerr$y,
type = "b",
main = "Interaction Plot for part by Operator",
xlab = "Part Number", ylab="Average measurement",
pch = c(0,2,6), lwd = 2.5, lty = 1:3,
col = c("dodgerblue","magenta","green3"))
soupc<-FrF2(16,5,generators="ABCD",randomize=FALSE)
soupc<-add.response(soupc, y)
modc<-lm(y~(.)^2, data=soupc)
library(daewr)
LGB(coef(modc)[-1], rpt = TRUE)
```

As can be seen in the plot of effects shown in **Figure 6-2**, the main effect

$E$  (Delay Time between mixing and packaging),  $BE$  (the interaction between Temperature and Delay Time), and  $DE$  (the interaction between Batch Weight and Delay Time) appear to be significant. If the hierarchical effect ordering can be assumed, this is the correct interpretation, and the three-factor and four-factor interactions can be assumed negligible. The printed report (`rpt=TRUE`) show the half-effect values.

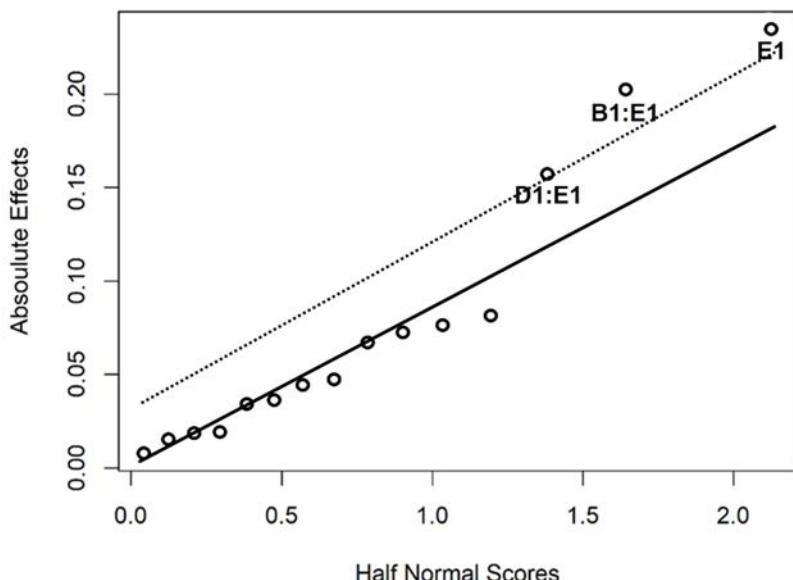


Figure 6-2. Normal Plot of Effects from the Fill Variability Experiment

#### Effect Report

Label	Half Effect	Sig(.05)
A1	0.07250	no
B1	0.04375	no
C1	0.01875	no
D1	-0.01875	no
E1	0.23500	yes
A1:B1	0.00750	no
A1:C1	0.04750	no
A1:D1	0.01500	no
A1:E1	0.07625	no
B1:C1	-0.03375	no
B1:D1	0.08125	no

```

B1:E1      0.20250      yes
C1:D1      0.03625      no
C1:E1     -0.06750      no
D1:E1      0.15750      yes

Lawson, Grimshaw & Burt Rn Statistic = 1.174362
95th percentile of Rn = 1.201

```

$E$  = Delay Time has a positive effect; this would normally mean that increasing the Delay Time between mixing and packaging would increase the response (or within batch variation). However, due to the unconventional assignment of 7 to the low (coded) level of factor  $E$  and 1 to the high (coded) level in **Table 6-3**, it actually tells us that on the average increasing the delay time between mixing and packaging decreases the within batch variability. Since the interactions BE and DE also appear to be significant, the average main effect of factor E does not tell the whole story. **Figure 6-3** shows the interaction plot for Temperature and Delay Time. Here it can be seen that Delay Time has little effect on the within batch variability when the mixer is cooled with cooling water during mixing. However, the within batch variability is decreased substantially by increasing delay time between mixing and packaging when the mixing was done at ambient temperature.

```

delay <- as.numeric(sub(-1, 7, soup$delay))
temp <- soup$Temp
interaction.plot(delay, temp, soup$y, ype = "b",
                 pch = c(24,18,22), leg.bty = "o",
                 main = "Interaction Plot for Mixing Temperature by
                 Delay time", xlab="Delay Time (days)",
                 ylab="Average S.D. Fill Weight",
                 lwd = 2.5, lty = 1:2, ol=c("dodgerblue","magenta"))

```

The **IAPlot** function can also be used to make several interaction plots (see **Figure 6-4**) simultaneously as shown in the code examples for this chapter.

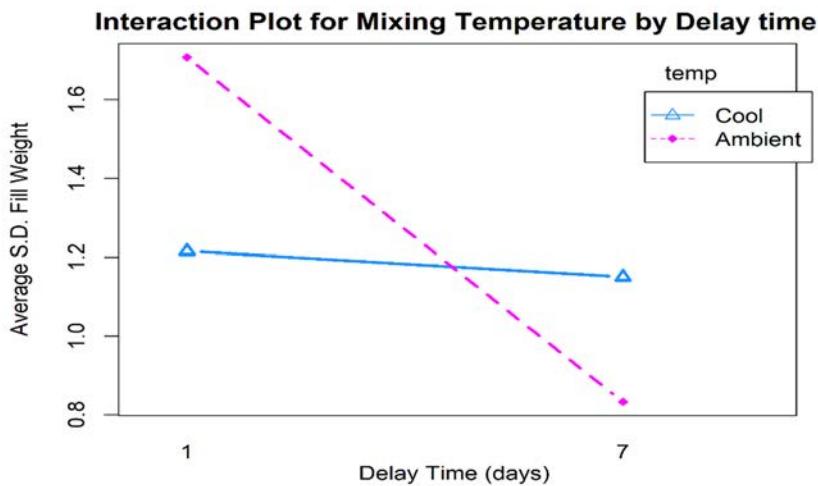


Figure 6-3. Interaction Plot for Mixing Temperature by Delay Time

```
library(FrF2)
IAPlot(soup, sel = c(2,4,5), abbrev = 7, lwd = 2)
```

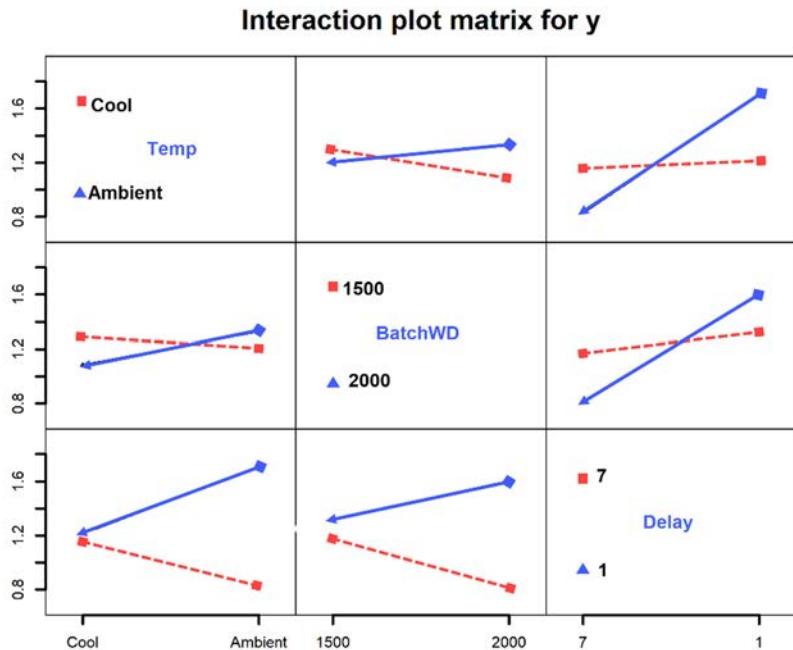


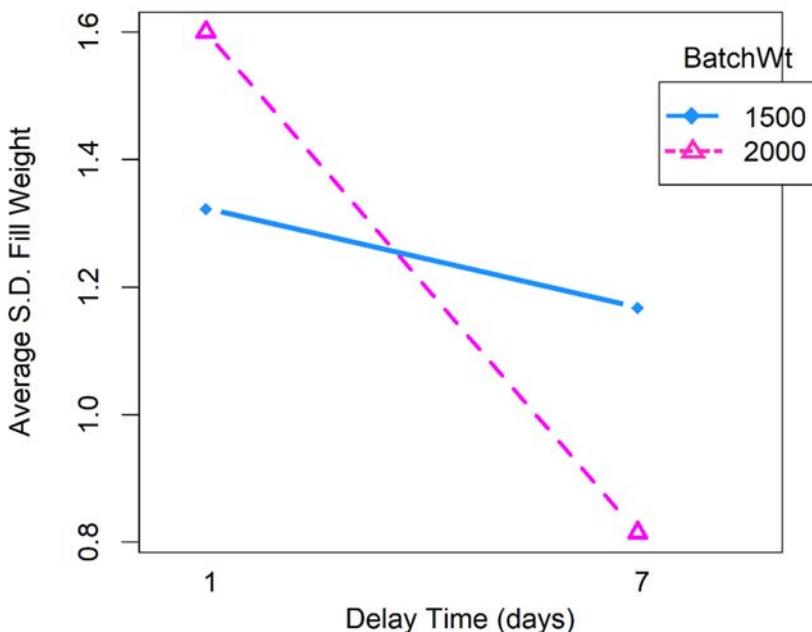
Figure 6-4. Interaction Plot for all factors

Figure 6-5 shows the interaction plot for Batch Weight and Delay Time.

There it can be seen that increasing the delay time between mixing and packaging has little effect on the variability within a batch for small (1500 lb) batches, while increasing the delay time between mixing and packaging decreases within batch variability for large (2000 lb) batches.

```
BatchWt <- ((as.numeric(soup$BatchWt)-1.5)/.5)*250+1750
interaction.plot(delay, BatchWt, soup$y, type = "b",
pch=c(18,24,22), leg.bty = "o",
main="Interaction Plot for Batch Weight by Delay
Time", xlab="Delay Time (days)",
ylab = "Average S.D. Fill Weight")
```

**Interaction Plot for Batch Weight by Delay time**



*Figure 6-5. Interaction Plot for Batch Weight by Delay Time*

Based on the results of this study, the minimum variation of intermix within a batch could be obtained by using the larger batch size (2000 lb), the ambient temperature at mixing, and a 7-day delay between mixing and packaging. It did not answer a new question prompted by the results: does it take a full 7-day delay between mixing and packaging or could a 3-day or 4-day delay reduce variability just as much? However,

the production and research staff agreed that the current results made sense, since the uniformity of intermix related to the hardness of the vegetable oil, which is affected by temperature and agitation. Production staff implemented the optimal conditions while research continued investigating the new question.

It is interesting to think that neither of the interactions would have been detected if one-factor-at-a-time experiments had been conducted, rather than the fractional factorial plan. The conclusions of one-factor-at-a-time type experiments may not be reproducible because it would not be realized that the effects (or absence of effects) seen could depend on other factor settings that are deemed to be insignificant.

### **6.3. Quarter and Higher Fractions of $2^k$ Designs**

In  $2^{k-1}$  designs only half the experimental runs are made, and each effect that can be estimated is confounded with one interaction. Likewise, in a quarter fraction  $2^{k-2}$  design, only one-fourth the experimental runs from the full  $2^k$  design are run, and each effect that can be estimated is confounded with three other interactions. In a one-eighth fraction, only one-eighth of the runs in the full factorial are made and each estimated effect will be confounded with seven other interactions, and so on. These designs may sound confusing at first because of the large number of effects confounded with each estimable effect. However, they are used quite frequently in practice, and by following the effect sparsity principle and the hierarchical ordering principle useful conclusions can usually be reached after just a fraction of the total runs required for a full factorial.

To construct a quarter fraction of a  $2^k$  design, start with a base design in  $2^{k-2}$  factors, then add two additional factors by making their coded factor levels equal to two interactions among the first  $k - 2$  columns.

$X_A$	$X_B$	$X_C$	{ $X_D$ }		{ $X_E$ }		$X_AX_BX_C$
			$X_AX_B$	$X_AX_C$	$X_BX_C$	-	
-	-	-	+	+	+	-	
+	-	-	-	-	+	+	
-	+	-	-	+	-	+	
+	+	-	+	-	-	-	

-	-	+	+	-	-	+
+	-	+	-	+	-	-
-	+	+	-	-	+	-
+	+	+	+	+	+	+
$X_A$	$X_B$	$X_C$	$X_D$	$X_E$		
-	-	-	+	+		
+	-	-	-	-		
-	+	-	-	+		
+	+	-	+	-		
-	-	+	+	-		
+	-	+	-	+		
-	+	+	-	-		
+	+	+	+	+		

There are two generators for the design above,  $D = AB$  and  $E = AC$ . From these it can be seen that  $I = ABD$  and  $I = ACE$ . Also, since  $I2 = I$ , a third equality, called the generalized interaction, is  $I = ABD(ACE)$  or  $I = BCDE$ . Combining the three equations obtained from the two generators and the generalized interaction results in the defining relation for the design  $I = ABD = ACE = BCDE$ . The confounding pattern, or alias structure, for the design is found by multiplying through the defining relation by each effect that can be estimated:

$$\begin{aligned}
 & AA + BD + CE + ABCDE \\
 & B + AD + ABCE + CDE \\
 & C + ABCD + AE + BDE \\
 & D + AB + ACDE + BCE \\
 & BC + ACD + ABE + DE \\
 & BE + ADE + ABC + CD
 \end{aligned}$$

Construction of one-eighth and higher fractions is similar. To construct a one-eighth fraction of a  $2^6$  design or  $2^{6-3}$  design, start with a base design in  $6 - 3 = 3$  factors, then add three additional factors by confounding them with interactions. For example, if we choose the generators  $D = A_B$ ,  $E = A_C$ , and  $F = B_C$ , the R code using F Staggered nested sampling experiments `rF2` below could be used to generate the design.

```

library(FrF2)
frac <- FrF2( 8, 6, generators = c("AB", "AC", "BC"))
frac

```

	A	B	C	D	E	F
1	-1	1	-1	-1	1	-1
2	-1	1	1	-1	-1	1
3	1	1	-1	1	-1	-1
4	-1	-1	1	1	-1	-1
5	1	-1	1	-1	1	-1
6	1	1	1	1	1	1
7	1	-1	-1	-1	-1	1
8	-1	-1	-1	1	1	1

```
class=design, type= FrF2.generators
```

There are eight runs in this design, and seven effects in addition to the overall mean can be estimated. Each effect will be aliased with seven interactions. To find what interactions are aliased with each effect that can be estimated, first find the defining relation. From the generators  $I = ABD = ACE = BCF$ . The two-factor generalized interactions are  $ABD(ACE) = BCDE$ ,  $ABD(BCF) = ACDF$ , and  $ACE(BCF) = ABEF$ . The three-factor generalized interaction is  $ABD(ACE)(BCF) = DEF$ . Combining the equations found from the generators and the generalized interactions the defining relation is:

$$I = ABD = ACE = BCF = BCDE = ACDF = ABEF = DEF$$

Multiplying through the defining relation by the main effects ( $A - F$ ), and the three-factor interaction ( $ABC$ ), the aliases for the seven effects that may be estimated can be determined. One quick way to do this in R is to use the aliases function in the [FrF2](#) package as shown previously in [Section 6.2](#).

## 6.4. Criteria for Choosing Generators for $2^{k-p}$ Designs

There is more than one alternative when selecting the generators for a  $2^{k-p}$  design. For example, to create a quarter fraction of a  $2^6$  design the generators  $E = ABC$  and  $=ABD\&$  could be used, or the generators  $E = AB$  and  $F = ACD$  could be used. The first selection results in the defining relation and alias structure (for the main effects only) shown below:

$$\begin{aligned}
I &= ABDE = ABDF = CDEF \\
A + BCE + BDF + ACDEF & \\
B + ACE + ADF + BCDEF & \\
C + ABE + ABCDF + DEF & \\
D + ABCDE + ABF + CEF & \\
E + ABC + ABDEF + CDF & \\
F + ABCEF + ABD + CDE &
\end{aligned}$$

The second set of generators results in the defining relation and alias structure (for the main effects only) below:

$$\begin{aligned}
I &= ABE = ACDF = BCDEF \\
A + BE + CDF + ABCDEF & \\
B + AE + ABCDF + CDEF & \\
C + ABCE + ADF + BDEF & \\
D + ABDE + ABF + CEF & \\
E + ABC + ACF + ABEF & \\
F + ABEF + ACD + BCDE &
\end{aligned}$$

Both generators result in 16-run fractional factorials, but the first set of generators might be preferable, since each main effect is confounded with three-factor interactions and higher order interactions, while using the second set of generators results in a design where main effects  $A$  and  $B$  are confounded with one two-factor interaction each. The first design has a smaller chance for confusion since the hierarchical ordering principle tells us that three-factor interactions are less likely to be important than two-factor interactions.

Three general criteria have been proposed for guiding the choice among the various possible sets of generators for any possible  $2^{k-p}$  design. These criteria are the **resolution criteria**, the **aberration criteria**, and the **clear effects criteria**.

Box and Hunter (1961) first proposed the **resolution criteria**. The resolution of the design is defined to be the length of the shortest word in the defining relation. For example, in the first defining relation for a  $2^{6-2}$  design shown above, the shortest word has length 4. Thus it is a resolution IV design. The shortest word for the second defining relation for a  $2^{6-2}$  design shown above has length 3, and it is therefore a resolution III design. In general, if the number of runs in two designs is the same, the design with the higher resolution is preferred.

In a **resolution  $R$  design** no effect involving  $i$  factors is aliased with effects of order less than  $R - i$ . For example, in designs with resolution  $V$ , main effects are aliased with four-factor interactions and higher order interactions, and two-factor interactions are aliased with three-factor interactions and higher order interactions. Therefore if all three-factor and higher order interactions can be assumed negligible, all main effects and two-factor interactions can be estimated from a resolution V design. In resolution IV designs, main effects are aliased with three-factor and higher order interactions. While in a resolution III design, main effects are confounded with two-factor interactions. Resolution III designs are normally used only in screening experiments where the purpose is to discover which factors are important enough to be studied further in follow-up experiments.

The **projective property** of a fractional factorial is another attribute that can be determined from its resolution. In a resolution R design, any subset of  $k = R - 1$  factors will form a full  $2^k$  design (with possible replication of some runs). Therefore, if an experiment is started with a resolution  $R$  fractional factorial design and only  $R - 1$  of the factors appear to be significant, then the data can be reanalyzed including only the  $R - 1$  significant factors. Since the design in these factors is a full factorial, interactions of all orders possible among the  $R - 1$  factors can be examined.

When two or more designs, created with different sets of generators, have the same number of runs and the same resolution, Fries and Hunter (1980) proposed another criterion for deciding which design is preferable. They called these criteria the **minimum aberration criteria**. If the number of words of length  $r$  in the defining relation of a design is defined to be  $A_r$ , then a design  $d_1$  is said to have less aberration than a design  $d_2$  if  $r$  is the smallest integer such that  $A_r(d_1) \neq A_r(d_2)$  and  $A_r(d_1) < A_r(d_2)$ . For example, if  $d_1$  is a resolution IV  $2^{7-2}$  design created with the generators  $F = ABCD$  and  $G = ABCE$ , it has less aberration than design  $d_2$ , created with the generators  $F = ABC$  and  $G = ADE$ , since the defining relation for  $d_1$  ( $I = ABCDF = ABCEG = DEFG$ ) has only one word of length 4, while the defining relation for  $d_2$  ( $I = ABCF = ADEG = BCDEFG$ ) has two words of length 4. For any  $k$

and  $p$  there is always a minimum aberration  $2^{k-p}$  design that has less aberration than any other  $2^{k-p}$  design. For two designs of the same resolution, the minimum aberration design will have less confounding of main effects with low order interactions and is generally preferred.

A final criterion that is useful when selecting the generators for a  $2^{k-p}$  design is the number of clear effects. Chen et al. (1993) define an effect to be clear if none of its aliases are main effects or two-factor interactions. In some cases, a design that is not the minimum aberration design may have clearer effects than the minimum aberration design. For example, Wu and Hamada (2000) explain that the  $2^{6-2}$  design with defining relation  $I = ABCE = ABDF = CDEF$  has all six main effects clear, while the  $2^{6-2}$  design with defining relation  $I = ABE = ACDF = BCDEF$  has three main effects (I, D, and F ) clear along with six two-factor interactions **BC**, **BD**, **BF**, **CE**, **DE**, and **EF**. In cases where some two-factor interactions are believed to be important a priori, the second design may be preferred over the first. Wu and Hamada's (2000) Table 4A lists the generators for the minimum aberration design and the design with the maximum number of clear effects (if different) for 8-run to 64-run designs.

The `FrF2` function in the R package `FrF2` can create minimum aberration designs. If the user does not specify generators when calling `FrF2`, like the example on page 277, `FrF2` automatically selects the set of generators that will result in a minimum aberration design. The code on the next page creates the minimum aberration  $2^{8-4}$  design, and the generators and aliases functions show the generators and alias pattern for the design `FrF2` created.

```
library(FrF2)
des1 <- FrF2( 16, 8 )
y <- runif( 16, 0, 1 )
library(DoE.base)
generators(des1)
```

```
$generators
[1] "E=ABC" "F=ABD" "G=ACD" "H=BCD"
```

```
aliases( lm( y ~ (.)^3, data = des1) )
```

A = B:C:E = B:D:F = B:G:H = C:D:G = C:F:H = D:E:H = E:F:G
B = A:C:E = A:D:F = A:G:H = C:D:H = C:F:G = D:E:G = E:F:H
C = A:B:E = A:D:G = A:F:H = B:D:H = B:F:G = D:E:F = E:G:H
D = A:B:F = A:C:G = A:E:H = B:C:H = B:E:G = C:E:F = F:G:H
E = A:B:C = A:D:H = A:F:G = B:D:G = B:F:H = C:D:F = C:G:H
F = A:B:D = A:C:H = A:E:G = B:C:G = B:E:H = C:D:E = D:G:H
G = A:B:H = A:C:D = A:E:F = B:C:F = B:D:E = C:E:H = D:F:H
H = A:B:G = A:C:F = A:D:E = B:C:D = B:E:F = C:E:G = D:F:G
A:B = C:E = D:F = G:H
A:C = B:E = D:G = F:H
A:D = B:F = C:G = E:H
A:E = B:C = D:H = F:G
A:F = B:D = C:H = E:G
A:G = B:H = C:D = E:F
A:H = B:G = C:F = D:E

FrF2 can also create designs with the maximum number of clear effects. For example, the call `FrF2(32,9)` produces the minimum aberration  $2^{9-4}$  design that has nine clear main effects and eight clear two-factor interactions. However, the call `FrF2(32,9,MaxC2=TRUE)` produces a  $2^{9-4}$  design that has all nine main effects and 15 clear two-factor interactions.

Consider the following example of the design and analysis of a  $2^{8-4}$  fractional factorial. AlmeidaeSilva et al. (2003) conducted an experiment to find the optimal conditions for culturing *Paecilomyces variotii* (a fungus commonly found in the air and soils of tropical countries) on eucalyptus *hemicellulosic hydrolyzate* with a view to producing microbial protein. Only 51.6% of the total dry mass of eucalyptus wood is utilized by Brazilian industry while the rest (branches, leaves, small trees, etc.) is left in the fields. The hemicellulose fraction of this waste can be easily removed by acid treatment, and the re-sulting hydrolyzate is rich in fermentable sugars. The fungus *P. variotii* was selected from among 21 species of yeasts and filamentous fungus for its performance on eucalyptus hemicellulose hydrolyzate. Protein biomass produced by this fungus during 72 hours of fermentation has an amino acid profile that is equal to or exceeds conventional products used for animal feed. The purpose of the experiments was to study the influence of inhibitors,

nutrients, and fermentation time on the biomass growth produced by *P. variotii*. **Table 6-5** shows the factors and levels that were to be studied.

**Table 6-5. Factors and Levels for Biomass Experiment**

Label	Factors	Levels	
		-	+
A	Inhibitors (Furfural and Acetic Acid)	1.25g/L	7.8g/L
B	Rice Bran	10.0g/L	30.0g/L
C	Urea	0.0g/L	2.0g/L
D	Magnesium Sulfate	0.0g/L	1.5g/L
E	Ammonium Sulfate	0.0g/L	2.0g/L
F	Potassium Nitrate	0.0g/L	2.0g/L
G	Sodium Phosphate	0.0g/L	2.0g/L
H	Fermentation Time	72 hrs	96 hrs

A  $2^{8-4}$  fractional factorial design was used with generators  $E = BCD$ ,  $F = ACD$ ,  $G = ABC$ , and  $H = ABD$ . This is the minimum aberration resolution IV design, and the clear effects in this design are the eight main effects. There are also seven aliased strings of two-factor interactions (shown below) that can be estimated.

$$\begin{aligned}
 & CG + DH + AB + EF \\
 & AC + BG + DF + EH \\
 & CF + AD + EG + BH \\
 & CH + DG + AE + BF \\
 & CD + GH + AF + BE \\
 & BC + AG + DE + FH \\
 & CE + FG + AH + BD
 \end{aligned}$$

The design as created by FrF2 is shown on the next page along with the resulting data.

```

library(FrF2)
culture <- FrF2( 16, generators =
  c("BCD", "ACD", "ABC", "ABD"), randomize = FALSE)
y1 <- c(5.75, 6.7, 11.12, 10.67, 4.92, 5.35, 2.81,
       10.83, 6.08, 7.27, 9.68, 4.2, 3.9, 3.78, 11.57,
       7.39 )
culture <- add.response( culture, y1 )

```

## culture

	A	B	C	D	E	F	G	H	y1
1	-1	-1	-1	-1	-1	-1	-1	-1	5.75
2	1	-1	-1	-1	-1	1	1	1	6.70
3	-1	1	-1	-1	1	-1	1	1	11.12
4	1	1	-1	-1	1	1	-1	-1	10.67
5	-1	-1	1	-1	1	1	1	-1	4.92
6	1	-1	1	-1	1	-1	-1	1	5.35
7	-1	1	1	-1	-1	1	-1	1	2.81
8	1	1	1	-1	-1	-1	1	-1	10.83
9	-1	-1	-1	1	1	1	-1	1	6.08
10	1	-1	-1	1	1	-1	1	-1	7.27
11	-1	1	-1	1	-1	1	1	-1	9.68
12	1	1	-1	1	-1	-1	-1	1	4.20
13	-1	-1	1	1	-1	-1	1	1	3.90
14	1	-1	1	1	-1	1	-1	-1	3.78
15	-1	1	1	1	1	-1	-1	-1	11.57
16	1	1	1	1	1	1	1	1	7.39

```
class=design, type= FrF2.generators
```

The R function `lm` is used to fit a model to the data. The code and results are shown below. The expression `formula = y(.)^2` in the call to the `lm` function causes it to fit a saturated model in the main effects and two-factor interactions. However, in this example there are 28 two-factor interactions but only seven of them are estimable since they are confounded in strings of four interactions. The `lm` function estimates all the two-factor interactions with *A*, that is, *AB*, *AC*, *AD*, *AE*, *AF*, *AG*, and *AH*. The estimates for the remaining two-factor interactions are labeled `NA` in the output and are not shown on the next page. Referring to the strings of aliased two-factor interactions on page 210, it can be seen that *AB* actually represents *CG + DH + AB + EF*, and so forth.

```
modf <- lm( y1 ~ (.)^2, data = culture)
summary(modf)
```

Call:

```
lm.default(formula = y1 ~ (.)^2, data = culture)
```

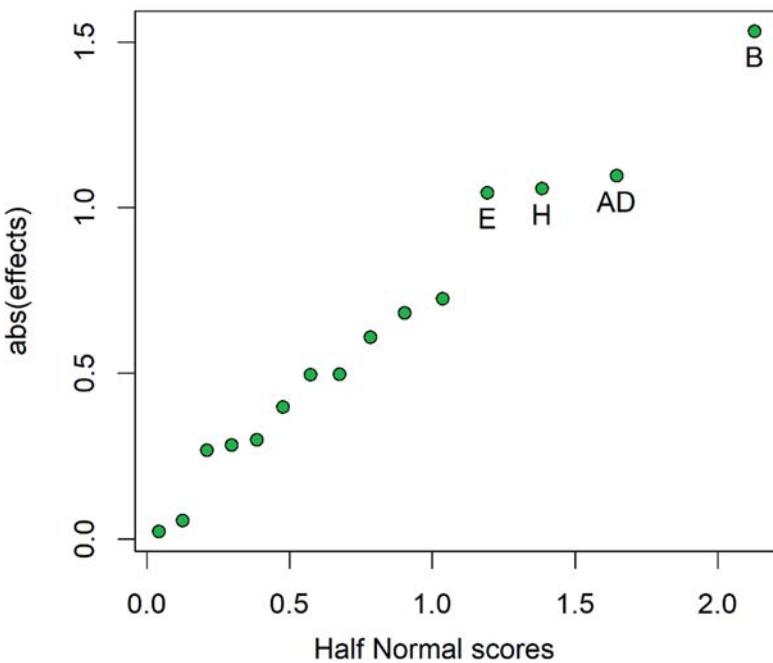
Residuals:

```
ALL 16 residuals are 0: no residual degrees of freedom!
```

```
Coefficients: (21 not defined because of singularities)
Estimate Std. Error t value Pr(>|t|)
```

(Intercept)	7.00125	NaN	NaN	NaN
A1	0.02250	NaN	NaN	NaN
B1	1.53250	NaN	NaN	NaN
C1	-0.68250	NaN	NaN	NaN
D1	-0.26750	NaN	NaN	NaN
E1	1.04500	NaN	NaN	NaN
F1	-0.49750	NaN	NaN	NaN
G1	0.72500	NaN	NaN	NaN
H1	-1.05750	NaN	NaN	NaN
A1:B1	-0.28375	NaN	NaN	NaN
A1:C1	0.49625	NaN	NaN	NaN
A1:D1	-1.09625	NaN	NaN	NaN
A1:E1	-0.39875	NaN	NaN	NaN
A1:F1	0.60875	NaN	NaN	NaN
A1:G1	0.29875	NaN	NaN	NaN
A1:H1	-0.05625	NaN	NaN	NaN
B1:C1	NA	NA	NA	NA
B1:D1	NA	NA	NA	NA
B1:E1	NA	NA	NA	NA
B1:F1	NA	NA	NA	NA
B1:G1	NA	NA	NA	NA
B1:H1	NA	NA	NA	NA
C1:D1	NA	NA	NA	NA
C1:E1	NA	NA	NA	NA
C1:F1	NA	NA	NA	NA
C1:G1	NA	NA	NA	NA
C1:H1	NA	NA	NA	NA
D1:E1	NA	NA	NA	NA
D1:F1	NA	NA	NA	NA
D1:G1	NA	NA	NA	NA
D1:H1	NA	NA	NA	NA
E1:F1	NA	NA	NA	NA
E1:G1	NA	NA	NA	NA
E1:H1	NA	NA	NA	NA
F1:G1	NA	NA	NA	NA
F1:H1	NA	NA	NA	NA
G1:H1	NA	NA	NA	NA

Residual standard error: NaN on 0 degrees of freedom  
 Multiple R-squared: 1, Adjusted R-squared: NaN  
 F-statistic: NaN on 15 and 0 DF, p-value: NA



**Figure 6-6. Half-Normal Plot of Effects from  $2^{8-4}$  *Paecilomyces variotii* Culture Experiment**

The authors of the article felt that the experiment had given them evidence that factor A (the inhibitor) had little effect, and they confirmed this by citing other published reports. They also felt the experiment showed main effects D (magnesium sulfate) and F (potassium nitrite) were insignificant. However, due to the confounding of two-factor interactions with the second largest effect (in absolute value) and the fact that nothing clearly stuck out on the half-normal plot of effects, no definite conclusions could be drawn. They decided to run another resolution V  $2^{5-1}$  follow-up experiment using factors B, C, E, G, and H with defining relation  $I = BCEGH$  and factors A, D, and F held constant at the mid-point of the levels used in the first experiment. This was a 16-run design, but if it could be safely assumed that main effects A, D, and F were negligible in the first set of sixteen experiments, then eight of the sixteen runs for the proposed follow-up design were already completed.

The  $2^{5-1}$  design as created by FrF2 is shown on the next page along

with the resulting data. Run numbers 3, 4, 7, 8, 11, 12, 15, and 16 were already completed in the  $2^{8-4}$  design as runs 14, 7, 6, 15, 13, 8, 12, and 16, respectively. The remaining eight experiments were completed in random order to get the results shown. Since this design is resolution V, all main effects and two-factor interactions are clear and can be estimated if three-factor and higher order interactions can be assumed negligible.

```
library(FrF2)
culture2 <- FrF2( 16, 5, factor.names =
c("B", "C", "E", "G", "H"), randomize = FALSE)
y <- c(3.37, 3.55, 3.78, 2.81, 5.53, 10.43, 5.35, 11.57,
2.93, 7.23, 3.9, 10.83, 11.69, 10.59, 4.92, 7.39)
culture2 <- add.response( culture2, y )
culture2
```

	B	C	E	G	H	y
1	-1	-1	-1	-1	1	3.37
2	1	-1	-1	-1	-1	3.55
3	-1	1	-1	-1	-1	3.78
4	1	1	-1	-1	1	2.81
5	-1	-1	1	-1	-1	5.53
6	1	-1	1	-1	1	10.43
7	-1	1	1	-1	1	5.35
8	1	1	1	-1	-1	11.57
9	-1	-1	-1	1	-1	2.93
10	1	-1	-1	1	1	7.23
11	-1	1	-1	1	1	3.90
12	1	1	-1	1	-1	10.83
13	-1	-1	1	1	1	11.69
14	1	-1	1	1	-1	10.59
15	-1	1	1	1	-1	4.92
16	1	1	1	1	1	7.39

Figure 6.7 shows the half-normal plot of effects from the follow-up design. The results of this follow-up experiment suggest that main effects, *B*-Rice Bran, *G*-Sodium Phosphate, and *E*-Ammonium Sulfate along with interactions *CH* (Urea × Fermentation Time) and *BH* (Rice Bran × fermentation Time) and *CE* (Urea × Ammonium Sulfate) appear to be significant. Since interactions exist, the main effects should not be interpreted in isolation.

```

library(daewr)
cfs <- coef(moda)[2:16]
effects <- cfs
names <- names(cfs)
halfnorm(cfs, names, alpha = .219, refline = FALSE)

```

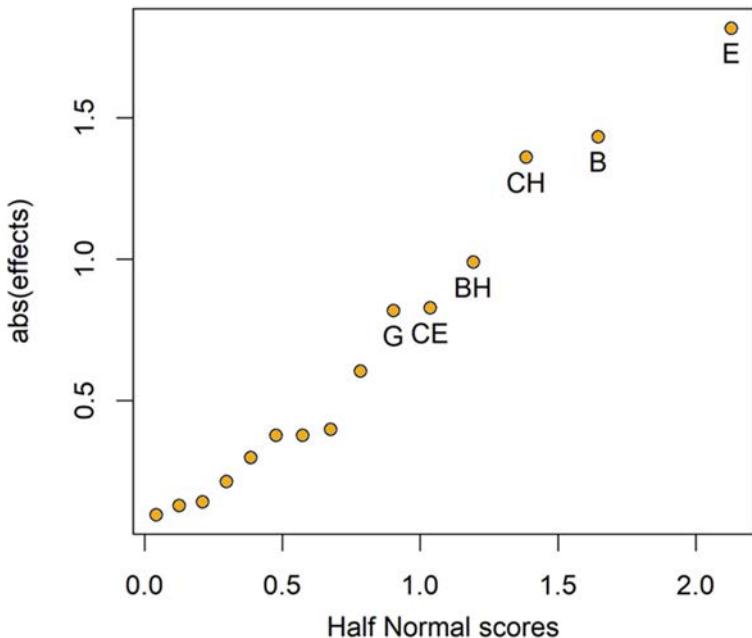


Figure 6-7. Half-Normal Plot of Effects from  $2^{5-1}$  *Paecilomyces variotii* Culture Experiment

Figure 6.8 shows the *BH* interaction plot. It shows how the effect of fermentation time depends upon the level of rice bran. When there is only 10 g/L of rice bran in the growth medium, it can be seen that increasing the fermentation time from 72 to 96 hrs increases the biomass produced. However, if there are 30 g/L of rice bran in the growth medium, increasing the fermentation time actually decreases the biomass produced.

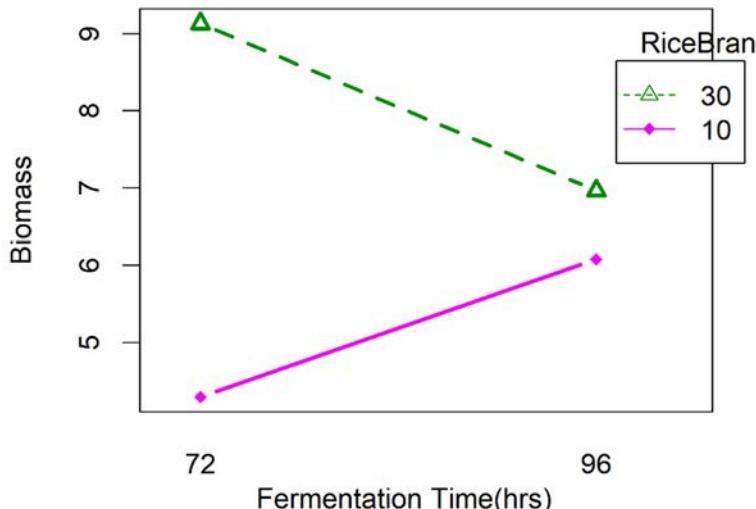
```

RiceBran <- 10*((as.numeric(culture2$B)-1.5) /.5) + 20
FermTime <- 12*((as.numeric(culture2$H)-1.5) / .5) + 84
interaction.plot(FermTime, RiceBran, culture2$y,
type = "b", pch=c(18,24,22), leg.bty="o",
main="Interaction Fermentation Time and"

```

```
Rice Bran Level", xlab = "Fermentation Time(hrs)",
ylab = "Biomass", lwd = 2.5, lty = 1:2,
col=c("magenta2","green4"))
```

### Interaction Fermentation Time and Rice Bran Level



*Figure 6-8. Interaction Plot for Fermentation Time and Level of Rice Bran*

Figure 6.9 shows the *CE* interaction plot. It shows the effect of ammonium sulfate upon biomass production depends upon the level of urea. Here it can be seen that increasing the level of ammonium sulfate from 0.0 to 2.0 g/L causes a greater increase in biomass when there is no urea added to the nutrients than when 2.0 g/L of urea is added to the nutrients. The maximum biomass occurs when there is 2 g/L of ammonium sulfate and no urea added to the nutrients.

```
Urea <- 1*((as.numeric(culture2$C)-1.5) / .5) + 1
AmonSulf <- 1*((as.numeric(culture2$E)-1.5) / .5) + 1
interaction.plot(AmonSulf, Urea, culture2$y,
type = "b", pch=c(18,24,22), leg.bty="o",
main="Interaction of
Urea and Ammonium Sulfate", xlab = "Ammonium
Sulfate(g/L)", ylab = "Biomass", lwd = 2.5,
lty = 1:2, col=c("aquamarine2","darkviolet"))
```

### Interaction Plot of Urea and Ammonium Sulfate

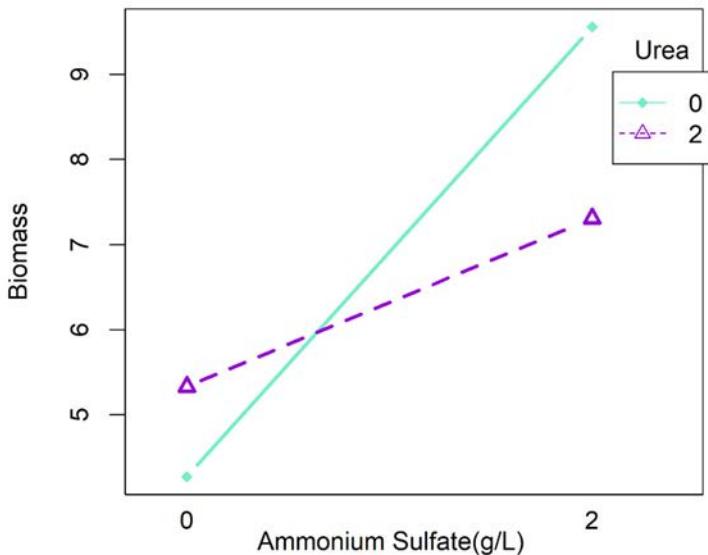


Figure 6-9. Interaction Plot for Urea and Ammonium Sulfate

**Figure 6-10** shows the interaction plot of fermentation time and urea, CH. On average, it can be seen that increasing the fermentation time seems to have little effect on biomass production. Also, on average, adding urea to the growth medium seems to have little effect on the biomass produced. However, as can be seen in the graph, the effect of fermentation time upon biomass depends upon whether urea is present, and its effect appears to be exactly opposite depending on whether 2.0 g/L of urea is added to the growth medium.

```
Urea <- 1*((as.numeric(culture2$C)-1.5) / .5) + 1
FermTime <- 12*((as.numeric(culture2$H)-1.5) / .5) + 84
interaction.plot(FermTime, Urea, culture2$y,
                 type = "b",
                 pch=c(18,24,22), leg.bty = "o", main = "Interaction of
                 Fermentation Time and Urea", xlab = "Fermentation
                 Time(hrs)", ylab = "Biomass", lwd = 2.5, lty = 1:2,
                 Col = c("dodgerblue","magenta"))
```

### Interaction Plot of Fermentation Time and Urea

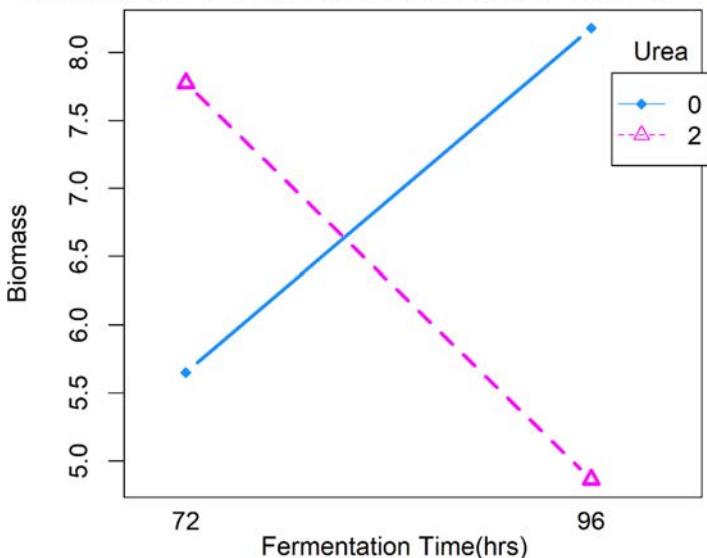


Figure 6-10. Interaction Plot of Fermentation Time and Urea

While it is possible to have an interaction between two factors that do not have significant main effects (like the example shown in **Figure 6.9**), it is rare. In Li et al.'s (2006) study of 113 published factorial experiments this happened less than 1% of the time. Usually interactions occur between factors where at least one of the two main effects are significant. This has been described as the effect heredity principle by Hamada and Wu (1992). In this experiment, since the two-factor interaction between fermentation time and urea is confounded with the three-factor interaction between rice bran, ammonium sulfate, and sodium phosphate (i.e.,  $CH = BEG$ ), and all three of the latter factors have significant main effects, it is possible that the large effect labeled as CH on the normal plot is actually the three-factor interaction. In this case, the effect heredity principle may overshadow the hierarchical ordering principle.

A three-factor interaction means that the effect of one factor depends upon the combination of levels of two other factors. A series of interaction plots (like those in the example in Section 3.6) are useful for explaining or interpreting a three-factor interaction. Figure 6.10 shows the effect of ammonium sulfate upon biomass at the four combinations

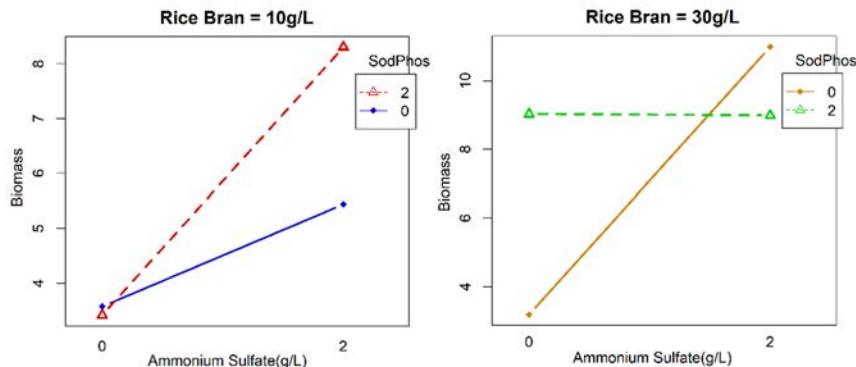
of rice bran and sodium phosphate.

```

culture3 <- culture2[culture2$B == -1,]
SodPhos <- 1*((as.numeric(culture3$G)-1.5) / .5) + 1
AmonSulf <- 1*((as.numeric(culture3$E)-1.5) / .5) + 1
interaction.plot(AmonSulf, SodPhos, culture3$y,
  type = "b", pch = c(18,24,22),
  leg.bty = "o",
  main="Rice Bran = 10g/L",
  xlab = "Ammonium Sulfate(g/L)", ylab = "Biomass",
  lwd = 2.5, lty = 1:2, col=c("blue","red"))

culture4 <- culture2[culture2$B == 1,]
SodPhos <- 1*((as.numeric(culture4$G)-1.5) / .5) + 1
AmonSulf <- 1*((as.numeric(culture4$E)-1.5) / .5) + 1
interaction.plot(AmonSulf, SodPhos, culture4$y,
  type = "b", pch=c(18,24,22), leg.bty="o",
  main="Rice Bran = 30g/L",
  xlab = "Ammonium Sulfate(g/L)", ylab = "Biomass",
  lwd=2.5,lty=1:2, col=c("orange","green"))

```



**Figure 6-11. Interaction Plots to Interpret Three-Factor Ammonium Sulfate by Rice Bran by Sodium Phosphate Interaction**

If this three-factor interaction is assumed to be important, the two-factor interactions do not tell the whole story. In Figure 6.10 it can be seen that adding 2.0 g/L of ammonium sulfate to the growth medium increases the biomass produced in general. However, this effect is greatest when there is 30 g/L of rice bran and no sodium phosphate in the growth medium. The optimum result appears to be with a

fermentation time of 72 hours, 30 g/L of rice bran, 2.0 g/L of ammonium sulfate, and 0.0 g/L of urea in the growth medium. There the biomass yield was predicted to be 11.57 g/L.

One way to validate the model predictions is to run one confirmation experiment at the predicted optimal conditions. However, the authors of the article instead chose to continue experimenting with three of the four factors (rice bran, ammonium sulfate, and fermentation time) to further increase biomass protein. They found conditions producing over 12.5 g/L, and these results will be shown in Chapter 10 exercises.

To recap this example, the researchers started with eight factors with two levels each as shown in Table 6.5. It would have required  $2^8 = 256$  experiments to complete a full factorial experiment. The effect sparsity principle and the hierarchical ordering principle suggest that this many experiments is probably not needed. With eight factors, it is doubtful that all main effects and a large number of interactions would be significant. A series of two fractional factorial experiments, employing a total of 24 runs, revealed that four of the main effects are significant and at most three interactions. Interaction graphs helped make a plausible interpretation of the results and identification of factor levels for optimal results. The interpretation of the interactions was critical for identification of optimal results and would never have been discovered if a seat-of-the-pants approach or vary one-factor-at-a-time plan had been utilized.

## 6.5. Augmenting Fractional Factorials

The last example introduced the idea of augmenting fractional factorial designs with additional experiments. Sometimes this consists of a simple confirmation experiment to validate a model prediction. In other cases additional experiments may be performed to deconfound certain main effects or interactions and increase the odds of discovering the correct model. In this section some formal procedures will be described that allow this to be done. **Section 6.5.1** describes procedures that will preserve the optimal orthogonality property of the augmented design. **Section 6.5.2** describes a procedure that reduces the number of additional experiments but does not preserve orthogonality.

### 6.5.1. Optional: rand seed settings

Consider setting a seed to make this script fully reproducible. Go to Advanced->Set Random Number Generator Seed, click the checkbox, and set Random Seed to any whole number.

### 6.5.2. Augmenting by Foldover or Mirror Image Designs

In a resolution III fractional factorial design, main effects are confounded with some two-factor interactions. If more than one effect appears to be significant after the analysis of data from a resolution III design, it is not clear whether all effects are due to main effects or whether some could be two-factor interactions. For example, in an eight-run resolution III design in six factors (designated by  $2^{6-3}_{III}$ ) with generators  $D = AB$ ,  $E = AC$ , and  $F = BC$ , the defining relation is

$$I = ABD = ACE = BCF = DEF = BCDE = ACDF = ABEF.$$

If after analysis of data from this design, effects  $B$  and  $D$  appear to be significant, it could be that the two main effects are the active effects. However, the effect heredity principle tells us there are two alternative explanations. The first alternative is the following. Since  $D$  is confounded with  $AB$  and  $EF$ , it could be that main effect  $B$  and the  $AB$  interaction are the active effects and that  $D$  only appears large because it is confounded with  $AB$ . The second alternative is that main effect  $D$  and the  $AD$  interaction are the active effects and  $B$  only appears large because it is confounded with  $AD$  and  $CF$ . With-out additional experiments, there is no way to determine which of the three explanations of the data is correct.

One way to break the confounding between main effects and two-factor interactions is to run an additional set of experiments that is the same as the first except that the coded factor levels on one or more factors has been reversed (Box, Hunter, & Hunter, 1978). This is called a foldover fraction. For example, in the  $2^{6-3}_{III}$  described in the last paragraph, if the signs were reversed for factor  $B$ , the defining relation would become  $I = -ABD = ACE = -BCF = DEF = -BCDE = ACDF = -ABEF$ . If the two eight-run designs were combined, as shown in Table 6.6, the resulting 16-run design would have defining relation  $I = ACE = DEF = ACDF$ . Although this is still a resolution III design, main effect  $B$  is clear,

and main effect  $D$  is no longer confounded with the  $AB$  interaction.

**Table 6-6.  $2^{6-3}_{III}$  Design Augmented by  $2^{6-3}_{III}$  Design with Signs Reversed on Factor B**

Run	A	B	C	D	E	F
1	-	-	-	+	+	+
2	+	-	-	-	-	+
3	-	+	-	-	+	-
4	+	+	-	+	-	-
5	-	-	+	+	-	-
6	+	-	+	-	+	-
7	-	+	+	-	-	+
8	+	+	+	+	+	+
9	-	+	-	+	+	+
10	+	+	-	-	-	+
11	-	-	-	-	+	-
12	+	-	-	+	-	-
13	-	+	+	+	-	-
14	+	+	+	-	+	-
15	-	-	+	-	-	+
16	+	-	+	+	+	+

In general, augmenting a resolution III design with another resolution III design, with the coded factor levels for one factor reversed, will make that factor and all of its two-factor interactions clear of other two-factor interactions. Montgomery and Runger (1996) show the defining relation for the combined design from the original plus foldover will contain those effects in the original design that were not sign reversed in the foldover fraction.

If a resolution III design is augmented with another resolution III design, with the coded factor levels reversed for all factors (called the **mirror image design**), the combined design will be resolution IV and all main effects will be clear of two-factor interactions. The defining relation for the combined design in this situation will only contain even length words from the original defining relation. For example, the  $2^{5-2}_{III}$  design with generators  $D = AB$ ,  $E = AC$  has defining relation  $I = ABD = ACE = BCDE$ . If it is combined with its mirror image fraction where the signs of all the coded factor levels have been reversed, the defining relation of the combined design will be  $I = BCDE$ .

The R `FrF2` package `fold.design` function can augment a design with its mirror image as shown in the example on the next page. This function can also augment with a foldover fraction by changing the `columns` option from 'full' to a character vector of factor names or a numeric vector of factor positions that should have their signs reversed.

```
library(FrF2)
des <- FrF2(8, 6, generators = c("AB", "AC", "BC"),
randomize = FALSE)
desa <- fold.design(des, columns = 'full')
desa
```

	A	B	C	fold	D	E	F
1	-1	-1	-1	original	1	1	1
2	1	-1	-1	original	-1	-1	1
3	-1	1	-1	original	-1	1	-1
4	1	1	-1	original	1	-1	-1
5	-1	-1	1	original	1	-1	-1
6	1	-1	1	original	-1	1	-1
7	-1	1	1	original	-1	-1	1
8	1	1	1	original	1	1	1
9	1	1	1	mirror	-1	-1	-1
10	-1	1	1	mirror	1	1	-1
11	1	-1	1	mirror	1	-1	1
12	-1	-1	1	mirror	-1	1	1
13	1	1	-1	mirror	-1	1	1
14	-1	1	-1	mirror	1	-1	1
15	1	-1	-1	mirror	1	1	-1
16	-1	-1	-1	mirror	-1	-1	-1

class=design, type= FrF2.generators.folded

When creating a resolution IV design by combining a resolution III design with its mirror image (i.e., signs reversed on all factors), it is possible to add one additional blocking factor that can account for any difference in the average response between the two sets of experiments. By having an additional factor, the experiments can be performed sequentially. The original resolution III design is completed first and the data analyzed. If there is only one factor that appears to be significant, there may be no need for additional experiments. If, on the other hand, two or more effects appear significant, the mirror image design can be completed to clear all main effects from two-factor interactions. The additional factor can account for any changes in the experimental environment that have

occurred since the first set of experiments. Including it in the analysis will prevent any unanticipated changes from biasing the effects of the factors studied.

Augmenting a resolution IV design with its mirror image design will not help to break strings of aliased two-factor interactions since the signs will not change on the two-factor interactions. However, augmenting a resolution IV design by a foldover fraction can be used to break strings of aliased two-factor interactions, when the signs are changed on only one or two of the factors. For example, the  $2^{8-4}_{IV}$  design with  $E = BCD$ ,  $F = ACD$ ,  $G = ABC$ , and  $H = ABD$  has seven strings of confounded two-factor interactions in groups of 4, i.e.,

$$\begin{aligned} & AB + EF + CG + DH \\ & AC + DF + BG + EH \\ & CD + BE + AF + GH \\ & AD + CF + EG + BH \\ & AE + BF + DG + CH \\ & BC + DE + AG + FH \\ & BD + CE + FG + AH. \end{aligned}$$

If this design is augmented with another design where the signs are reversed on factor  $A$ , all the two-factor interactions involving factor  $A$  will be clear of other two-factor interactions.

In general, Montgomery and Runger (1996) show that the alias sets for the combined design are obtained from a partition of the alias sets in the original fraction by grouping effects that are sign reversed. For example, in the  $2^{6-2}_{IV}$  design with generators  $E = ABC$  and  $F = BCD$ , the aliases for the 16 effects that can be estimated (up to three-factor interactions) are:

$$\begin{aligned} & I + ABCE + BCDF + ADEF \\ & A + BCE + DEF \\ & B + ACE + CDF \\ & C + BDF + ABE \\ & D + BCF + AEF \\ & E + ABC + ADF \\ & F + BCD + ADE \\ & CE + AB \end{aligned}$$

$$\begin{aligned}
& AC + BE \\
& AD + EF \\
& BC + AE + DF \\
& DE + AF \\
& CF + BD \\
& CD + BF \\
& ABD + CDE + ACF + BEF \\
& ACD + BDE + CEF + ABF
\end{aligned}$$

If this design were augmented by the foldover design with all signs for factor A reversed, 32 effects could be estimated, and the aliases for each of them could be determined from the list above. Every effect containing A changes signs.

Group effects with like signs together to get the following 32 groups.

$$\begin{aligned}
& ABCE + ADEF, I + BCDF \\
& A, BCE + DEF \\
& ACE, B + CDF \\
& ABE, C + BDF \\
& AEF, D + BCF \\
& ABC + ADF, E \\
& ADE, F + BCD \\
& AB, CE \\
& AC, BE \\
& AD, EF \\
& AE, BC + DF \\
& AF, DE + \\
& ACDE + ABEF, CF + BD \\
& ABDE + ACEF, CD + BF \\
& ABD + ACF, CDE + BEF \\
& ACD + ABF, BDE + CEF
\end{aligned}$$

If the  $2^{6-2}_{IV}$  design were augmented by the same design with signs reversed on factors A and B, effects containing a single A or a single B will reverse signs. Effects containing both A and B or neither A nor B will not reverse signs, and the aliases for the 32 estimable effects will be different than those shown above. When only a few effects are found significant after analysis of data from a resolution IV design, a foldover

that will separate the significant main effects from two-factor interactions (that involve the significant main effects) can usually be found.

Consider two designs used to optimize a drinking water filter to illustrate the augmenting of a fractional factorial to estimate interactions. High concentrations of arsenic are reported in ground water in countries such as Argentina, Bangladesh, Chile, China, India, Japan, Mexico, Mongolia, Nepal, Poland, Taiwan, Vietnam, and some parts of the United States, and studies have shown people exposed to high levels of arsenic are prone to develop various forms of cancer. Iron oxide coated sand (IOCS) has been used to remove arsenic from ground water in simple household filtration systems in Bangladesh. Ramakrishna et al. (2006) conducted a study with the objective of systematically studying the effects of several factors on the arsenic removal rate for IOCS.

The experiments consisted of making a coating solution composed of ferric nitrate and sodium hydroxide, with NaOH added to adjust the pH. This solution was aged and then poured over clean dry sand and mixed. The mixture was dried for 12 hours, and then used to filter water solutions spiked with a known concentration of arsenic. The response was the percentage of arsenic removed, and the factors varied in the study are shown in **Table 6-7** below.

**Table 6-7. Factors and Levels for Arsenic Removal Experiment**

Label	Factors	Levels	
		-	+
A	coating pH	2.0	12.0
B	drying temperature	110°	800°
C	Fe concentration in coating	0.1 M	2 M
D	number of coatings	1	2
E	aging of coating	4 hrs	12 days
F	pH of spiked water	5.0	8.0
G	mass of adsorbent	0.1 g	1 g

A  $2^{7-4}_{III}$  fraction factorial design, with generators  $D = AB$ ,  $E = AC$ ,  $F = BC$ , and  $G = ABC$ , was used for the first set of experiments. The results are shown in the top half of **Table 6-8**. These experiments are shown in

the standard order, not the random order in which they were run.

**Table 6-8.  $2^{7-4}_{III}$  Design Augmented by Foldover  $2^{7-4}_{III}$  with Signs Reversed on all Factors**

Run	Block	A	B	C	D	E	F	G	% Removal of As
1	1	-	-	-	+	+	+	-	69.95
2	1	+	-	-	-	-	+	+	58.65
3	1	-	+	-	-	+	-	+	56.25
4	1	+	+	-	+	-	-	-	53.25
5	1	-	-	+	+	-	-	+	94.4
6	1	+	-	+	-	+	-	-	73.45
7	1	-	+	+	-	-	+	-	10
8	1	+	+	+	+	+	+	+	2.11
9	2	+	+	+	-	-	-	+	16.2
10	2	-	+	+	+	+	-	-	52.85
11	2	+	-	+	+	-	+	-	9.05
12	2	-	-	+	-	+	+	+	31.1
13	2	+	+	-	-	+	+	-	7.4
14	2	-	+	-	+	-	+	+	9.9
15	2	+	-	-	+	+	-	+	10.85
16	2	-	-	-	-	-	-	-	48.75

A half-normal plot of the effects calculated from the first eight runs (left as an exercise) revealed that factors *B* (temperature of drying the coated sand) and *F* (pH of the arsenic spiked water) appeared to be significant. Both factors had negative effects and the high level of each resulted in a lower percentage of arsenic removal. This implies that the percent arsenic removed is only influenced by the temperature of drying the coated sand and by the pH of the spiked water solution. However, interpreting the results to mean that only the two main effects are active could be misleading. *B* is confounded with three two-factor interactions *AD* + *CF* + *EG* and *F* is confounded with *BC* + *DE* + *AG*. According to the effect heredity principle, the model including *B* and *BC* (which imply arsenic removal is influenced by the temperature of drying the coated sand and by the *Fe* concentration in the coating), and the model including *F* and *CF* (which imply arsenic removal is influenced by the pH of the spiked water and by the *Fe* concentration in the coating) are also plausible explanations of the data. From the eight experiments in the  $2^{7-3}_{III}$  it is impossible to tell which of the three plausible models is

appropriate since the effects are completely confounded.

The original eight experiments were augmented with the mirror image design, shown in the bottom half of **Table 6-8**, in order to deconfound the main effects from strings of two-factor interactions. An additional blocking factor is included in the combined design to account for any difference in the two sets of experiments. A half-normal plot of the effects calculated from the combined experiments (left as an exercise) revealed that the blocking factor representing the difference in the two groups of experiments, main effect  $F$  (pH of the arsenic spiked water), main effect  $B$  (temperature of drying the coated sand), main effect  $A$  (pH of the coating), and the effect representing the confounded string of interactions  $AD + CF + EG$  all appeared to be significant. The effect heredity principle would suggest two plausible models to explain the data (ignoring the block term). The first model is ( $F, B, A, AD$ ), or:

$$\begin{aligned}\% \text{ removal} = & 37.76 - 12.99 \left( \frac{\text{pH } s - 7.0}{2.0} \right) - 11.76 \left( \frac{\text{temp} - 455^\circ}{345^\circ} \right) \\ & - 8.89 \left( \frac{\text{pH } c - 7.0}{5.0} \right) \\ & - 10.09 \left( \frac{\text{pH } s - 7.0}{2.0} \right) \left( \frac{\text{number coats} - 0.75}{0.5} \right)\end{aligned}$$

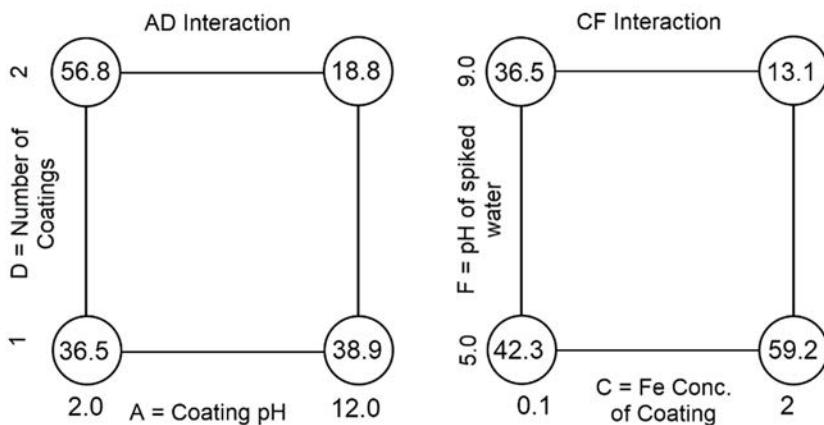
This model suggests that increasing the pH of the ferric nitrate sodium hydroxide solution used to coat the sand has little effect on arsenic removal when the sand is only coated once, but that it substantially decreases the arsenic removal when the sand is coated twice. This can be visualized in the left side of *In Figure 6-16, we can see that the best model with two parameters includes main effect  $F$ , interaction  $FG$  and has an adjusted  $R^2=0.8686$ .*

Table 6-12. If this model is correct, it would imply that maximum arsenic removal in a sand filtration system can be achieved when the coated sand is dried at the low temperature ( $B = -$ ), the coating solution has a low pH ( $A = -$ ), and the sand is coated twice ( $D = +$ ).

The second plausible model is ( $F, B, A, CF$ ), or:

$$\% \text{ removal} = 37.76 - 12.99 \left( \frac{\text{pH } s - 7.0}{2.0} \right) - 11.76 \left( \frac{\text{temp} - 455^\circ}{345^\circ} \right) \\ - 8.89 \left( \frac{\text{pH } c - 7.0}{5.0} \right) \\ - 10.09 \left( \frac{\text{Fe} - 1.05M}{0.95M} \right) \left( \frac{\text{pH } s - 7.0}{2.0} \right)$$

This model suggests that increasing the Fe concentration in the coating solution increases the arsenic removal when the pH of the water being filtered is low but decreases it when the pH of the water being filtered is high. This can be visualized in the right side of **Figure 6-12**. If this model is correct, it would imply that maximum arsenic removal in a sand filtration system can be achieved when the coated sand is dried at the low temperature ( $B = -$ ), and when the Fe concentration in the coating solution is high if the pH of the water to be filtered is low, or when the Fe concentration in the coating solution is low if the pH of the water to be filtered is high.



**Figure 6-12. Plausible Interaction Plots for Arsenic Removal Experiment**

The two models are contradictory. The first model implies that the pH of the coating solution and the number of times the sand is coated influence the percent of arsenic the sand filter can remove. However, the second model implies the pH of the coating solution and the number of times the sand is coated does not affect arsenic removal; rather, the Fe concentration of the coating solution does matter. The optimum level of this concentration must be determined by the pH of the water to be

filtered.

In 16 experiments the researchers were able to narrow down the list of factors that influence arsenic removal from the original seven down to five. But, because the two interactions  $AD$  and  $CF$  are completely confounded, there is no way from the results of these two sets of experiments to determine which model is correct and what is the best way to coat sand for a filtration system. If another foldover fraction of 16 experiments were run reversing the signs on one of the factors  $A$ ,  $D$ ,  $C$ , or  $F$ , the combined set of 32 experiments would allow for estimating all the terms ( $F$ ,  $B$ ,  $A$ ,  $AD$ , and  $CF$ ), and a reliable model could be established. The next section will describe another way of augmenting the data to estimate all the terms with fewer additional experiments.

### 6.5.3. Augmenting by Optimal Design

In matrix notation the model with the terms ( $F$ ,  $B$ ,  $A$ ,  $AD$ , and  $CF$ ), from the last section, and the 16-run design shown in **Table 6-8** can be written as:

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

where

$$\mathbf{y} = \begin{pmatrix} 69.95 \\ 58.65 \\ 56.25 \\ 53.25 \\ 94.40 \\ 73.45 \\ 10.00 \\ 2.11 \\ 16.20 \\ 52.85 \\ 9.05 \\ 21.10 \\ 7.40 \\ 9.90 \\ 10.85 \\ 45.75 \end{pmatrix}, \mathbf{X} = \begin{pmatrix} 1 & -1 & -1 & -1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 \\ 1 & -1 & 1 & 1 & -1 & 1 & 1 \\ 1 & -1 & -1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & -1 \\ 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & -1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 & -1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & 1 & 1 \end{pmatrix}, \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_{bl} \\ \beta_A \\ \beta_B \\ \beta_F \\ \beta_{AD} \\ \beta_{CF} \end{pmatrix}$$

This model cannot be fit to the data since the last two columns in the  $\mathbf{X}$  matrix are identical, and the least squares normal equations are singular. However, it is not necessary to double the number of runs, which would

be required if augmented by a foldover, in order to estimate all of the parameters in the model. Doubling the number of runs could preserve the orthogonality property so that all estimated effects would be uncorrelated. However, any additional runs that would make the  $\mathbf{X}'\mathbf{X}$  matrix nonsingular will allow estimation of all the effects of interest.

One criterion that can be checked, to ensure the  $\mathbf{X}'\mathbf{X}$  matrix nonsingular, is  $|\mathbf{X}'\mathbf{X}|$ , and Dykstra (Dykstra, 1971) has shown that choosing a set of additional runs that will maximize  $|\mathbf{X}'\mathbf{X}|$  for the combined set of runs is called the **D-optimal design**. The motivation for *D-optimal* designs comes from the fact that the variance covariance matrix of the regression coefficients in the linear model is  $\sigma^2(\mathbf{X}'\mathbf{X})^{-1}$  and the reciprocal of the volume of a confidence ellipsoid for  $\hat{\beta}$  for is proportional to the determinant of  $\mathbf{X}'\mathbf{X}$ . Orthogonal designs are always *D-optimal* and have the maximum  $|\mathbf{X}'\mathbf{X}|$  for their run size, but when the experimenter is willing to forgo uncorrelated effect estimates in order to reduce the number of runs required, a *D-optimal* design for a reduced run size will be the best choice.

Another criterion that can be used to choose additional runs is the  $tr(\mathbf{X}'\mathbf{X})^{-1}$ , since the variances of the regression coefficients are the diagonal elements of  $\sigma^2(\mathbf{X}'\mathbf{X})^{-1}$ . A design that minimizes  $tr(\mathbf{X}'\mathbf{X})^{-1}$  is said to be **A-optimal**, since it minimizes the average variance of the regression coefficients.

The `optFederov` function in the R `AlgDesign` package (Wheeler B., 2012) can find both *D-optimal* and *A-optimal* designs when given the model and a list of candidate design points. This procedure uses Federov's (1972) algorithm to find the subset of the candidate design points that will maximize  $|\mathbf{X}'\mathbf{X}|$  or minimize  $tr(\mathbf{X}'\mathbf{X}^{-1})$ .

To illustrate how this function can be used to augment an existing design, consider augmenting the design for the arsenic removal experiments (shown in **Table 6-7**) so that both the *AD* and *CF* interactions can be included in the model. The design shown in Table 6.8 was created with a modification of the code used earlier to produce a combined file (original plus foldover). This is shown again in the code below, and the combined file is stored in the design data frame `augm`.

```

library(FrF2)
des2 <- FrF2( 8, 7, generators = c("AB", "AC", "BC",
"ABC" ), randomize = FALSE)
augm <- fold.design(des2)

```

The columns in the design data frame `augm` are factors, and the `optFederov` function needs a data frame of numerical candidates. Therefore, after creating the augmented design with the `FrF2` and `fold.design` functions from the R package `FrF2`, the columns of `augm` are converted into numerical coded variables and combined into the new data frame `augmn` as shown below.

```

A <- (as.numeric( augm$A ) - 1.5 ) / .5
B <- (as.numeric( augm$B ) - 1.5 ) / .5
C <- (as.numeric( augm$C ) - 1.5 ) / .5
D <- (as.numeric( augm$D ) - 1.5 ) / .5
E <- (as.numeric( augm$E ) - 1.5 ) / .5
F <- (as.numeric( augm$F ) - 1.5 ) / .5
G <- (as.numeric( augm$G ) - 1.5 ) / .5
Block <- augm$fold
augmn <- data.frame(A, B ,C, D, E, F, G, Block)

```

Next the `gen.factorial` function in the R package `AlgDesign` is used to create a list of 128 candidate points forming a full factorial in the data frame `cand`. The columns in this data frame are numerical coded factors and do not need to be converted for `optFederov`.

```

library(AlgDesign)
cand <- gen.factorial( levels = 2, nVar = 7,
varNames = c("A", "B", "C", "D", "E", "F", "G"))

```

The data frame `all`, consisting of the augmented fractional factorial and the candidate points, is created. This data frame contains one factor, `Block`, with indicators for the source of the data (`original`, `mirror`, `cand`).

```

Block <- rep('cand', 128)
cand <- data.frame( A=cand$A, B=cand$B, C=cand$C,
D=cand$D, E=cand$E, F=cand$F, G=cand$G, Block)
all <- rbind( augmn, cand)

```

Finally, the `optFederov` function is called to select eight runs from the `cand` block to augment the 16 runs in the original and mirror blocks so that the result will contain a total of 24 runs in three blocks. The options `augment=TRUE, rows=fr` tells the `optFederov` function to keep the first 16 runs (indicated by the elements of the vector `fr`) in the design and select the next eight runs from the remaining rows in `all`. The option `criterion="D"` statement specifies a D-optimal design. This is the default for `optFederov` if this option is omitted. To generate an A-optimal design change the option to `criterion="A"`.

```
Fr <- 1:16
optim <- optFederov( ~ A + B + F + I(A*D) + I(C*F),
  data = all, nTrials = 24, criterion = "D",
  nRepeats = 10, augment = TRUE, rows=fr)
```

The eight new runs in the third block are printed as shown below. After completing the experiments for this block, the model including main effects  $A$ ,  $B$ , and  $F$ , the  $AD$  and  $CF$  interactions, and a factor for the block differences could be fit to the data using a regression function such as `lm`.

```
newruns <- optim$design[ 17:24, ]
newruns
```

	A	B	C	D	E	F	G	Block
20	1	1	-1	-1	-1	-1	-1	cand
21	-1	-1	1	-1	-1	-1	-1	cand
32	1	1	1	1	-1	-1	-1	cand
51	-1	1	-1	-1	-1	1	-1	cand
54	1	-1	1	-1	-1	1	-1	cand
58	1	-1	-1	1	-1	1	-1	cand
63	-1	1	1	1	-1	1	-1	cand
89	-1	-1	-1	1	-1	-1	1	cand

## 6.6. Plackett-Burman (PB) Model Robust Screening Designs

Resolution III  $2^{k-p}$  fractional factorial designs are often used for screening experiments where the objective is to determine which factors (from a list assembled by brainstorming) are important enough to be studied in more detail in follow-up experiments. However, the

number of runs in an  $2^{k-p}$  fractional factorial design is always a power of 2, i.e., 8, 16, 32, etc., and these limited choices for run size can be restrictive in screening experiments. For example, to examine 8 factors requires at least 16 experiments, and to examine 16 factors requires 32 experiments when using a fractional factorial design.

Resolution III Plackett and Burman (Plackett & Burman, 1946) designs, or PB designs, are available in run sizes that are multiples of 4, i.e., 8, 12, 16, 20, etc. These designs were originally discovered by two British statisticians during World War II while studying the effect of a number of factors on the performance of antiaircraft proximity fuse prototypes. Like  $2^{k-p}$  fractional factorial designs, these designs have two levels for each factor; and for run sizes that are powers of 2, they are the same as a  $2^{k-p}$  design. For other run sizes, they retain the desirable orthogonality property of  $2^{k-p}$  designs, but they do not have generators or a defining relation. The designs for run sizes of 12, 20, and 24 can be created by cyclically rotating the factor levels for the first run. **Table 6-9** shows the factor levels for the first run in these designs.

**Table 6-9. Factor Levels for First Run of Plackett-Burman Design**

Run Size	Factor Levels
12	+ + - + + + - - - + -
20	+ + - - + + + + - + - + - - - + + -
24	+ + + + + - + - + + - - + + - - + - + - - -

**Table 6-10** shows how these factor levels are used to create a 12-run Plackett-Burman design in 11 factors. The factor levels for the first run are copied directly from the first line of Table 6.9. In the second run, the level for the first factor ( $A$ ) is the level of the eleventh factor ( $L$ ) in the first run, and the factor levels for factors  $B$  through  $L$  for the second run are the factor levels for factors  $A$  through  $K$  from the first run. This pattern is continued, cyclically rotating the factor levels for each run to create the factor levels for the next run until the last run. For run number 12, all the factors are set to their low (-) level.

To create a 12-run design in  $k$  factors, where  $k < 11$ , proceed exactly as above, but after completion use only  $k$  columns in the design to define factor setting. To create a 20- or 24-run Plackett-Burman design proceed

exactly as before but start with the row in **Table 6-9** with the correct run size. (Plackett & Burman, 1946)

**Table 6-10. 12-Run Plackett-Burman Design**

Run	A	B	C	D	E	F	G	H	J	K	L
1	+	+	-	+	+	+	-	-	-	+	-
2	-	+	+	-	+	+	+	-	-	-	+
3	+	-	+	+	-	+	+	+	-	-	-
4	-	+	-	+	+	-	+	+	+	-	-
5	-	-	+	+	+	+	-	+	+	+	-
6	-	-	-	+	-	+	+	-	+	+	+
7	+	-	-	-	+	-	+	+	-	+	+
8	+	+	-	-	-	+	-	+	+	-	+
9	+	+	+	-	-	-	+	-	+	+	-
10	-	+	+	+	-	-	-	+	-	+	+
11	+	-	+	+	+	-	-	-	+	-	+
12	-	-	-	-	-	-	-	-	-	-	-

Plackett-Burman designs can be created easily using the **FrF2** package. The example below illustrates the use of the **pb** function in that package to create the design shown in **Table 6-10**.

```
library(FrF2)
pb( nruns = 12, randomize = FALSE)
```

```
A B C D E F G H J K L
1 1 1 -1 1 1 1 -1 -1 -1 1 -1
2 -1 1 1 -1 1 1 1 -1 -1 -1 1
3 1 -1 1 1 -1 1 1 1 -1 -1 -1
4 -1 1 -1 1 1 -1 1 1 1 -1 -1
5 -1 -1 1 -1 1 1 -1 1 1 1 -1
6 -1 -1 -1 1 -1 1 1 -1 1 1 1
7 1 -1 -1 -1 1 -1 1 1 -1 1 1
8 1 1 -1 -1 -1 1 -1 1 1 -1 1
9 1 1 1 -1 -1 -1 1 -1 1 1 -1
10 -1 1 1 1 -1 -1 -1 1 -1 1 1
11 1 -1 1 1 1 -1 -1 -1 1 -1 1
12 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
class=design, type= pb
```

To create a 20- or 24-run Plackett-Burman design with the **pb** function in **FrF2**, simply substitute **nruns=20** or **nruns=24** for **nruns=12**. Equivalent Plackett-Burman designs can be created using a different

generator (or combination of factor settings for the first run) than that shown in Table 6.9. However, this does not mean that an arbitrary generator can be used. One set of valid alternate generators for the 12-, 20-, and 24-run Plackett-Burman design is shown in Lawson (Lawson J., 2010). The R package [BsMD](#) (Barrios, 2009) contains the data frame [PB12Des](#), which is the equivalent to a 12-run Plackett-Burman design that can be created with this alternate generator. (Plackett & Burman, 1946)

As an example of the use of a Plackett-Burman design, consider the data in **Table 6-11**, which comes from a study by Hunter et al. (Hunter, Hodsi, & Eager, 1982) of the fatigue life of weld-repaired castings. Seven factors were assigned to the first seven columns of the Plackett-Burman design matrix shown in **Table 6-11**, and the last four columns were not assigned to factors in defining the experiments. The purpose of the experiments was to identify the factors that affect the fatigue life of the weld repairs. The response,  $y$ , is the log of the lifetime of each weld repair. We will use this data to illustrate the analysis of data from a Plackett-Burman design. Analysis of this data has been presented in the literature by several others.

Although there is no defining relation in a Plackett-Burman design, it is a resolution III design and the unassigned columns in **Table 6-11** represent confounded strings of interactions. A first step in the analysis would be to calculate an effect for each of the 11 columns in **Table 6-11** and make a half-normal plot.

**Table 6-11. Design Matrix and Lifetime Data for Cast Fatigue Experiment**

Run	A	B	C	D	E	F	G	c8	c9	c10	c11	
1	+	-	+	+	+	-	-	-	+	-	+	4.733
2	-	+	+	+	-	-	-	+	-	+	+	4.625
3	+	+	+	-	-	-	+	-	+	+	-	5.899
4	+	+	-	-	-	+	-	+	+	-	+	7.000
5	+	-	-	-	+	-	+	+	-	+	+	5.752
6	-	-	-	+	-	+	+	-	+	+	+	5.682
7	-	-	+	-	+	+	-	+	+	+	-	6.607
8	-	+	-	+	+	-	+	+	+	-	-	5.818

9	+	-	+	+	-	+	+	+	-	-	-	5.917
10	-	+	+	-	+	+	+	-	-	-	+	5.863
11	+	+	-	+	+	+	-	+	-	+	-	6.058
12	-	-	-	-	-	-	-	-	-	-	-	4.809

The R code to retrieve the 12-run Plackett-Burman design data frame from the `BsMD` package, rename and reorder the columns to match **Table 6-11**, and then add the response data is shown below.

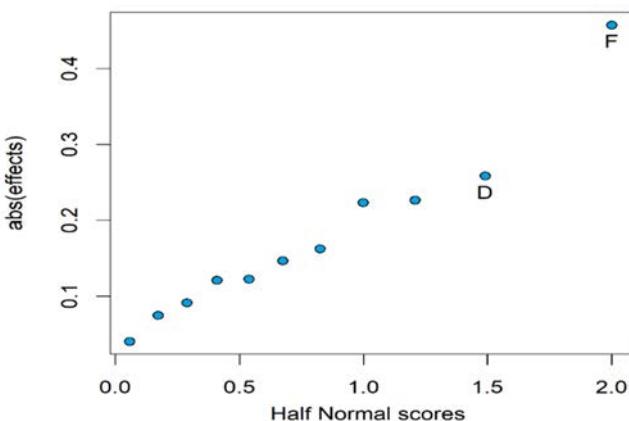
```
> data( PB12Des, package = "BsMD" )
> colnames(PB12Des) <- c("c11", "c10", "c9", "c8", "G",
  "F", "E", "D", "C", "B", "A")
> castf <- PB12Des[c(11,10,9,8,7,6,5,4,3,2,1)]
> y <- c(4.733, 4.625, 5.899, 7.0, 5.752, 5.682, 6.607,
  5.818, 5.917, 5.863, 6.058, 4.809)
> castf <- cbind( castf, y )
```

After organizing the data frame in the form of **Table 6-11**, the `lm` function was used to estimate the coefficients for each of the 11 columns and the `halfnorm` function in the package `daewr` was used (as shown below) to create the half-normal plot of effects. The plot is shown in **Figure 6-13**.

```
cfs <- coef(modpb)[2:12]
names<-names(cfs)
halfnorm(cfs, names, alpha = .35, refline=FALSE)
```

This plot shows that main effect  $F$  is clearly significant and that the next largest effect is main effect  $D$ . The authors of the original article concluded that the model including these two factors was the correct model, and they fit it to the data by regression, obtaining an  $R^2 = 0.5867$ . However, the same caution should be exercised that was illustrated in the water filtration example presented in **Section 6.5**.

Because this Plackett-Burman design has resolution III, two-factor interactions are confounded with main effects and there may be other plausible models for the data. A mirror image design could be added to the Plackett-Burman design in order to clear main effects of two-factor interactions; however, this is not necessary, due to the complex aliasing for this type design.



**Figure 6-13. Half-Normal Plot of Absolute Regression Coefficients from Cast Fatigue Experiment**

By complex aliasing we mean that each interaction is partially confounded with many main effects rather than being completely confounded with one main effect as it would be in a  $2^{k-p}_{III}$  design. Lin and Draper (Lin & Draper, 1992) and Wang III and Wu (Wang & Wu, 1995) showed that designs with complex aliasing have a hidden projection property. The hidden projection property allows some interactions to be estimated even though the design is resolution III. This can be illustrated graphically as shown in **Figure 6-14**. Color Map Comparison of Confounding between PB and FF Designs. The graph on the left side of the figure shows the color map of correlations computed from the design matrix for the Plackett-Burman design. The color map on the right side of the figure was computed from a resolution III  $2^{7-4}$  fractional factorial.

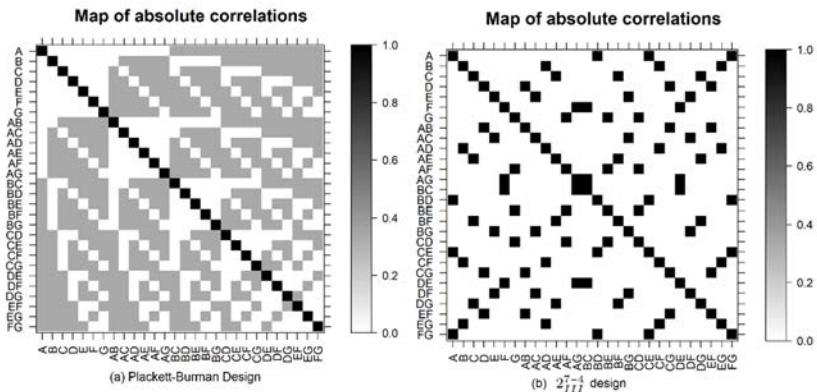
```

library(daewr)
castfr <- castf[, c(1:7)]
colormap(castfr, mod = 2)
library(FrF2)
design <- nFrF2(8, 7)
library(daewr)
colormap(design, mod = 2)

```

In these graphs we can see that each main effect is completely correlated or confounded with exactly three two-factor interactions for the resolution III fractional factorial design, but for the Plackett-Burman

design, each main effect is partially confounded (correlation coefficient  $\pm 0.333$ ) with five two-factor interactions. Since the correlations between main effects and interactions are not  $\pm 1$  for the Plackett-Burman design, some interactions can be included in the model as long as the total number of terms in the model is less than the number of runs in the design. The color maps were created with the `colormap` function from the `daewr` package that is illustrated in the R code for this chapter on web page for the book.



**Figure 6-14. Color Map Comparison of Confounding between PB and FF Designs**

For Plackett-Burman designs where only a subset of the factors appear to be important, Wang and Wu (1995) have shown that it is likely that a model can be fit to the data by regression, which includes the important main effects and two-factor interactions involving the important main effects. When interaction terms are included in the model, the design becomes non-orthogonal and effects must be estimated by regression, but Wang and Wu have shown that the  $D$ -efficiency remains quite high. Usually some type of regression subset selection procedure is used in order to identify the most appropriate model.

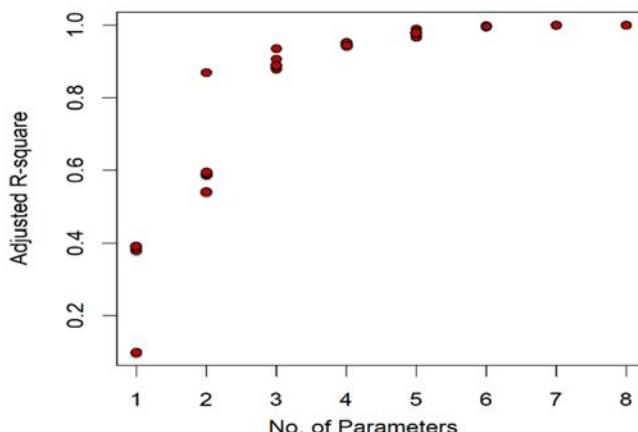
For the data in **Figure 6-11**, the R code on the next page uses the `regsubsets` function in the `leaps` package (Lumley, 2009) to perform an all-subsets regression including all main effects and two-factor interactions. The first statement in the code copies the main effect and response columns from the data frame `castf` into the reduced data frame `castfr`. The model statement `y ~ (. )^2` specifies that main effect and two-factor interactions will be included. The option `nvmax=4`

tells the `regsubsets` function to include at most four terms in the model, and the option `nbest = 4` tells the function to only keep the results for the best four models of each size. The first `plot` statement graphs the number of terms in the model vs. the adjusted  $R_a^2 = 1 - (n - 1)ss_E/(n - p) ss_{Total}$  (where  $n$  is the number of runs and  $p$  is the number of terms in the model) for the 16 best resulting models, and the second plot statement graphs the terms in the model vs. the model  $R^2$ .

```
castfr <- castf[, c(1:7, 12)]
modpbr <- regsubsets(y ~ (.)^2, data = castfr,
                      method = "exhaustive", nvmax = 8, nbest = 4)
plot(c(rep(1:8, each = 4)), rs$adjr2, xlab = "No. of
      Parameters", ylab = "Adjusted R-square")
plot(modpbr, scale = "r2")
```

The summary of the object `modpb` created by the `regsubsets` function is voluminous, but **Figure 6-15**, In **Figure 6-16**, we can see that the best model with two parameters includes main effect  $F$ , interaction  $FG$  and has an adjusted  $R^2=0.8686$ .

Table 6-12, and **Figure 6-16** condense the results. **Figure 6-15** shows that not much can be gained in terms of the adjusted  $R^2$  by including three or four parameters in the model. For that reason, the best model with two parameters best for this data.



**Figure 6-15.** Plot of Number of Model Terms by Adjusted  $R^2$

In **Figure 6-16**, we can see that the best model with two parameters

includes main effect  $F$ , interaction  $FG$  and has an adjusted  $R^2 = 0.8686$ .

**Table 6-12. Results of All-Subsets Regression**

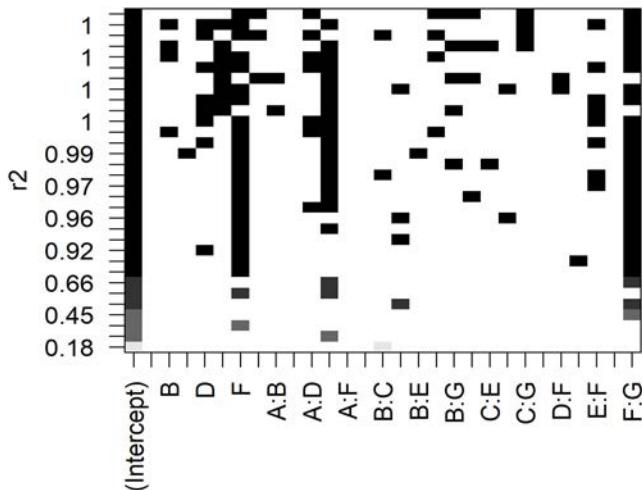
Number of Terms in Model	Adjusted R-Square	Variables in Model
1	0.3921	FG
1	0.3896	F
1	0.3814	AE
1	0.0993	BC
2	0.8686	F FG
2	0.5891	AE FG
2	0.5870	F AE
2	0.5403	BD FG
3	0.9348	F AE FG
3	0.9056	F BD FG
3	0.8886	D F FG
3	0.8785	F DG FG
4	0.9507	F AE EF FG
4	0.9465	F AE CD FG
4	0.9439	F AD AE FG
4	0.9438	F BD CF FG

```
library(leaps)
castfr <- castf[ , c(1:7, 12)]
modpbr <- regsubsets(y ~ .)^2, data = castfr,
method = "exhaustive", nvmax = 8, nbest = 4)
plot(c(rep(1:8, each = 4)), rs$adjr2,
xlab = "No. of Parameters",
ylab="Adjusted R-square")
plot(modpbr,scale = "r2")
```

**Figure 6-16** also shows that the terms appearing most frequently in all the models fit by the all-subsets regression were the intercept  $F$  and  $FG$ .

**Table 6-13** shows a summary of the data classified by the levels of factors  $F$  and  $G$ . There it can be seen that the consistently longest log fatigue life for the weld repaired castings occurs when factor  $F$  is at its high level and factor  $G$  is at its low level. In this example seven factors were under study. After just 12 experiments it was found that only two of the factors

( $F$  and  $G$ ) were important, and the conditions for maximum life (among the combinations studied) were discovered.



**Figure 6-16. Plot of Model Terms by  $R^2$**

**Table 6-13. Summary of Data from Cast Fatigue Experiment**

Factor F	Factor G	
	-	+
-	4.733	5.899
	4.625	5.752
	4.809	5.818
+	6.058	5.682
	7.000	5.917
	6.607	5.863

Resolution V fractional factorials allow estimation of all important main effects and all two-factor interactions involving the important main effects if three-factor and higher order interactions are assumed negligible. But these designs may require too many runs to be practical for screening. Resolution IV and resolution III fractional factorials usually require follow-up experiments (like the examples in **Section 6.5**) in order to estimate all important main effects and associated two factor interactions. However, using regression subset procedures, Plackett-Burman designs will allow fitting models involving the important main effects and a few interactions that need not be specified in advance. In this sense they can be called model robust because they are efficient for

fitting several possible models. In a literature survey of published experiments, Li et al. (2006) and Berquist et al. (2011) found that less than half the main effects and only 5–10% of two-factor interactions turn out to be important in screening designs. Therefore, by utilizing Plackett-Burman designs, the need for follow-up experiments to unconfound main effects from two-factor interactions or break confounding among strings of confounded two-factor interactions is reduced. In many cases, the Plackett-Burman design can be used as a one-step screening and optimization experiment as described by Lawson (2003).

Since the number of factors in a screening design can usually be accommodated by a subset of the columns in a Plackett-Burman design, Fairchild (2011) used an exhaustive search to determine which subset of columns in 12-and 20-run PB designs would result in the highest probability of a nonsingular design involving the main effects and any three two-factor interactions. His optimal subsets of Plackett-Burman designs can be produced by the `OptPB` function in the daewr package, as shown below for a 20-run design in 9 factors. In the output only the first few runs of the design are shown.

```
library(daewr)
OPB<-OptPB(20, 9, randomize = FALSE)
head(OPB)
```

	A	B	C	D	E	H	M	O	P
1	1	1	-1	-1	1	1	-1	-1	-1
2	-1	1	1	-1	-1	1	1	-1	-1
3	1	-1	1	1	-1	1	-1	-1	-1
4	1	1	-1	1	1	1	1	1	-1
5	-1	1	1	-1	1	-1	-1	-1	1
6	-1	-1	1	1	-1	-1	1	1	-1

Loepky et al. (Loepky, Sitter, & Tang, 2007) provide a more extensive catalog that gives the optimal subset of columns for estimating a few main effects and all possible two-factor interactions involving those main effects for 20-run designs.

Since 16-run Plackett-Burman designs are the same as a 16-run fractional factorial, they do not have the model robust feature of the 12-, 20-, and 24-run designs. For this reason, Jones and Montgomery (2010) have proposed alternate 16-run screening designs for 6, 7, and 8 factors.

These alternate screening designs were selected from projections of Hall's (1961) 16-run designs to produce a desirable correlation matrix involving all factors and all possible two-factor interactions involving those factors. The resulting designs have properties similar to the 12-, 20-, and 24-run Plackett-Burman designs and will allow fitting models that include main effects and a few interactions that are not specified in advance. These designs can be produced by the `Altscreen(nfac)` function in the `daewr` package, where `nfac` = 6, 7, or 8. Below is an example where only the first few runs of the resulting design are shown.

```
library(daewr)
ascr <- Altscreen(6, randomize= FALSE)
head(ascr)
```

	A	B	C	D	E	F
1	1	1	1	1	1	1
2	1	1	-1	-1	-1	-1
3	-1	-1	1	1	-1	-1
4	-1	-1	-1	-1	1	1
5	1	1	1	-1	1	-1
6	1	1	-1	1	-1	1

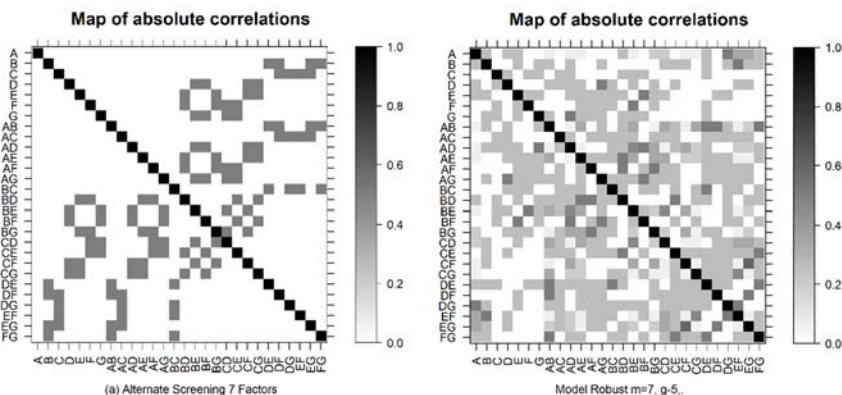
Li and Nachtsheim (2000) also developed 8-, 12- and 16-run model-robust screening designs using an exchange algorithm. Their designs have properties similar to Jones and Montgomery's (2010) 16-run designs and allow for estimating  $m$  main effects and up to  $g$  two-factor interactions, which need not be specified in advance. These designs can be produced by the `ModelRobust()` function in `daewr`. There are three 8-run designs, five 12-run designs, and four 16-run designs, which each accommodate a different number of main effects and interactions. They can be retrieved using the model name as the argument to the function as shown in the example below, that retrieves the 8-run design for  $m = 5$  main effects and up to  $g = 2$  interactions. Again, in this example, only the first few runs are shown. To retrieve a list of the names of all the designs in the catalog call the `ModelRobust` function with the design name left blank.

```
MR8 <- ModelRobust('MR8m5g2')
head(MR8)
```

	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

In Jones and Montgomery's (2010) designs, main effects are not confounded with each other and are partially confounded with two-factor interactions. While in Li and Nachtsheim's (2000) designs, main effects are partially confounded with two-factor interactions as well as with other main effects. This can be visualized for the 16 run designs for 7 factors shown in **Figure 6-17**.

Since the designs produced by the `Altscreen` and `ModelRobust` functions have complex aliasing, like the Plackett-Burman designs, they are best analyzed with a regression subset procedure as described above. Some authors such as Lin (1999) have suggested the use of forward stepwise regression to identify an appropriate model for data arising from a design with complex aliasing.



**Figure 6-17. Color Map Comparison of Confounding between Alternate Screening and Model Robust Designs**

To illustrate this procedure in R, consider again the data for the cast fatigue experiment. The step function could be used as shown below to identify a model.

```

NULL <- lm( y ~ 1, data = castfr )
up <- lm( y ~ (. )^2, data = castfr )
step( NULL, scope = list(lower = NULL, upper = up),
      direction = "forward", steps=4)

```

This code operates on the reduced data frame `castfr` containing only the main effect columns and the response from the Plackett-Burman design created in the code shown earlier. `null` defines a model with the minimum number of terms to be considered. In this case that is a model with only an intercept. `up` defines the set of all terms to be considered for the model. In this statement the model formula, `y ~ (. )^2`, creates all main effects and two-factor interactions. In the call of the `step` function, `null` is specified as the starting model and the option `steps=4` specifies the maximum number of forward steps. This should normally be set to 1/3 of the number of runs in the design because due to the effect sparsity principle there will rarely be more than that number of important effects. The `step` function uses the AIC or equivalently Mallows's  $C_p$  to decide whether additional terms should be added to the model. Running this code results in two forward steps. In the first step, main effect  $F$  is added to the model, and in the second and final step, main effect  $D$  is added to the model.

In the analysis of the cast fatigue experiment discussed earlier, two alternative models were plausible. Authors of the original article describing the experiment thought the model including main effects  $F$  and  $D$  was appropriate, but using all-subsets regression, the two-variable model ( $F$  and  $FG$ ) and the three-variable model ( $D$ ,  $F$ , and  $FG$ ) fit the data much better. Both of these models do not include any interactions that do not involve at least one of the main effects in the model (effect heredity). In this example, the three-term model contains one additional term that is not in the two-term model, and that term can be tested for significance, in a separate fit.

The downside of using a forward selection procedure is that it identifies one and only one model for the data. In some cases, this can be misleading and predictions from this model may not be accurate. This can be caused when there is more than one model that fits the data well. A forward selection only identifies one of these models and it may not

be the correct one. A better approach would be to use an all-subsets selection procedure. When choosing the model from the all-subsets procedure make sure the adjusted  $R_a^2$  is at or slightly less than the maximum, and that the model that obeys the effect heredity principle (i.e., it includes no interactions where neither parent main effect is in the model).

Hamada and Wu (1992) proposed a more involved iterative stagewise forward stepwise regression approach, guided by the principle of effect heredity, that overcomes some of the objections to using a straightforward regression. Jones and Nachtsheim (2011) proposed a simpler approach to forward regression that also forces effect heredity. Their approach, incorporated with the “Combine” option in the JMP forward stepwise regression, requires that any forward step that enters an interaction effect to the model also enters the main effects involved in that interaction. This avoids finding models that do not obey effect heredity (a situation that occurred less than 1% of the time in Li *et al.*’s (2006) study of 113 published factorial experiments).

In the `daewr` package, the functions `ihstep` and `fhstep` can perform a forward regression that enforces effect heredity. `ihstep` performs the first step of a hierarchical forward stepwise regression. If an interaction or quadratic term is entered first, the parent main effects are also entered into the model. The arguments for this function are a vector of response data and a data frame containing the design. It prints a summary of the model and returns a vector of the terms in the model. The `fhstep` function adds to a model already created. It performs a single step of a hierarchical forward stepwise regression by entering additional term(s) to a model already created by `ihstep` or `fhstep`. If an interaction or quadratic term is entered first, the parent main effects are also entered into the model. An example of calling these functions on the data from the cast fatigue experiment and a portion of the output are shown below.

```
des <- castfr[ ,c(1, 2, 3, 4, 5, 6,7 )]
y <- castfr[ ,8]
library(daewr)
trm <- ihstep( y, des , m = 0, c = 7)
```

```

Call:
lm.default(formula = y ~ .), data = d1)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.49700 -0.07758  0.02650  0.07867  0.44500 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 5.73025   0.07260  78.930 7.4e-13 ***
F            0.45758   0.07260   6.303 0.000232 ***
G            0.09158   0.07260   1.261 0.242669    
F.G         -0.45875   0.07260  -6.319 0.000228 ***  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2515 on 8 degrees of freedom
Multiple R-squared:  0.9104, Adjusted R-squared:  0.8767 
F-statistic: 27.08 on 3 and 8 DF,  p-value: 0.0001531

```

**fhstep( y, des, trm, m = 0, c = 7)**

```

Call:
stats::lm(formula = y ~ .), data = d2)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.33925 -0.08217 -0.02267  0.10534  0.36612 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 5.73025   0.06516  87.939 6.48e-12 ***
D           -0.11831   0.06911  -1.712 0.130661    
F            0.45758   0.06516   7.022 0.000207 ***  
G            0.09158   0.06516   1.405 0.202676    
F.G         -0.41931   0.06911  -6.067 0.000507 ***  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2257 on 7 degrees of freedom
Multiple R-squared:  0.9368, Adjusted R-squared:  0.9007 
F-statistic: 25.95 on 4 and 7 DF,  p-value: 0.0002713

```

[1] "D" "F" "G" "F:G"

In the first call to **ihstep**, the *FG* interaction correlated most with the

response, so the terms  $F$ ,  $G$ , and  $FG$  were included in the initial model to preserve effect heredity. In the call to `fhstep`, the additional argument `trm` is required. This is the vector of terms included in the initial model by `ihstep`. In the example, the function `fhstep` finds the  $AE$  interaction term correlates highest with the residuals from the initial model, and therefore the terms  $A$ ,  $E$ , and  $AE$  were added to the model. `fhstep` can be called repeatedly adding terms to a previous model at each call. This results in the model including  $D$ ,  $F$ ,  $G$ , and  $FG$ . All terms in this model are significant and the adjusted  $R^2$  is .9007, but the model does not obey effect heredity.

The use of the functions `ihstep` and `fhstep` are simple to use and have a better chance of finding a reasonable model than forward stepwise regression. However, in some cases, when analyzing data from a design with complex aliasing, an all-subsets regression may reveal several models with the same number of terms, similar  $R_a^2$ 's, each obeying the effect heredity principle, and yet have few terms in common. In this situation selecting the correct model with the data at hand may be impossible.

If this is the case, additional follow-up experiments may still be required to provide data that will allow accurate discrimination between the alternate models. The `optFederov` function in the package `AlgDesign` can be used as shown in **Section 6.5.3** to find an additional block of data that will make the augmented design  $D$ -optimal for the model composed of all the terms from the competing models. Then a model can be fit involving all terms, and the significance of each term can be tested.

In many situations, model robust screening designs with complex aliasing may reduce the total number of experiments required to identify the important main effects and two factor interactions. However, the data analysis with a regression subsetting procedure can be more involved than the simple analysis of  $2^{k-p}$  designs. Also, if a three-factor interaction like that shown in **Figure 6-11** is important, it would be very difficult to detect with regression subset selection. Therefore both traditional  $2^{k-p}$  designs and model robust designs with complex aliasing should have their place in an experimenter's toolbox.

## **6.7. Mixed Level Factorials and Orthogonal Arrays (OAs)**

In the preliminary stage of experimentation, where the objective may be to determine which factors are important from a long list of candidates, two-level fractional factorial designs or Plackett-Burman designs are often appropriate. If a factor has quantitative levels, the two levels are denoted symbolically by (-) and (+), where (-) represents the lowest level the experimenter would consider, and (+) represents the highest level the experimenter would consider. The high and low are usually spread out as far as feasibly possible in order to accentuate the signal or difference in response between the two levels. If a factor has qualitative levels, the (-) and (+) designations are arbitrary, but the two levels chosen normally would be two that the experimenter believes should result in the maximum difference in response.

Sometimes, however, two levels for each factor may not be adequate. In cases where the experimenter would like to consider nonlinear effects of quantitative factors or qualitative factors with more than two alternatives, two-level fractional designs will not be suitable. For example, Fannin et al. (1981) report an experiment investigating the effects of four three-level factors and two two-level factors upon the rate of bacterial degradation of phenol for the purpose of evaluating the fate of chemicals in aquatic ecosystems. A full factorial would require  $3^4 \times 2^2 = 324$  experiments; however, the study was completed using only a fraction of these runs by utilizing a mixed level fractional factorial design based on an orthogonal array. Taguchi (1991) describes an experiment to determine the factors that affect the durability of an auto clutch spring. The factors and levels are shown in the **Table 6-14**.

The levels of factors *A*, *C*, *F*, and *G* represented discrete alternatives that were of interest. Factors *B*, *D*, and *E* were continuous factors and three levels were included in order to determine whether there was a curvilinear relation between these factor levels and durability of the clutch springs. There was also interest in the interaction between factors *D* and *F* and the interaction between factors *D* and *G*. A full factorial would require  $3^5 \times 2^2 = 972$  experiments, but Taguchi was able to get valid data with a fractional design that included only 27 experiments.

**Table 6-14. Table of factors and levels for the auto clutch spring**

Factor	Description	Levels
A	Shape	3 alternatives
B	Hole ratio	2 possibilities
C	Coining	2 possibilities
D	Stress $\sigma_t$	90 65 40
E	Stress $\sigma_c$	200 170 140
F	Shop peening	3 alternatives
G	Outer perimeter planning	3 alternatives

An orthogonal array  $OA(N, s_1^{m_1}, \dots, s_\gamma^{m_\gamma}, 2)$  of strength 2 is an  $N \times m$  matrix,  $m = m_1 + \dots + m_\gamma$  where  $m_i$  columns have  $s_i (\geq 2)$  symbols (or levels) such that for any subset of two columns all possible combinations of the symbols occur equally often in the matrix. These designs are orthogonal in the main effects and are of resolution III. However, in some cases like the Plackett-Burman designs, they can be used to estimate a limited number of interactions as well. A necessary condition for the existence of an orthogonal array is that the number of runs  $N$  be divisible by the product of each possible pair of factor levels. For example, in the experiment described by Taguchi, with five three-level factors and two two-level factors,  $3^5 \times 2^2$ , no orthogonal array fractional factorial exists with  $N = 54$  runs because 54 is not divisible by  $2 \times 2 = 4$ . In addition, no orthogonal array exists with  $N = 24$  runs because 24 is not divisible by  $3 \times 3 = 9$ . However, an orthogonal array with  $N = 36$  happens to exist since 36 is divisible by 4, 6, and 9.

Entire books have been written on methods for obtaining particular orthogonal array designs; however, there are R functions that will do the work. Specifically, the function `show.oas` in the package `DoE.base` (Groemping, 2012) displays a list of orthogonal arrays (from a catalog taken mainly from Kuhfeld, (2009)) that would be appropriate for a given number of factors and levels. The call

```
show.oas(factors=list(nlevels=c(3,2),number=c(5,2)))
```

shown in the code on the next page requests a list of orthogonal arrays that would be appropriate for a design with five three-level factors and two two-level factors. The result shows 10 orthogonal arrays in the catalog that would work. The first orthogonal array (number 80 in the

catalog) has 36 runs (L36) with 11 two-level factors (2.11) and 12 three-level factors (3.12), and so forth.

```
library("DoE.base")
show.oas(factors=list(nlevels=c(3,2), number=c(5,2)))
```

```
no suitable resolution IV or more array found
122 orthogonal arrays found, the first 10 are listed
      name    nruns      lineage
80     L36.2.11.3.12   36  3~12;12~1;:(12~1!2~11;)
81     L36.2.10.3.8.6.1 36
85     L36.2.4.3.13   36  3~12;12~1;:(12~1!2~4;3~1;)
87     L36.2.3.3.9.6.1 36
89     L36.2.2.3.12.6.1 36  3~12;12~1;:(12~1!2~2;6~1;)
90     L36.2.2.3.5.6.2 36
374    L72.2.47.3.12   72  2~44;3~12;4~1;:(4~1!2~3;)
375    L72.2.46.3.8.6.1 72  2~43;3~8;4~1;6~1;:(4~1!2~3;)
380    L72.2.44.3.12.4.1 72
382    L72.2.43.3.8.4.1.6.1 72
```

The `oa.design` function in the `DoE package.base` finds an appropriate array in the catalog and assigns the factors to columns in the array. The code below illustrates the use of this function to find a fractional factorial array design that includes five three-level factors and two two-level factors. The option `columns="min3"` causes the factors to be assigned to columns in an orthogonal array in a way that aliasing of main effects with two-factor interactions is minimal.

```
# might take a few minutes to run
des<- oa.design(nlevels=c(3,3,3,3,3,2,2), nruns = 36,
columns = "min3", randomize = TRUE, seed = 104)
```

If the function `oa.design` does not find an orthogonal array that meets the specified requirements it will return a full factorial, replicated for enough residual degrees of freedom.

When the purpose of experimentation is to screen the important factors in a preliminary experiment, the number of runs required for an orthogonal array may be more than the experimenter is willing to run. For example, in the  $3^5 \times 2^2$  experiment, only 13 degrees of freedom are required to estimate all the main effects. If the experimenter, relying on the effect sparsity principle, believes that only one or two factors and

their interaction will be important (like the last example presented), then 36 runs may be more than necessary.

In this case, an orthogonal main effects plan (that provides uncorrelated estimates of all main effects) (Addelman & Kempthorne, 1979) or a near orthogonal array can be used. Although there are methods of constructing orthogonal main effect plans manually (Dey, 1985), an alternative is to create a near orthogonal array by selecting a D-optimal subset of a full factorial design or orthogonal array. The R code on the next page illustrates how this can be done using the `oa.design` function in the package `DoE.base`, and the `optFedorov` function in the package `AlgDesign`. The `oa.design` function is used to create a 36-run orthogonal array set of candidates, and the `optFedorov` function is used to select an 18-run D-optimal subset that is stored in the object `optim`. The function `oa.design` by default labels the factors in the design, A-G.

```
# might take a few minutes to run
des<- oa.design(nlevels = c(3,3,3,3,3,2,2), nruns = 36,
columns = "min3", randomize = TRUE, seed = 104)
# Create 18 run near-orthogonal array
library(DoE.base)
cand <- oa.design(nlevels = c(3,3,3,3,3,2,2),
nrns = 36, columns = "min3", seed=104)
library(AlgDesign)
optim<-optFedorov(~ A + B + C + D + E + F + G, cand,
nRepeats = 10, nTrials = 18, criterion = "D")
optim$design
```

```
$design
  A B C D E F G
1  3 3 1 1 2 1 1
2  2 1 3 2 1 2 2
3  2 2 3 2 3 2 1
5  2 2 1 1 1 2 2
6  1 1 2 1 2 2 1
8  2 2 2 2 2 1 2
9  2 3 2 3 3 2 1
12 1 3 2 1 3 2 2
14 1 3 1 2 1 1 2
16 3 1 2 3 1 1 1
17 2 1 1 1 3 1 2
18 1 2 1 3 3 1 1
23 3 1 1 2 3 2 2
```

```

25 3 2 3 1 3 1 2
29 1 3 3 3 2 1 2
30 1 3 3 2 1 2 1
33 3 2 2 2 1 1 2
35 3 2 1 3 2 2 2

```

Creating a design in this way guarantees low correlation between the factors in the design. If the experimenter wanted to estimate the interactions between the last three-level factor  $E$  and the two two-level factors  $F$  and  $G$ , in addition to the main effects, the function call would be changed as follows:

```

optim<-optFederov(~A+B+C+D+E+F+G+E:F+F:G, cand,
nRepeats = 10, nTrials = 18, criterion = "D")

```

Schoen (2010) compares optimal designs to orthogonal array designs for estimating main effects and all two-factor interactions.

As an example of the use of a near orthogonal array for estimating main effects, consider a conjoint analysis described by Wang et al. (2004). These types of studies are commonly done in market research to determine the relative importance of different product attributes with respect to a customer's preference for the product. The purpose of this particular study was to assess the market potential for using low-grade hardwood lumber (with knots and other character marks) to produce fine furniture since the low-grade hardwood is underutilized. The products studied were hardwood chairs, and the product attributes (or factors) and levels are shown in **Table 6-15**.

A questionnaire was developed where potential customers were asked to rate product alternatives, composed of combinations of the levels shown in **Table 6-15**, on a 7-point scale (1 = least preferred to 7 = most preferred). The surveys were conducted in the Vermont state fair where an exhibit booth was set up where respondents were able to see the design and character mark density of the chairs and ask questions about them. A label was attached to each chair indicating the price and guarantee policy. Potential respondents were enticed into filling out the survey by giving them a chance to enter a raffle to win the chair of their choice.

There are  $4^2 \times 3 \times 2 = 96$  possible combinations of levels of the factors

shown in **Table 6-15**, which would be too many to ask a respondent to rate. Twelve profiles or product alternatives were created from these levels using a mixed-level fractional factorial design.

**Table 6-15. Product Attributes and Levels for Conjoint Study**

Factor	Product Attribute	Levels
A	Design	1 = Rocking Chair
		2 = Arm Chair
		3 = Office Chair I
		4 = Office Chair II
B	Price	1 = \$350.00
		2 = \$425.00
		3 = \$500.00
		4 = \$575.00
C	Density of Marks	1 = Clear
		2 = Medium
		3 = Heavy
D	Guarantee Policy	1 = 1-Year
		2 = Unconditional

The following R code could be used to search for an orthogonal array for this example. The output would show the smallest orthogonal array for examining a  $4^2 \times 3 \times 2$  design was 48, which is still too many alternatives to ask a respondent to rate.

```
optim<- optFederov(~A+B+C+D+E+F+G+E:F+F:G,
nRepeats = 10, nTrials = 18, criterion = "D")
show.oas(factors=list(nlevels=c(4,3,2),number=c(2,1,1)))
```

```
4 resolution IV or more arrays found
      name nruns lineage
32    L192.2.3.3.1.4.2   192
34    L192.2.1.3.1.4.3   192
46    L384.2.4.3.1.4.2   384
49    L576.2.2.3.1.4.2.6.1   576
55    orthogonal arrays found,
the first 10 are listed
      name nruns                             lineage
131  L48.2.34.3.1.4.2   48 4~12;12~1;:(12~1!2~4;3~1;)(4~1!2~3;)
136  L48.2.31.3.1.4.3   48 4~12;12~1;:(12~1!2~4;3~1;)(4~1!2~3;)
141  L48.2.28.3.1.4.4   48 4~12;12~1;:(12~1!2~4;3~1;)(4~1!2~3;)
145  L48.2.25.3.1.4.5   48 4~12;12~1;:(12~1!2~4;3~1;)(4~1!2~3;)
150  L48.2.22.3.1.4.6   48 4~12;12~1;:(12~1!2~4;3~1;)(4~1!2~3;)
```

154 L48.2.19.3.1.4.7 48 4~12;12~1;:(12~1!2~4;3~1;)(4~1!2~3;) \\
 158 L48.2.16.3.1.4.8 48 4~12;12~1;:(12~1!2~4;3~1;)(4~1!2~3;) \\
 162 L48.2.13.3.1.4.9 48 4~12;12~1;:(12~1!2~4;3~1;)(4~1!2~3;) \\
 166 L48.2.10.3.1.4.10 48 4~12;12~1;:(12~1!2~4;3~1;)(4~1!2~3;) \\
 169 L48.2.7.3.1.4.11 48 4~12;12~1;:(12~1!2~4;3~1;)(4~1!2~3;)

The code below shows how a 12-run near orthogonal array could be created as a subset of an orthogonal array.

**Table 6-16** shows the 12-run design in actual factor levels that was used in this study. Respondents were asked to rate all 12 profiles. The average rating from 122 Vermont respondents is shown in the last column. The raw (summary) data for the surveys consisted of a count of the number of respondents

**Table 6-16. Mixed Level Fractional Factorial for Conjoint Study**

Profile	Design	Price (\$)	Density of Marks	Guarantee Policy	Average/ Rating
1	Rocking Chair	350	Clear	1-Year	5.32273
2	Rocking Chair	425	Heavy	1-Year	5.27871
3	Rocking Chair	575	Medium	Unconditional	5.35539
4	Arm Chair	425	Medium	Unconditional	4.73211
5	Arm Chair	500	Clear	Unconditional	4.75073
6	Arm Chair	575	Heavy	1-Year	4.24606
7	Office Chair I	350	Heavy	Unconditional	4.62892
8	Office Chair I	500	Medium	1-Year	3.94293
9	Office Chair I	575	Clear	1-Year	3.85872
10	Office Chair II	350	Medium	1-Year	4.39812
11	Office Chair II	425	Clear	Unconditional	4.71872
12	Office Chair II	500	Heavy	Unconditional	4.51137

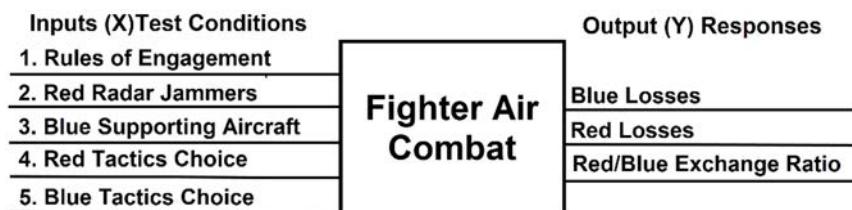
## 6.8. Example – Force-Level Encounter Assessment

Frequently, military testers encounter the problem of engaging in simulated combat operations against an “aggressor” adversary to determine methods of employing some new system or capability—tactics development. In the Air Force, force sizes range from one versus one to 50–75 aircraft encounters (“many vs. many”) in the periodic Red Flag exercises outside Las Vegas, Nevada. Valiant Shield, a June 2006 exercise, involved 22,000 personnel, 280 aircraft, and more than 30 ships (including three aircraft carriers and their strike groups) in the

Pacific Ocean and surrounding lands. (Johnson, Hutto, Simpson, & Montgomery, 2012)

Such large-scale force encounters offer appropriate scale to realistically exercise military systems against an unpredictable thinking adversary. In this sense, exercises are the best simulation of combat short of war. On the other hand, large-scale encounters are unwieldy, noisy, and offer fewer battles as experimental units than smaller force exercises. Experimental controls may restrict tactical free-play, thus hindering fighting force training. Nevertheless, force exercises are an important opportunity to test our military systems and tactics in an environment far too expensive for any single military test activity to afford on its own. This case illustrates effective experimentation in the midst of large force exercises. This case was adapted from McAllister's dissertation research (2003) concerning tactical employment of fighters. Air Force doctrine calls for rapidly establishing air supremacy—the unrestricted use of air and space—while denying it to the adversary. For the case study, eight friendly (traditionally "Blue") fighters with modern sensors, weapons, and communications contest the airspace with eight adversary ("Red") fighters. Engagements of this size are typical of air combat exercises such as Red Flag. (McAllister, 2003)

**Figure 6-18** illustrates some possible input and output conditions for the engagement. "SA" refers to the gold standard of air combat: situational awareness—accurately knowing where friends and enemies are. Lack of (or loss of) SA is frequently a terminal condition in air combat.

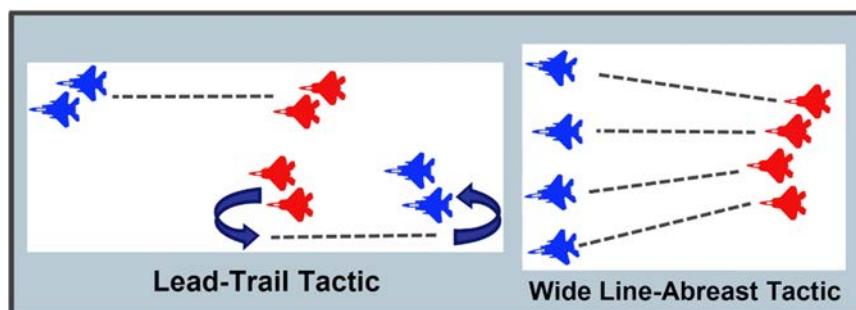


**Figure 6-18. Notional Blue–Red force engagement of eight fighters per side.**

The output measures count the losses on both sides and the exchange ratio. Combat exchange ratios have a long history and useful interpretations but are uninformative if the losses are zero on either side. McAllister (2003) considered three adjustments to the exchange

ratios to deal with these problems.

On the input side, some discussion is in order. Rules of engagement (ROE) specify the conditions under which a fighter is authorized to engage and destroy another aircraft. Rules of engagement may range from loose—allowing the destruction of any aircraft not positively identified to be friendly (a relatively quick process)—to tight ROE calling for closing the target for positive visual identification. Looser ROE allow sensors and missiles to be employed at maximum range (usually to Blue's advantage), whereas tighter ROE delay missile firings considerably. Radar jammers are employed to mask own-side aircraft from the enemy. This condition counts the number of dedicated standoff jamming aircraft available to the Red forces. Blue supporting assets refers to the number of airborne early warning, command and control, and intelligence aircraft available to the Blue side. Finally, the Red and Blue tactics options are inserted in the experiment in an attempt to answer whether one Blue tactic is universally superior to the other and whether Red's choice of tactics should influence Blue's tactical choices. As an illustration of such tactics, consider **Figure 6-19** and the two notional tactics developed for the Blue forces.



**Figure 6-19. Notional Blue tactical employment choices.**

A prime tenant of modern air warfare is to avoid closing (merging) with the adversary and engaging in what is popularly known as a dogfight. Such turning engagements nullify superior U.S. weapons and sensors, putting even relatively unsophisticated opponents in a position from which they may be able to destroy Blue aircraft. With the Lead-Trail tactic, one pair of fighters is always positioned to engage the adversary while the other turns away to maintain stand-off distance from the

adversary. With the Line-Abreast tactic, all four shooters are available for the initial salvo, maximizing the number of first-shot missiles in the air. The drawback to line abreast is that all four fighters turn away simultaneously, increasing the risk of a dogfight when Blue fighters turn back into the engagement.

### 6.8.1. Choice of Experimental Designs and Data Generation

As originally stated, the objective is to determine whether any tactical choices are superior for the Blue forces across an array of typical combat encounters. The experiment begins with a fractional factorial screening design with five factors, each at two levels (originally it was  $(2^3)(3^2)$ ): a fraction requiring 16 trials and yielding excellent information on the five main effects and 10 two-factor interactions.

The design table and constructive response data are provided in **Table 6-17** and **Table 6-19**. The ROE values represent the number of seconds typically required for a positive identification under the two rule sets; both Red and Blue supporting aircraft are represented by numeric counts, and the Red=Blue tactics choices are designated by the closest approach of the two adversary forces, with “0” representing a possible merge and resulting dogfight between Red and Blue fighters.

The simulated data shown in Table 6-19 were generated by an Excel Monte Carlo simulation created some years ago. The simulation has been used to produce sample data for classroom instruction, tactics development planning discussions, and a variety of technical papers (McAllister 2003 is an example). The Excel simulation emulates up to four missile exchanges between Red and Blue forces. It ends when the simulated missiles are exhausted or one force loses 50% of their aircraft. (Johnson, Hutto, Simpson, & Montgomery, 2012)

### 6.8.2. Full Factorial Design

If we were to analyze all five factors under homogeneous conditions, we would have to provide a full factorial design. In this case, it would be a  $(2^3)(3^2)$  or 72 runs. Yet, this same design can be easily converted into a 2k by converting the 3 level factors into two 2-level factors. For our experiment, the Red Jammers variable with three counts was divided

into low count level (count = 0 or 1) and a high-count level (count = 2) and Blue Support AC variable, with three counts, was divided into low count level (count = 2 or 4) and a high level (count = 8).

**Table 6-17. Design factors and levels**

Factor	Name	Units	Type	Design Value
A	ROE_t_ID	seconds	Numeric	10,60
B	Red_Jammers	count	Numeric	0,2
C	Blue_spt_AC	count	Numeric	2,8
D	Red_Tactic	nm	Numeric	0,5
E	Blue_Tactic	nm	Numeric	0,5

**Table 6-18. Simulated tactics—Development design and exchange ratios**

Std Units	ROE (A)	Red Jammers (B)	Blue Support Air (C)	Red Tactics (D)	Blue Tactics (E)	Red/Blue Kill Ratio (Kratio)
1	60	0	2	0	0	0.3
2	10	2	2	0	0	1.3
3	10	0	8	0	0	1.0
4	60	2	8	0	0	1.3
5	10	0	2	5	0	2.0
6	60	2	2	5	0	0.3
7	60	0	8	5	0	3.0
8	10	2	8	5	0	0.0
9	10	0	2	0	5	1.0
10	60	2	2	0	5	0.3
11	60	0	8	0	5	1.0
12	10	2	8	0	5	9.0
13	60	0	2	5	5	3.0
14	10	2	2	5	5	0.5
15	10	0	8	5	5	9.0
16	60	2	8	5	5	0.0

### 6.8.3. Load FrF2 package & Kill Ratio Data

```
library(FrF2)
k_ratio <-
```

```
read.csv("https://raw.githubusercontent.com/stricje1/Dat  
a/master/Craft_Data.csv")
```

#### 6.8.4. Load the Yield Factors

```
yield_factors <-  
read.csv("https://raw.githubusercontent.com/stricje1/Dat  
a/master/Craft_Data.csv")  
colnames(yield_factors) <- c("Factor", "A", "B", "C",  
"D", "E", "Y")  
yield_factors
```

	Factor	A	B	C	D	E	Y	
1		1	60	0	2	0	0	0.3
2		2	10	2	2	0	0	1.3
3		3	10	0	8	0	0	1.0
4		4	60	2	8	0	0	1.3
5		5	10	0	2	5	0	2.0
6		6	60	2	2	5	0	0.3
7		7	60	0	8	5	0	3.0
8		8	10	2	8	5	0	0.0
9		9	10	0	2	0	5	1.0
10		10	60	2	2	0	5	0.3
11		11	60	0	8	0	5	1.0
12		12	10	2	8	0	5	9.0
13		13	60	0	2	5	5	3.0
14		14	10	2	2	5	5	0.5
15		15	10	0	8	5	5	9.0
16		16	60	2	8	5	5	0.0

#### 6.8.5. Fractional Factorial Design 1

By using the R `FrF2` package we can determine the fractional factorial design. The highest resolution for this experimental design is Resolution III, with 5 factors and 16 runs.

```
runs <- 2^(6-2)  
nam2 <- c("A", "B", "C", "D", "E")  
frac_design <- FrF2(runs, factor.names = nam2,  
default.levels = c("0", "1"))  
summary(frac_design)
```

```
Call:  
FrF2(runs, factor.names = nam2, default.levels = c("0", "1"))
```

```

Experimental design of type FrF2
16 runs

Factor settings (scale ends):
A B C D E
1 0 0 0 0 0
2 1 1 1 1 1

Design generating information:
$legend
[1] A=A B=B C=C D=D E=E

$generators
[1] E=ABCD

Alias structure:
[[1]]
[1] no aliasing among main effects and 2fis

The design itself:
A B C D E
1 0 0 0 1 0
2 0 1 1 0 1
3 0 1 1 1 0
4 1 0 0 1 1
5 1 0 1 1 0
6 1 1 0 0 1
7 1 0 0 0 0
8 0 0 0 0 1
9 0 1 0 0 0
10 1 1 0 1 0
11 0 0 1 0 0
12 1 1 1 1 1
13 0 1 0 1 1
14 1 0 1 0 1
15 0 0 1 1 1
16 1 1 1 0 0
class=design, type= FrF2

```

Next, we place the data into a dataframe using the R data frame function.

```
frac1 <- data.frame(frac_design)
```

As observed, the aliasing structure of the fractional factorial design can be obtained and the generators of this design,  $A = A$ ,  $B = B$ ,  $C = C$ ,

$D = D$ , and  $E = E$ . In this particular experimental design, the generator of the fractional factorial design is  $E = ABCD$ , that is, the main effect corresponding to the factor `Blue_Tactic`, This specific structure also suggests that there are no aliasing among main effects and second order effects.

```
aliasprint(frac_design)
```

```
$legend  
[1] A=A B=B C=C D=D E=E  
  
[[2]]  
[1] no aliasing among main effects and 2fis
```

### 6.8.6. Fractional Factorial Design 2

By using the aliased interactions and multiplying by their main effects, the generator can also be determined as  $I = ABD = ACE = BCE$ . By adding this generator row to the matrix, it can be corroborated that they all equal the generator column:

```
frac2_design <- FrF2(runs, factor.names =  
c("A", "B", "C", "D", "E"), default.levels = c("-1", "1"))  
frac2 <- data.frame(frac2_design)  
frac2["I"] = 1
```

### 6.8.7. Randomization, Blocking and Replication

It is important to note the principles that guide experimental design: randomization, blocking and replication. In this experiment, the runs shown above must be completely randomized. That means, they should be randomly selected, assigned, and executed. Replication and blocking increases the precision of the results. Yet, in this experimental study we will not be using replication as the purpose is to have reduced number of experimental runs. If we decided to increase the number of runs for some reason, we might be better off generating a higher resolution design, such as a  $2^{6-2}$ .

### 6.8.8. Adding Response Variable to the Design

The next step in the design is to gather the data. As stated previously this has to be randomly selected and the experimental runs must be

done in a random order.

As a first step, given that in this case we have a database from where to obtain the data, the data is allocated as samples that match the different factor levels. For our fractional factorial design the order of experimental runs was developed as follows:

### 6.8.9. Randomization for First Design

```
rand_fd <- k_ratio[sample(nrow(frac1)),]  
rand_fd
```

	Std.Units	A	B	C	D	E	Kratio
10	10	60	2	2	0	5	0.3
16	16	60	2	8	5	5	0.0
13	13	60	0	2	5	5	3.0
12	12	10	2	8	0	5	9.0
1	1	60	0	2	0	0	0.3
5	5	10	0	2	5	0	2.0
4	4	60	2	8	0	0	1.3
9	9	10	0	2	0	5	1.0
7	7	60	0	8	5	0	3.0
8	8	10	2	8	5	0	0.0
2	2	10	2	2	0	0	1.3
6	6	60	2	2	5	0	0.3
14	14	10	2	2	5	5	0.5
15	15	10	0	8	5	5	9.0
3	3	10	0	8	0	0	1.0
11	11	60	0	8	0	5	1.0

### 6.8.10. Randomization for Second Design

```
rand_fd2 <- k_ratio[sample(nrow(frac2)),]  
rand_fd2
```

	Std.Units	A	B	C	D	E	Kratio
3	3	10	0	8	0	0	1.0
8	8	10	2	8	5	0	0.0
15	15	10	0	8	5	5	9.0
10	10	60	2	2	0	5	0.3
6	6	60	2	2	5	0	0.3
9	9	10	0	2	0	5	1.0
7	7	60	0	8	5	0	3.0
4	4	60	2	8	0	0	1.3
11	11	60	0	8	0	5	1.0
12	12	10	2	8	0	5	9.0

```

16      16 60 2 8 5 5    0.0
14      14 10 2 2 5 5    0.5
5       5 10 0 2 5 0    2.0
2       2 10 2 2 0 0    1.3
13      13 60 0 2 5 5    3.0
1       1 60 0 2 0 0    0.3

```

### 6.8.11. Model Comparison

We compared models two ways. First we compared the main effects and main effects with second order effects, Second, we compared the models using the two different designs (`frac1` and `frac2`). After comparing the outcome, we remodeled using the main effects  $A, B, C, D, E$  and the second order interactions  $A:C, A:E, B:D$ , and  $C:D$ . We then created two models, a linear model, and an ANOVA model, using the second design.

```

mod3 <- lm(Kratio ~ A+B+C+D+E+A:C+A:E+B:D+C:D,
             data = rand_fd)
summary(mod3)

```

Call:

```
lm.default(formula = Kratio ~ A + B + C + D + E + A:C + A:E +
           B:D + C:D, data = rand_fd)
```

Residuals:

Min	1Q	Median	3Q	Max
-1.3125	-0.7937	0.0125	0.7937	1.2875

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )	*
(Intercept)	-4.017500	1.579780	-2.543	0.04390	*
A	0.056333	0.030947	1.820	0.11857	
B	1.075000	0.500573	2.148	0.07536	.
C	0.765000	0.234789	3.258	0.01729	*
D	0.818333	0.309475	2.644	0.03832	*
E	0.918000	0.243589	3.769	0.00930	**
A:C	-0.010667	0.004719	-2.260	0.06453	.
A:E	-0.015800	0.005663	-2.790	0.03158	*
B:D	-0.620000	0.141583	-4.379	0.00467	**
C:D	-0.026667	0.047194	-0.565	0.59253	
---					
Signif. codes:	0 '***'	0.001 '**'	0.01 '*'	0.05 '.'	0.1 ' '

Residual standard error: 1.416 on 6 degrees of freedom

```
Multiple R-squared:  0.9021, Adjusted R-squared:  0.7552
F-statistic: 6.142 on 9 and 6 DF,  p-value: 0.01931
```

```
anova(mod3)
```

#### Analysis of Variance Table

Response: Kratio

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
A	1	13.323	13.323	6.6460	0.041883 *
B	1	3.610	3.610	1.8009	0.228159
C	1	15.210	15.210	7.5876	0.033091 *
D	1	0.422	0.422	0.2108	0.662329
E	1	13.323	13.323	6.6460	0.041883 *
A:C	1	10.240	10.240	5.1083	0.064535 .
A:E	1	15.603	15.603	7.7834	0.031583 *
B:D	1	38.440	38.440	19.1761	0.004673 **
C:D	1	0.640	0.640	0.3193	0.592530
Residuals	6	12.028	2.005		
	---				
Signif. codes:	0	'***'	0.001	'**'	0.01 '*' 0.05 '.' 0.1 ' ' 1

```
aov3 <- aov(Kratio ~ A+B+C+D+E+A:C+A:E+B:D+C:D, data =
rand_fd)
summary(aov3)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
A	1	13.32	13.32	6.646	0.04188 *
B	1	3.61	3.61	1.801	0.22816
C	1	15.21	15.21	7.588	0.03309 *
D	1	0.42	0.42	0.211	0.66233
E	1	13.32	13.32	6.646	0.04188 *
A:C	1	10.24	10.24	5.108	0.06453 .
A:E	1	15.60	15.60	7.783	0.03158 *
B:D	1	38.44	38.44	19.176	0.00467 **
C:D	1	0.64	0.64	0.319	0.59253
Residuals	6	12.03	2.00		
	---				
Signif. codes:	0	'***'	0.001	'**'	0.01 '*' 0.05 '.' 0.1 ' ' 1

```
anova(aov3)
```

#### Analysis of Variance Table

Response: Kratio

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
A	1	13.323	13.323	6.6460	0.041883 *

```

B      1  3.610  3.610  1.8009 0.228159
C      1 15.210 15.210  7.5876 0.033091 *
D      1  0.422  0.422  0.2108 0.662329
E      1 13.323 13.323  6.6460 0.041883 *
A:C    1 10.240 10.240  5.1083 0.064535 .
A:E    1 15.603 15.603  7.7834 0.031583 *
B:D    1 38.440 38.440 19.1761 0.004673 **
C:D    1  0.640  0.640  0.3193 0.592530
Residuals 6 12.028  2.005
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

### 6.8.12. Estimation

The model can be further tested, and the main and interaction effects can be estimating using a linear model:

```

fit1 <- lm(Kratio ~ C+D+E+A:E+B:D, data=rand_fd2)
summary(fit1)

```

```

Call:
lm.default(formula = Kratio ~ C + D + E + A:E + B:D, data =
rand_fd2)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.7375 -0.9062 -0.2250  0.7750  3.3125 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.637500  1.078280 -0.591  0.56750  
C            0.325000  0.149531  2.173  0.05485 .  
D            0.470000  0.219764  2.139  0.05817 .  
E            0.897000  0.252490  3.553  0.00525 ** 
E:A          -0.015200  0.005075 -2.995  0.01346 *  
D:B          -0.405000  0.126881 -3.192  0.00962 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

Residual standard error: 1.794 on 10 degrees of freedom
Multiple R-squared:  0.7379, Adjusted R-squared:  0.6068 
F-statistic:  5.63 on 5 and 10 DF,  p-value: 0.01004

```

### 6.8.13. Final Modeling & Analysis

Referring to the first mode results, the main effects concerning the `ROE_t_ID` and `Red_Jammers` were not statistically significant. On the

other hand, `Blue_Tactic`, `Bule_Spt_AC`, and `Red_Tactic` appears to be significant with the kill ratio. Analysis of variance performed indicated significant differences among the Blue Support Aircraft, Red Tactics Choice, and ROE.

With respect to interaction effects, there are only two interaction effects that seems to be statistically significant: (1) the is the effect of ROE and Blue Tactics Choice, and (2) Red Jammers and Red Tactics Choice..

If we reduce the model to the significant interactions Blue Tactics Choice, Blue Support Aircraft, and ROE the resulting model is as follows:

```
fit2 <- lm(k_ratio$Kratio ~ (k_ratio$E +
  k_ratio$C:k_ratio$E +
  k_ratio$C:k_ratio$E:k_ratio$A))
summary(fit2)
```

```
Call:
lm.default(formula = k_ratio$Kratio ~ (k_ratio$E +
k_ratio$C:k_ratio$E +
  k_ratio$C:k_ratio$E:k_ratio$A))

Residuals:
    Min      1Q  Median      3Q     Max 
-1.6735 -0.8515  0.1088  0.3559  2.7735 

Coefficients:
              Estimate Std. Error t value
Pr(>|t|)    
(Intercept)  1.15000  0.4549   2.528   0.0265 *
k_ratio$E    -0.22667  0.1989  -1.140   0.2766 
k_ratio$E:k_ratio$C  0.25463  0.0374   6.813 1.87e-05 ***
kratio$E:kratio$C:kratio$A -0.00389  0.0006  -6.240 4.32e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.287 on 12 degrees of freedom
Multiple R-squared:  0.8383, Adjusted R-squared:  0.7979 
F-statistic: 20.74 on 3 and 12 DF,  p-value: 4.864e-05
```

```
fit3 <- aov(k_ratio$Kratio ~ (k_ratio$E +
  k_ratio$C:k_ratio$E +
```

```
k_ratio$C:k_ratio$E:k_ratio$A))  
summary(fit3)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
k_ratio\$E	1	13.32	13.32	8.049	0.0150 *
k_ratio\$E:k_ratio\$C	1	25.20	25.20	15.228	0.0021 **
kratio\$E:kratio\$C:kratio\$A	1	64.45	64.45	38.937	4.32e-05
***					
Residuals	12	19.86	1.66		
---					
Signif. codes:	0	'***'	0.001	'**'	0.01 '*' 0.05 '.' 0.1 ' ' 1

```
anova(fit3)
```

Analysis of Variance Table

Response: k\_ratio\$Kratio

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
k_ratio\$E	1	13.323	13.323	8.049	0.01498 *
k_ratio\$E:k_ratio\$C	1	25.205	25.205	15.228	0.00210 **
kratio\$E:kratio\$C:kratio\$A	1	64.448	64.448	38.937	4.32e-05
***					
Residuals	12	19.862	1.655		
---					
Signif. codes:	0	'***'	0.001	'**'	0.01 '*' 0.05 '.' 0.1 ' ' 1

This second model seems to be more accurate although it can be even reduced some more as only two interaction effects appear to be significant

#### 6.8.14. Calculate Main Effects

The numerical results of these main effects are provided in this report for ROE, Red Jammers, Blue Supporting Aircraft, Red Tactics Choice, and Blue Tactics Choice. Main effects calculated were as follows:

```
me_roe <- mean(subset(k_ratio$Kratio, k_ratio$A ==  
"10")) -  
mean(subset(k_ratio$Kratio, k_ratio$A == "60"))  
me_red_jam <- mean(subset(k_ratio$Kratio, k_ratio$B  
=="0")) -  
mean(subset(k_ratio$Kratio, k_ratio$B == "2"))  
me_blue_spt <- mean(subset(k_ratio$Kratio,k_ratio$C==  
"2")) -
```

```

mean(subset(k_ratio$Kratio, k_ratio$C == "8"))
me_red_tac <- mean(subset(k_ratio$Kratio, k_ratio$D=="0"))
mean(subset(k_ratio$Kratio, k_ratio$D == "5"))
me_blue_tac <- mean(subset(k_ratio$Kratio, k_ratio$E=="0"))
mean(subset(k_ratio$Kratio, k_ratio$E == "5"))

```

### 6.8.15. Print Main Effects

```

[1] "ROE_t_ID      = 1.825"
[1] "Red_Jammers   = 0.95"
[1] "Blue_spt_AC   = -1.95"
[1] "Red_Tactic    = -0.325"
[1] "Blue_Tactic   = -1.825"

```

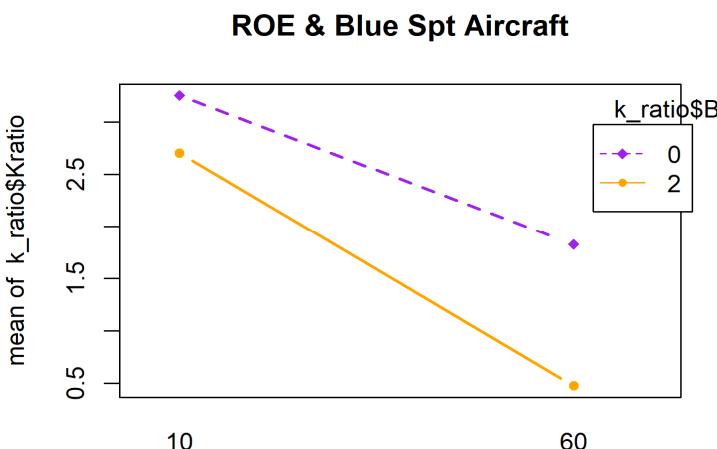
### 6.8.16. Interaction Plots

The following interaction plots represent the interaction of all second-order effects, significant or otherwise. The code to generate the first plot is (ROE and Red Radar Jammers):

```

interaction.plot(response = k_ratio$Kratio, k_ratio$A,
                  k_ratio$B, type = "b", pch = c(18,20), leg.bty="o",
                  lwd = 2, col=c("purple", "orange"), main = "ROE &
Blue Spt Aircraft")

```



*Figure 6-20. Interaction plot for ROE and Blue Supporting Aircraft*

## ROE & Red Jammers

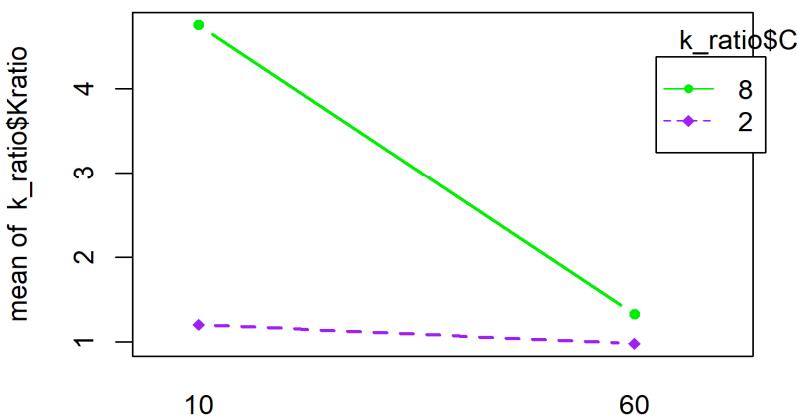


Figure 6-21. Interaction plot for ROE and Red Radar Jammers

## ROE & Red Tactics

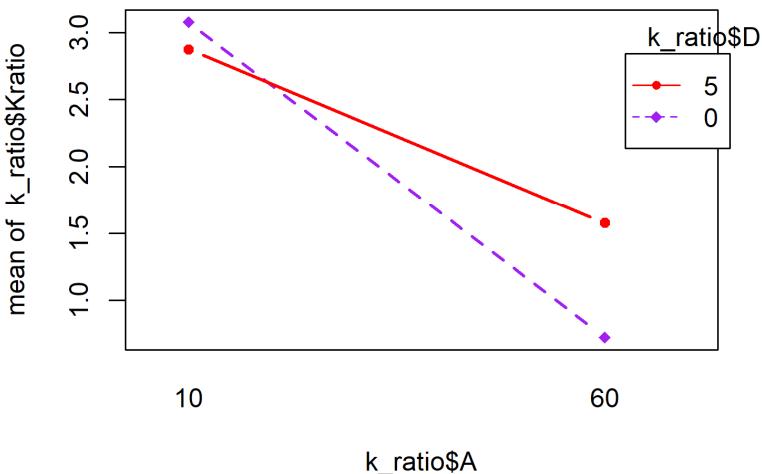


Figure 6-22. Interaction plot for ROE and Red Tactics

### ROE & Blue Tactics

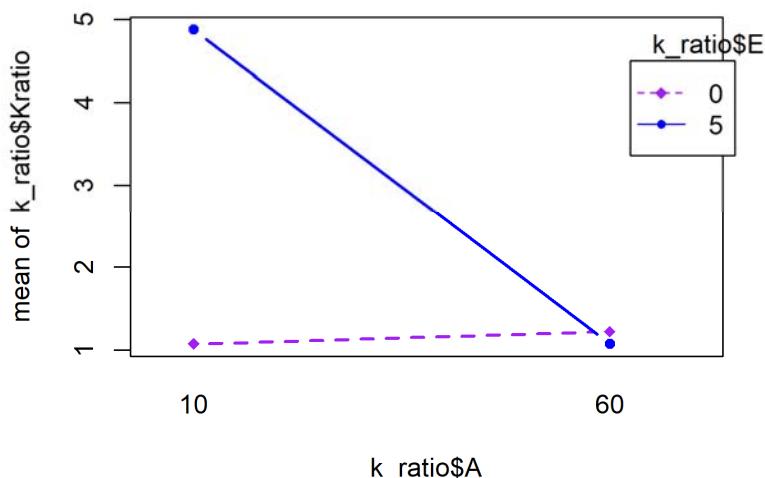


Figure 6-23. Interaction plot for ROE and Blue Tactics

### Red Jammers & Blue Spt Aircraft

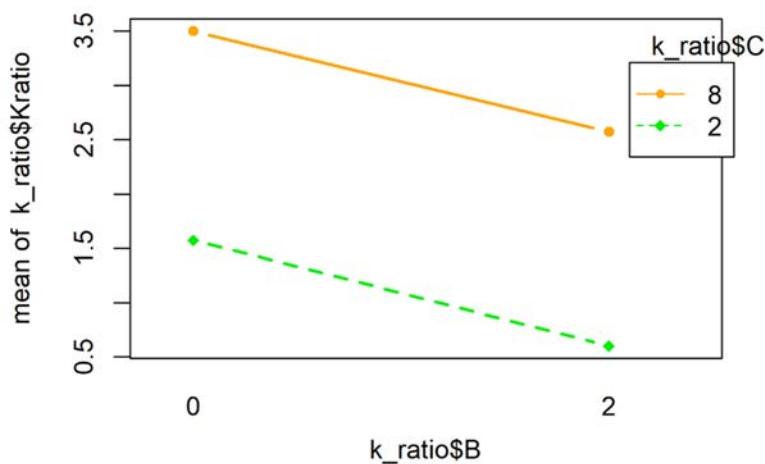


Figure 6-24. Interaction plot for Red Radar Jammers and Blue Supporting Aircraft

### Red Jammers & Red Tactics

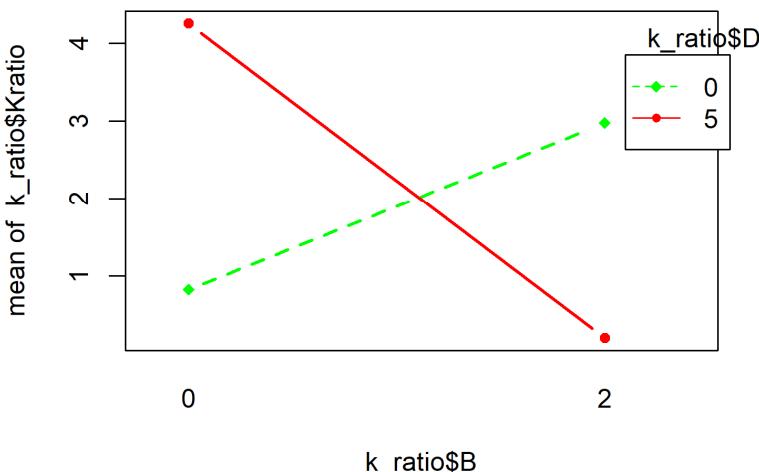


Figure 6-25. Interaction plot for Red Radar Jammers and Red Tactics

### Red Jammers & Blue Tactics

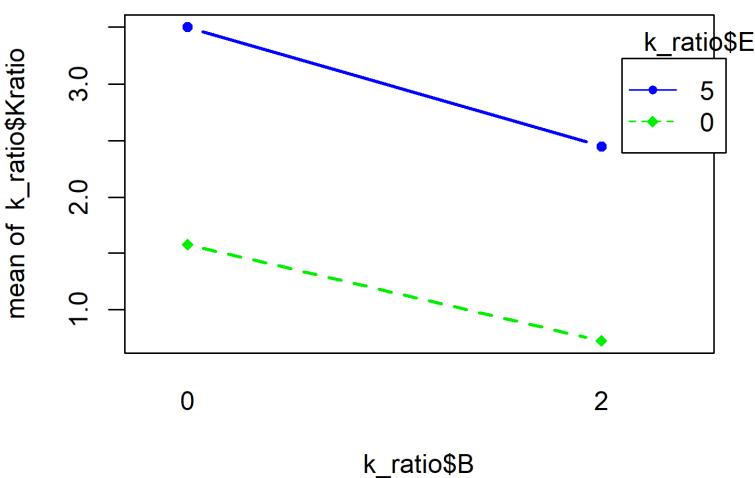


Figure 6-26. Interaction plot for Red Radar Jammers and Blue Tactics

### Blue Spt Aircraft & Red Tactics

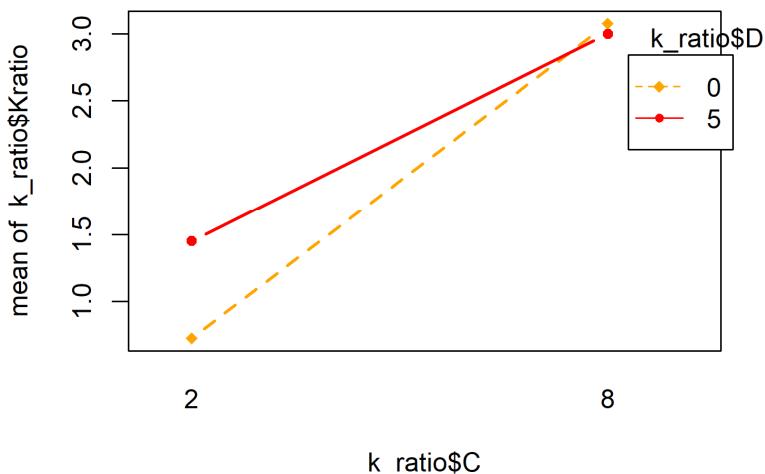
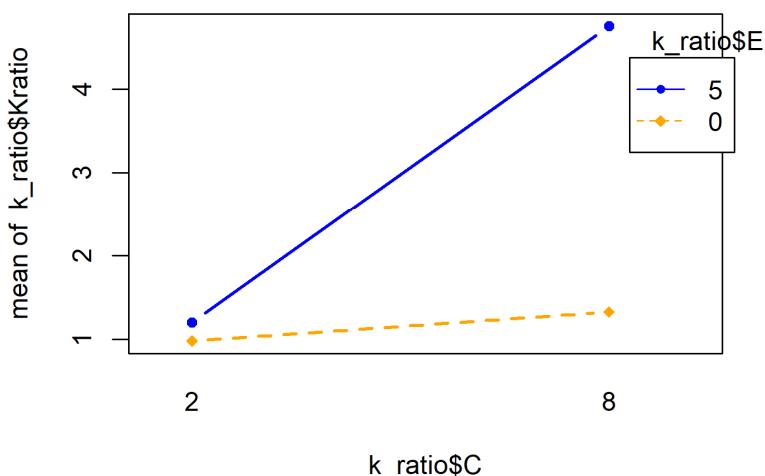
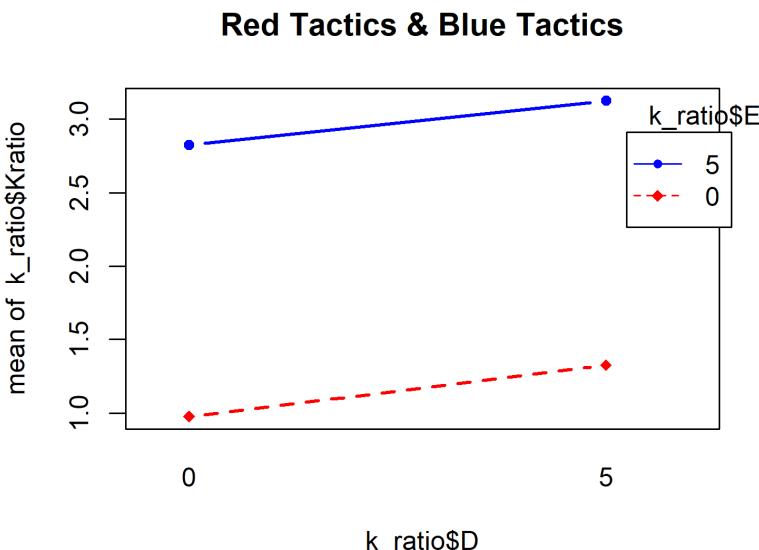


Figure 6-27. Interaction plot for Blue Supporting Aircraft and Red Tactics

### Blue Spt Aircraft & Blue Tactics



*Figure 6-28. Interaction plot for Blue Supporting Aircraft and Blue Tactics*



*Figure 6-29. Interaction plot for Red Tactics and Blue Tactics*

## 6.9. Example Experiment: Crosslinking Amine-Modified Silica Aerogels with Epoxies

In Chapter 3, we looked at this example for applying a  $2^2$  factorial design. However most of the factors involved have three levels, making it more relevant as a  $2^3$  factorial design. What follows is the original solution developed by NASA's Glenn Research Center.

Aerogels comprise a special class of low-density open-cell solid foams (typically, with porosity over 90%) which exhibit many unique properties such as exceptionally lightweight, high surface area, low thermal conductivity, extremely low dielectric constant, low sound wave transmission, high optical transparency in a wide range of wavelengths close to that of glass, and a very low refractive index (Fricke, 1988) (Emmerling, et al., 1995) (Ma, Roberts, Prévost, Jullien, & Scherer, 2000) (Woignier & Phalippou, 1987). These properties result from the microstructure of aerogels, which consists of a three-dimensional amorphous solid skeleton network with interconnected nanometer-sized pores in between. Silica aerogels are used for thermal and electrical insulation, especially in space applications, oxygen and

humidity sensors, aerosol particle collectors, space mirror protectors, catalyst supports, battery electrodes, etc. (Woignier & Phalippou, 1988). (Meador, et al., 2005)

### **6.9.1. The Laboratory Experiment**

NASA Glenn Research Center and the Ohio Aerospace Institute have recently demonstrated that templated polymerization of di-, tri-, and tetra-isocyanates on the surface of the nanoparticle building blocks of silica aerogels increases resulting conformal coatings increase the density of the native aerogels by a factor of 2-3 but the strength of the resulting materials may increase by more than two orders of magnitude.

The mesoporous surfaces of *tetramethoxysilane* (TMOS)-derived silica aerogels have been modified with amines by copolymerization of TMOS with *aminopropyltriethoxysilane* (APTES): I am very happy to have acronyms for these compounds. The amine sites have become anchors for crosslinking the nanoparticles of the skeletal backbone of the aerogel by attachment of *di*-, *tri*- and *tetra*-functional epoxies. Processing variables such as amount of APTES used to make the gels, the epoxy type and concentration used for crosslinking, as well as the crosslinking temperature and time were varied according to a multivariable DOE model. It was found that while elastic modulus follows a similar trend with density, maximum strength is attained neither at the maximum density nor at the highest concentration of -NH<sub>2</sub> groups, suggesting surface saturation effects. Aerogels crosslinked with the tri-functional epoxide always show improved strength compared with aerogels crosslinked with the other two epoxides under identical conditions. Solid <sup>13</sup>C NMR studies show residual unreacted epoxides, which condense with one another by heating crosslinked aerogels at 150 °C. (Medor, et al., 2005)

### **6.9.2. Experimental Design**

The effect of polymer accumulation on the particles has been quantified as a function of the processing parameters,  $a$ ,  $e$ ,  $c$ ,  $t$  and  $\theta$  as described above using a statistical experimental design approach, by following aerogel properties such as physical dimensions, density, surface area, porosity, strength, and flexibility. It was deemed reasonable to assume

that linear and non-linear effects of any variable on any physical property could be captured adequately by a full quadratic model of the form:

*physical property*

$$\begin{aligned}
 &= A + Ba + Ce + Dc + Et + F\theta + Ga^2 \\
 &+ Hc^2 + It^2 + J\theta^2 + Kae + Lac \\
 &+ Mat + Nat + Oec + Pe\theta + Qet \\
 &+ Rc\theta + Set + Tt\theta
 \end{aligned} \tag{Eq. 6-1}$$

where  $A$  through  $T$  are coefficients that would be derived empirically from experimental data. The model contains terms for first order effects of all five variables and second order terms for  $a$ ,  $c$ ,  $t$  and  $\theta$ , as well as all possible two-way interaction terms. (Owing to the discrete nature of variable epoxy type,  $e$ , there is no physical meaning to a second order term,  $e^2$ .) To evaluate first and second order terms for  $a$ ,  $c$ ,  $t$  and  $\theta$ , a minimum of three levels of each variable must be considered. **Table 6-19** provides a summary of the factors and levels used for the experiment.

**Table 6-19. Table of factors and levels**

Factor	Product Attribute	Levels
a	APTES percent	1 = 0%
		2 = 25%
		3 = 50%
c	Epoxy percent	1 = 15%
		2 = 45%
		3 = 75%
e	Epoxy type	1 = di
		2 = tri
		3 = tetra
t	Time	1 = 16 h
		2 = 44 h
		3 = 72 h
$\theta$	Temp	1 = 50 C
		2 = 72.5 C
		3 = 90 C

A full-factorial design to evaluate this model would contain at least 243 experiments ( $3^5$  experiments representing three levels each of five variables), not counting repeats. To minimize the number of

experiments, however, a *D-optimal* experimental design strategy was used. (Montgomery, 1997) To evaluate the desired model efficiently according to this type of nonclassical design, a set of experimental runs is computer-generated from the 243 candidate experiments. In total, only thirty-three aerogel samples were needed, including 5 repeats to assess model reliability and accuracy. These were prepared according to a planned scheme in random order, and were analyzed for their physical dimensions, density, surface area, porosity, strength, and flexibility.

**Table 6-20** summarizes the design runs and the experimental results.

**Table 6-20. Design of Experiments and Physical Characterization for the Resulting Epoxy Crosslinked Silica Aerogels**

Ru n	APTES percent	Epoxy type	Epoxy percent	Time , h	Temp , C	Densit y, g/cm <sup>3</sup>	Surfac e Area, c m <sup>2</sup> /g	Averag e Pore Diam. dA	Load Force , kg	Max Stress , 10 <sup>5</sup> N/m <sup>2</sup>	Modulu s, Mpa	Weight loss %
1	50	tetra	75	16	50	0.42	358	105	2.318	3.7	44.07	62
2	25	tri	15	44	72.5	0.42	443	116	5.112	7.78	53.5	66
3	0	tetra	15	72	95	0.44	662	125	2.497	3.46	33.33	58
4	50	di	16	95	0.3	462	114	0.99	1.33	13.3	60	60
5	25	tri	45	44	72.5	0.48	350	134	5.82	8.77	72.64	64
6	25	tri	45	44	72.5	0.47	355	151	6.942	10.12	71.95	63
7	25	tri	45	44	95	0.51	280	143 f	11.45	16.93	87.93	68
8	50	di	75	72	50	0.32	446	186	0.978	1.39	18.84	57
9	50	di	15	72	95	0.3	180	0.965	1.32	11.56	57	57
10	25	tri	44	72.5	0.49	328	146	9.439	14.16	82.17	63	63
11	25	tri	75	44	72.5	0.59	290	120	5.275	8.88	126.29	64
12	0	tri	45	44	72.5	0.36	267	140	2.225	2.97	18.28	54
13	0	tetra	75	72	50	0.24	670	83'	a	a	a	14
14	50	di	15	16	50	0.28	493	86	0.661	0.96	14.55	47
15	25	tri	45	44	72.5	0.49	309	141	5	8.04	74.67	63
16	25	tetra	45	44	72.5	0.42	450	125	3.196	4.95	46.5	55
17	0	tetra	15	16	50	0.2	821	112	a	a	a	10
18	25	tri	44	72.5	0.48	343	147	7.269	10.99	78.69	62	62
19	50	tetra	15	16	95	0.32	464	132	1.466	2	17.66	59
20	50	tri	45	44	72.5	0.4	278	91	5.75	7.47	34.98	70
21	0	di	75	72	95	0.24	444	100	0.341	0.45	3.24	32
22	25	tri	45	16	72.5	0.44	345	156	5.155	7.44	63.17	60
23	50	tetra	75	72	95	0.41	335	104	2.195	3.15	32.99	66
24	50	tetra	15	72	50	0.32	459	99	1.521	2.18	19.43	56
25	0	di	15	16	95	0.26	784	175	a	a	a	13
26	25	tri	45	44	72.5	0.47	312	151	7.775	11.28	71.46	63
27	25	tri	45	72	72.5	0.49	314	147	12.95	19.05	80.08	64
28	25	tri	45	44	50	0.47	316	119	6.043	9.39	70.41	60
29	0	di	15	72	50	0.21	856	62	a	a	a	10

30	25	di	45	44	72.5	0.33	546	145	1.855	2.79	24.35	45
31	0	tetra	75	16	95	0.29	423	115	0.44	0.67	6.94	28
32	28	tri	35	72	50	0.45	b	b	7.21	10.03	52	b
33	29	tri	35	72	50	0.48	b	b	7.39	11.57	50.7	b

a = Samples too fragile for testing

b = Samples not tested

### 6.9.3. Exploratory Data Analysis

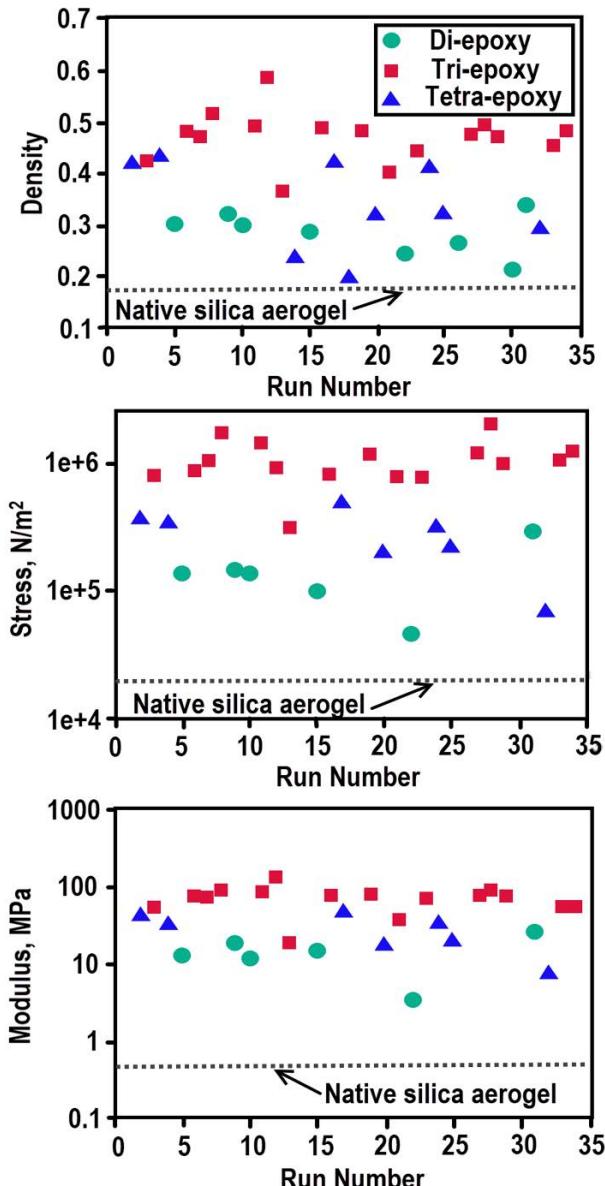
The density of the samples was determined from their physical dimensions and their weight. Surface area and pore diameters were determined by nitrogen adsorption *porosimetry*. Mechanical strength data (i.e., stress at break point and elastic modulus) were obtained by a three-point bend test method. **Figure 6-30** shows plots the density and mechanical strength data for all samples vs. the run order, distinguished by epoxy type. Since samples were prepared and tested in random order, such time-series plots should and do display a random distribution of values, indicating that no time-dependent errors (e.g., aging of monomers, temperature drift, etc.) are present. For comparison, dotted lines in the graphs show typical values for density, stress at breakpoint and modulus for native aerogels (i.e., non APTES).

Even without further analysis, these plots dramatically show that while the density of epoxy-crosslinked aerogels is at most 2-3 times higher than the density of native silica aerogels, both stress at breakpoint and modulus increase by as much as two orders of magnitude. In addition, these plots illustrate that the increases in all three properties is most striking for the tri-functional epoxy.

More systematically, values for selected measured responses (density, surface area, average pore diameter, stress at rupture, and modulus) were analyzed using linear least squares regression of eq I. All continuous, independent variables were orthogonalized (transformed to the -1 to 1 range) prior to modeling to minimize correlation among terms. Terms not statistically significant (< 90% confidence) were dropped from the model one at a time by the stepwise modeling technique. Summary statistics and significant terms in the models are shown in **Table 6-21**.

Without knowing the random number seed for the experiment, we ran our own models and yielded very close results, which are provided in

**Table 6-22.** The R code that generated the analysis for **Table 6-22** can be downloaded from one of my GitHub repositories (see the Preface).



*Figure 6-30. Time series plots of runs from the experimental design compared to native silica aerogels made using TMOS (no APTES)*

## 6.9.4. Experimental Results

**Table 6-21. Significant terms and summary statistics for response surface models**

Response	Significant Terms	R <sup>2</sup>	Standard (RMS) error
Density	a, c, e, t, $\theta$ $a^2, c^2$ $a^*t, a^* \theta, a^*c, e^*c, e^*t, t^* \theta, a^*e$	0.999	0.0066 g/cm
BET Surface Area	a, c, e, t, $\theta$ $a^2, c^2$ $a^* \theta, a^*c, e^*c, e^* \theta, t^* \theta, a^*e$	0.988	26.54 m <sup>2</sup> /g
Average Pore Diameter	a, e, t, $\theta$ $a^2, t^2, c^2$ $a^*t, e^*c, e^* \theta, a^*e$	0.962	7.75 Å
Stress at rupture (log transformed)	a, c, e, t, $\theta$ $a^2, c^2$ $a^*t, a^* \theta, a^*c, e^*c, t^* \theta$	0.985	1.197 N/m <sup>2</sup>
Modulus (log transformed)	a, c, e, t, $\theta$ $a^2, c^2$ $a^*t, a^* \theta, a^*c, e^*c, t^* \theta, e^*t$	0.998	1.006 MPa

**Table 6-22. My R linear model results for comparison**

Response	Significant Terms	Multiple R-squared	Standard (RMS) Error
Density	a, c, e, t $a^2, c^2$ $a^*t, a^*c, e^*t, t^* \theta, a^*e$	0.994	0.0152 g/cm
Surface Area	a, c, e, $\theta$ $a^2, c^2$ $a^* \theta, e^*c, e^* \theta, t^* \theta, a^*e$	0.993	18.51 m <sup>2</sup> /g
Average Pore Diameter	a, e, t, $\theta$ $a^2, t^2, c^2$ $a^*t, e^*c, e^* \theta, a^*e$	0.942	13.94 Å
Max Stress (log transformed)	a, c, e, t, $\theta$ $a^2, c^2$ $a^*t, a^* \theta, a^*c, e^*c, t^* \theta$	0.982	0.318 N/m <sup>2</sup>
Modulus (log transformed)	a, c, e $a^2, c^2$ $a^*t, a^* \theta, a^*c$	0.998	0.102 MPa

For the remainder of the example, I used an online ShinyApp that I created called doer or design of experiments using R.

## **6.10. doer R ShinyApp**

This is the R code used to generate these results in doer. I copied this into an R script and rerun to reproduce these results. So, what is `doer` or DOE using R? In the online documentation, I wrote:

“The purpose of this guide is to provide step-by-step instructions on how to generate and evaluate test designs using the Shiny app, `doer` (Design of Experiments in R) based on the R package `skpr` (Morgan-Wall and Khouri 2021). While guides exist to show how to perform these actions using a combination of JMP and the Excel-based Test Design Suite, this method produces superior results and capabilities all within a single software platform. Shiny, R ,and R Studio are also integrated into the Air Force Datalab VAULT environment, which allows for cloud storage and the use of multi-core computing power.”

“By using the `doer` GUI to implement the R `skpr` package, we can more easily determine the correct number of replications for our test design, as well as the associated conservative power estimates. We also gain the ability to use Monte Carlo simulation methods for designs with non-normal response variables or blocked designs.”

### **6.10.1. Load skpr Package**

```
library(skpr)
```

### **6.10.2. The doer GUI**

When you run `doer`, from <https://stricje1.shinyapps.io/doer/>, the first view is the “Basic” tab in the sidebar, inputs for:

- Trails (default value is 12)
- Model (example:  $\sim X_1 + X_2 + X_3 + X_1:X_2 + X_1:X_2 + X_2:X_3 + X_1:X_2:X_3$  )
- Number of Factors (default is 1; the above example has 3).

For each Factor, there are inputs for:

- Changes (Easy or Hard)
- Type (Continuous, Categorical, or Discrete Numeric)
- Name ( $X_1, X_2, \dots$ )

- Low (Lower value) and High (Upper value)
- Breaks (number of breaks, so that -2 to 2 with 4 breaks is (-2,-1), (-1,0), (0,1),(1,2))

The screenshot shows the doer software interface. At the top, there are two buttons: "Generate Design" and "Evaluate Design". Below them are three tabs: "Basic" (selected), "Advanced" (highlighted in blue), and "Model". Under the "Basic" tab, there is a "Trials" input field set to 28, a "Model" input field containing the equation  $\sim X1 + X2 + X3 + X4 + X5$ , and a "Number of Factors" input field set to 5. A large green box labeled "Factor 1" contains settings for "Changes" (set to "Easy") and "Type" (set to "Categorical"). It also has input fields for "Name" (set to "X1") and "Levels (separate with commas)" (set to "0%, 25%, 50%").

*Figure 6-31. The doer Basic tab view*

### 6.10.3. The Advanced Tab

Here is where you select optimality. We have selected *D*-optimal for this design. There are other optimality options in the dropdown menu, including *A*-optimal. This is also where you set the random number seed to ensure the design is repeatable. Other options include selecting Detailed Output and Advanced Design Diagnostics.

**Optimality**

D

**Repeats**

20

**Variance Ratio**

1

Set Random Number Generator Seed

Parallel Search

Include Blocking Columns in Run Matrix

Detailed Output

Advanced Design Diagnostics

**Color**

Plasma

*Figure 6-32. The doer Advanced tab view*

#### 6.10.4. doer Model Tab

In the Model tab, you can select the type of Model, the Model Family for the Generalized Linear Model (GLM), the alpha level, and others.

**Model Type**

Linear Model

Generalized Linear Model

Survival Model

**Alpha**

0 0.05 1

**SNR**

2

**Number of Simulations**

1000

**GLM Family**

gaussian

Parallel Evaluation

Colorblind Palette

*Figure 6-33. The doer Model tab view*

When we have selected all of the design options that we require, we

press the large Generate Design button at the upper left corner.

Design											
28-run D-optimal design											
Run	X1	X2	X3	X4	X5	Run	X1	X2	X3	X4	X5
1	50%	15%	tetra	44h	90C	15	0%	75%	tetra	44h	72.5C
2	25%	15%	di	16h	72.5C	16	50%	15%	di	16h	72.5C
3	50%	45%	tri	16h	90C	17	0%	75%	tetra	72h	50C
4	25%	45%	tetra	44h	72.5C	18	0%	45%	tri	44h	72.5C
5	25%	15%	tetra	44h	90C	19	50%	45%	di	44h	50C
6	0%	45%	tetra	16h	72.5C	20	0%	75%	di	16h	90C
7	0%	15%	di	44h	50C	21	50%	45%	tetra	72h	50C
8	25%	15%	tetra	72h	50C	22	0%	15%	tri	72h	90C
9	0%	15%	tri	16h	50C	23	25%	75%	tri	72h	72.5C
10	25%	75%	di	44h	50C	24	0%	45%	di	72h	90C
11	50%	75%	tetra	16h	90C	25	50%	75%	tri	44h	50C
12	25%	75%	tetra	16h	50C	26	25%	45%	di	72h	90C
13	25%	45%	tri	16h	50C	27	50%	15%	tri	72h	72.5C
14	25%	75%	tri	44h	90C	28	50%	75%	di	72h	72.5C

*Figure 6-34. The generated design*

Rather than running design evaluation in doer, we used the Generate Code tab, copied the R code there, and paste it into a script in R Studio. The script follows.

### 6.10.5. Generating candidate set:

```
candidateset = expand.grid(X1 = c("0%","25%","50%"),
                           X2 = c("15%","45%","75%"),
                           X3 = c("di","tri","tetra"),
                           X4 = c("16h","44h","72h"),
                           X5 = c("50C","72.5C","90C"))
```

### 6.10.6. Generating design:

```
design = gen_design(candidateset = candidateset,
                     model = ~X1 + X2 + X3 + X4 + X5 ,
                     trials = 28)
```

### 6.10.7. Evaluating (Monte Carlo) Design:

```
eval_design_mc(design = design,
               model = ~X1 + X2 + X3 + X4 + X5 ,
               alpha = 0.05,
               effectsize = 2,
               detailedoutput = TRUE)
```

	parameter		type	power	anticoef	alpha	glmfamily
trials	nsim						
1	(Intercept)	effect.power.mc	0.999	NA	0.05	gaussian	
2	X1	effect.power.mc	0.932	NA	0.05	gaussian	
3	X2	effect.power.mc	0.937	NA	0.05	gaussian	
4	X3	effect.power.mc	0.944	NA	0.05	gaussian	
5	X4	effect.power.mc	0.941	NA	0.05	gaussian	
6	X5	effect.power.mc	0.944	NA	0.05	gaussian	
7	(Intercept)	parameter.power.mc	0.999	1	0.05	gaussian	
	X11	parameter.power.mc	0.919	1	0.05	gaussian	
9	X12	parameter.power.mc	0.922	-1	0.05	gaussian	
10	X21	parameter.power.mc	0.935	1	0.05	gaussian	
11	X22	parameter.power.mc	0.935	-1	0.05	gaussian	
12	X31	parameter.power.mc	0.927	1	0.05	gaussian	
13	X32	parameter.power.mc	0.947	-1	0.05	gaussian	
14	X41	parameter.power.mc	0.933	1	0.05	gaussian	
15	X42	parameter.power.mc	0.933	-1	0.05	gaussian	
16	X51	parameter.power.mc	0.933	1	0.05	gaussian	
17	X52	parameter.power.mc	0.941	-1	0.05	gaussian	
	blocking	error_adjusted_alpha	power_lcb	power_ucb			
1	FALSE		0.05	0.9944411	0.9999747		
2	FALSE		0.05	0.9145869	0.9468119		
3	FALSE		0.05	0.9201120	0.9512531		

```

4 FALSE 0.05 0.9278923 0.9574245
5 FALSE 0.05 0.9245509 0.9547866
6 FALSE 0.05 0.9278923 0.9574245
7 FALSE 0.05 0.9944411 0.9999747
8 FALSE 0.05 0.9003283 0.9351565
9 FALSE 0.05 0.9036064 0.9378588
10 FALSE 0.05 0.9178989 0.9494797
11 FALSE 0.05 0.9178989 0.9494797
12 FALSE 0.05 0.9090858 0.9423464
13 FALSE 0.05 0.9312448 0.9600507
14 FALSE 0.05 0.9156900 0.9477022
15 FALSE 0.05 0.9156900 0.9477022
16 FALSE 0.05 0.9156900 0.9477022
17 FALSE 0.05 0.9245509 0.9547866
=====
Info=====
* Alpha = 0.05 * Trials = 28 * Blocked = FALSE
* Evaluating Model = ~X1 + X2 + X3 + X4 + X5
* Anticipated Coeffs = c(1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1)
* Contrasts = `contr.sum`
* Parameter Analysis Method = `lm(...)`
* Effect Analysis Method = `car::Anova(fit, type = "III")`
* MC Power CI Confidence = 95%

```

This shows that the power of our design is 95%. Next, we take the design and use it to analyze the experimental data in **Table 6-20..**

### 6.10.8. Analysis

How to analyze this experiment when the data have been collected: 1. (to run, remove one # from this section) 2. First, assign the results to a column in the data frame. Each 3. entry in the vector corresponds to the result from that run in the design.

Next, we construct the design matrix by adding the dependent variable or response, Density.

```
design$Y = density$Density
```

Now analyze the generalized linear model with several linear models, using `lm()`.

### 6.10.9. Linear model (lm) with main effects only

```

design.lm1 <- lm(formula = Y ~ . , data = design,
  contrasts = list(X1 = contr.sum, X2 = contr.sum, X3 =
contr.sum, X4 = contr.sum, X5 = contr.sum))
summary(design.lm1)

```

Call:

```

lm.default(formula = Y ~ . , data = design,
  contrasts = list(X1 = contr.sum, X2 = contr.sum,
X3 = contr.sum, X4 = contr.sum, X5 = contr.sum))

```

Residuals:

	Min	1Q	Median	3Q	Max
	-0.11240	-0.06281	-0.01269	0.06300	0.13708

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	0.410088	0.017375	23.602	1.97e-14 ***
X11	-0.007865	0.024738	-0.318	0.754
X12	-0.011199	0.024738	-0.453	0.656
X21	0.018801	0.024738	0.760	0.458
X22	0.013246	0.024738	0.535	0.599
X31	-0.005643	0.024738	-0.228	0.822
X32	0.013246	0.024738	0.535	0.599
X41	-0.004532	0.024738	-0.183	0.857
X42	0.031023	0.024738	1.254	0.227
X51	0.009912	0.024738	0.401	0.694
X52	0.025468	0.024738	1.030	0.318
---				
Signif. codes:	0 ****	0.001 **	0.01 *'	0.05 .'
	0.1	'	'	1

Residual standard error: 0.0915 on 17 degrees of freedom  
Multiple R-squared: 0.3002, Adjusted R-squared: -0.1114  
F-statistic: 0.7293 on 10 and 17 DF, p-value: 0.6887

### 6.10.10. lm with main effects and 2nd order effects

```

design.lm2 <- lm(formula = Y ~ .^2 , data = design,
  contrasts = list(X1 = contr.sum, X2 = contr.sum,
X3 = contr.sum, X4 = contr.sum, X5 = contr.sum))
summary(design.lm2)

```

Call:

```

lm.default(formula = Y ~ .^2, data = design,
  contrasts = list(X1 = contr.sum, X2 = contr.sum,
X3 = contr.sum, X4 = contr.sum, X5 = contr.sum))

```

Residuals:

ALL 28 residuals are 0: no residual degrees of freedom!

Coefficients: (23 not defined because of singularities)

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	0.4071605	NaN	NaN	NaN
X11	-0.0049383	NaN	NaN	NaN
X12	-0.0082716	NaN	NaN	NaN
X21	-0.0609259	NaN	NaN	NaN
X22	0.0385185	NaN	NaN	NaN
X31	-0.0027160	NaN	NaN	NaN
X32	0.0161728	NaN	NaN	NaN
X41	0.0842593	NaN	NaN	NaN
X42	-0.0574074	NaN	NaN	NaN
X51	-0.0225926	NaN	NaN	NaN
X52	0.0364815	NaN	NaN	NaN
X11:X21	-0.1457407	NaN	NaN	NaN
X12:X21	0.1140741	NaN	NaN	NaN
X11:X22	0.1159259	NaN	NaN	NaN
X12:X22	-0.1370370	NaN	NaN	NaN
X11:X31	-0.0228395	NaN	NaN	NaN
X12:X31	0.0404938	NaN	NaN	NaN
X11:X32	-0.0150617	NaN	NaN	NaN
X12:X32	-0.0283951	NaN	NaN	NaN
X11:X41	0.1635185	NaN	NaN	NaN
X12:X41	-0.0155556	NaN	NaN	NaN
X11:X42	-0.0581481	NaN	NaN	NaN
X12:X42	NA	NA	NA	NA
X11:X51	0.0003704	NaN	NaN	NaN
X12:X51	-0.0370370	NaN	NaN	NaN
X11:X52	-0.0387037	NaN	NaN	NaN
X12:X52	NA	NA	NA	NA
X21:X31	0.0344444	NaN	NaN	NaN
X22:X31	0.0466667	NaN	NaN	NaN
X21:X32	0.0011111	NaN	NaN	NaN
X22:X32	NA	NA	NA	NA
X21:X41	NA	NA	NA	NA
X22:X41	NA	NA	NA	NA
X21:X42	NA	NA	NA	NA
X22:X42	NA	NA	NA	NA
X21:X51	NA	NA	NA	NA
X22:X51	NA	NA	NA	NA
X21:X52	NA	NA	NA	NA
X22:X52	NA	NA	NA	NA
X31:X41	NA	NA	NA	NA
X32:X41	NA	NA	NA	NA

X31:X42	NA	NA	NA	NA
X32:X42	NA	NA	NA	NA
X31:X51	NA	NA	NA	NA
X32:X51	NA	NA	NA	NA
X31:X52	NA	NA	NA	NA
X32:X52	NA	NA	NA	NA
X41:X51	NA	NA	NA	NA
X42:X51	NA	NA	NA	NA
X41:X52	NA	NA	NA	NA
X42:X52	NA	NA	NA	NA

Residual standard error: NaN on 0 degrees of freedom  
 Multiple R-squared: 1, Adjusted R-squared: NaN  
 F-statistic: NaN on 27 and 0 DF, p-value: NA

The results indicate that the model is a perfect fit, hence no error results.  
 Is the model a perfect fit or is it overfit? To answer this let us look at some additional models with significant factors only.

### 6.10.11. ANOVA Model with Main Effect only

```
design.aov1 <- aov(Y~., data = design)
summary(design.aov1)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
X1	2	0.00224	0.001118	0.134	0.876
X2	2	0.01921	0.009603	1.147	0.341
X3	2	0.00342	0.001709	0.204	0.817
X4	2	0.01726	0.008632	1.031	0.378
X5	2	0.01893	0.009463	1.130	0.346
Residuals	17	0.14232	0.008372		

### 6.10.12. ANOVA Model with main effects and 2nd order effects

```
design.aov2 <- aov(Y~.^2, data = design)
summary(design.aov2)
```

	Df	Sum Sq	Mean Sq
X1	2	0.00224	0.001118
X2	2	0.01921	0.009603
X3	2	0.00342	0.001709
X4	2	0.01726	0.008632
X5	2	0.01893	0.009463
X1:X2	4	0.01792	0.004481
X1:X3	4	0.01946	0.004866
X1:X4	3	0.06042	0.020141

X1:X5	3 0.02285 0.007618
X2:X3	3 0.02166 0.007219

## 6.11. Review of Important Concepts

When more than one factor is under study, factorial designs are preferable to studying each factor in separate designs. Experimentation is more efficient, and interactions can be detected, which could be the most important information obtained from the experiments. However, if the list of factors to be studied is long, factorial designs may require too many experiments to be practical even when there are only two levels of each factor and no replicates. Fractional factorial designs (CRFF, PB, or OA) require only a fraction of the runs for a full factorial yet preserve many of the benefits such as orthogonality and ability to detect interactions. **Figure 6-35** illustrates when these designs should be used in relation to the designs presented in earlier chapters.

In order to preserve benefits of factorial designs, the runs in a fractional factorial must be carefully chosen. For half-fractions of  $2^k$  designs this is accomplished by choosing a full factorial in  $k - 1$  factors and assigning the coded factor levels of the  $k$ th factor to the product of the coded factor levels of all the other factors. The defining relation is used to determine the alias structure of the design that shows which interaction is confounded or aliased with each estimable effect. No replicates are included in a fractional factorial design and the analysis is conducted by making a normal or half-normal plot of the effects estimated by regression. The principles of effect sparsity and hierarchical ordering help in interpreting the results.

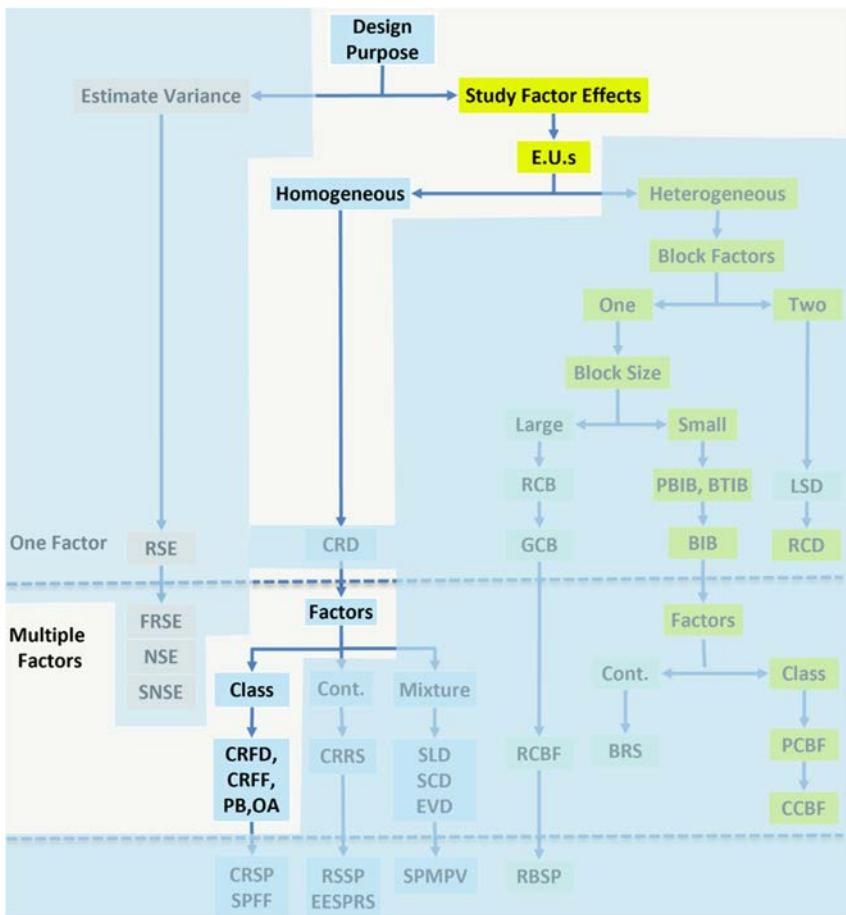
The subset of runs for one-quarter and higher fractions of  $2^k$  factorials can also be carefully selected by choosing generators that confound  $p$  added factors to interactions among the  $k - p$  factors in the base design. From the generators the defining relation and alias structure for the design can be determined. The `FrF2` function in the R package `FrF2` can generate designs that have the highest resolution and minimum aberration among all  $2^{k-p}$  designs with the same number of runs. Designs of resolution  $R$  confound main effects with  $R - 1$  and higher order interactions. These designs also have a projective property. When only

$R - 1$  factors or less appear significant after running a resolution  $R$  design, the insignificant factors can be ignored and a full factorial in  $R - 1$  factors will remain. The  $2^{k-p}$  fractional factorial projects to a  $2^{R-1}$  full factorial in  $R - 1$  factors.

After analysis of data from a resolution III fractional factorial, the interpretation may be straightforward if only one factor appears to have a significant effect. When more than one factor appears to have a significant effect, the interpretation may be more difficult due to aliasing of interactions with main effects. Augmenting a resolution III fractional factorial with a foldover or mirror image fraction can often eliminate the confounding between apparently important main effects and interactions and can allow for a simple interpretation of the effects after only a fraction of the runs needed for a full factorial. When confounded strings of two-factor interactions appear significant when analyzing data from a resolution IV design, augmenting the design with certain foldover fractions or  $D$ -optimal subsets of runs can provide the data to determine which interactions are significant.

Plackett-Burman and model robust screening designs preserve many of the benefits of factorial and fractional factorial designs such as orthogonality among main effects and the projective property. In addition, these designs are available with a wider choice of run sizes than there are for fractional factorial designs. Regression subset selection procedures can often detect the important main effects and interactions in a Plackett-Burman or model robust screening design without requiring additional experiments such as a foldover or  $D$ -optimal fraction.

Finally, when more than two levels are required for some factors, mixed level fractional factorials can be created using orthogonal array and orthogonal main effect plans that are generated with the `oa.design` function in the `DoE.base` package. These designs are similar to Plackett-Burman designs in that they are orthogonal in the main effects and have no defining relation. Many of these designs also have a projective property, and interactions can be detected without additional experiments using methods of analysis similar to those employed for Plackett-Burman designs.



*Figure 6-35. Design Selection Roadmap*

## 7. Incomplete and Confounded Block Designs

---

### 7.1. Introduction

One of the two purposes for randomized block designs, described in Chapter 4, was to group heterogeneous experimental units together in homogeneous subgroups called blocks. This increases the power or precision for detecting differences in treatment groups. The overall  $F$ -test for comparing treatment means, in a randomized block design, is a ratio of the variability among treatment means to the variability of experimental units within the homogeneous blocks.

One restriction on randomized block designs is that the number of experimental units in the block must be greater than or equal to the number of levels of the treatment factor. For the RCB design the number of experimental units per block,  $t$ , is equal to the number of levels of the treatment factor, and for the GCB design the number of experimental units per block is equal to the number of levels of the treatment factor times the number of replicates per block,  $tr$ .

When the number of levels of the treatment factor is large in a randomized block design, the corresponding number of experimental units per block must also be large. This could cause a problem. For example, frequently in agricultural field tests of new hybrid crops (called varietal trials) the number of hybrids could be very large (50–100). Since the experimental units are plots of land, the larger the number of plots included in a block the more diverse these plots will likely be since they cover a wider physical area. When experimental units are small animals, it is possible to group them into small blocks of littermates that are genetically more homogeneous. However, when larger block sizes are required, animals from different litters will be included within a block, making the groups less homogeneous. Less homogeneity within the blocks results in more variability among the experimental units within a block and defeats one of the purposes for using a randomized block design.

One solution to the problem is to construct block designs where each block only contains a subset of the possible levels of the treatment factor. In that way the number of experimental units per block, or block

size, can be kept small. Designs that do this are called **incomplete block designs**. There are two common types of incomplete block designs. The first type is called a balanced incomplete block design or BIB, and the other type is called a partially balanced incomplete block design or PBIB. Incomplete block designs are also useful when the physical requirements of experimentation make it impossible to test all levels of the treatment factor within each block of experimental units. Incomplete block designs are commonly used in agricultural experimentation, animal science, educational testing, and food science.

As an example of a situation where an incomplete block design would be useful, consider the following situation. In food science, taste panels are often used to test palatability of new recipes for new food products. In these panels, subjects are asked to taste and rate different recipes. The different recipes represent the levels of the treatment factor. A random sample of subjects (or the experimental units) is included to represent the potential market. Since there is wide variability in taste among subjects it is better to use a blocked design where each subject tastes and rates each recipe. However, if there are many recipes to be tested, subjects will lose their ability to discriminate, and it will be difficult to detect any differences. A solution is to have each subject (block) taste only a subset of the recipes.

## **7.2. *Balanced Incomplete Block (BIB) Designs***

Care must be taken when choosing the subset of treatment combinations tested in each block of an incomplete block design. If one treatment factor level is left out of every block, it cannot be compared to the other treatment levels. If different treatment levels are unequally represented, some comparisons of factor levels will have more precision than others.

The optimal way to create an incomplete block design is to have each treatment level equally replicated and appearing within a block with every other treatment level an equal number of times. This is called a balanced incomplete block design or BIB. By doing this, all pairwise differences of least squares treatment means will have the same standard error, and the power for detecting a difference in any two

means will be the same.

The simplest way to construct a BIB for the case where there are  $t$  levels of the treatment factor and  $k < t$  experimental units per block is to form all possible subsets of  $k$  treatment levels chosen from  $t$ . For example, in the taste panel described above, if there were  $t = 6$  recipes to be tested, and it was determined that each subject could taste at most  $k = 3$  recipes without losing discriminatory power,  $\binom{6}{3} = 20$  subjects would be required. All possible subsets are listed below.

$$\begin{array}{lllll} (1\ 2\ 3), & (1\ 2\ 4), & (1\ 2\ 5), & (1\ 2\ 6), & (1\ 3\ 4) \\ (1\ 3\ 5), & (1\ 3\ 6), & (1\ 4\ 5), & (1\ 4\ 6), & (1\ 5\ 6) \\ (2\ 3\ 4), & (2\ 3\ 5), & (2\ 3\ 6), & (2\ 4\ 5), & (2\ 4\ 6) \\ (2\ 5\ 6), & (3\ 4\ 5), & (3\ 4\ 6), & (3\ 5\ 6), & (4\ 5\ 6) \end{array}$$

Thus, subject one would taste recipes 1, 2, and 3 in a random order; subject 2 would taste recipes 1, 2, and 4, and so forth. This plan is completely balanced in that each treatment level or recipe is replicated  $r = 10$  times (or tasted by 10 subjects) and each pair of treatment levels occurs together in the same block  $\lambda = 4$  times. For example, treatment levels 1 and 2 occur together only in the first four blocks on the first line above. By inspection, it can be seen that all pairs of treatment levels occur together in only four blocks.

Although taking all possible subsets of size  $k$  chosen from  $t$  is the simplest way to form a balanced incomplete block design, there may be other balanced incomplete block designs that require fewer blocks. If the precision of the design does not require as many as  $\binom{b}{k}$  blocks, there would be no need to use that many. For example, if a practical size difference in recipes could be detected with less than 20 subjects and 10 replicates of each treatment level, in the taste test panel, perhaps a BIB design could be found with less than 20 blocks. If  $r$  is the number of times a treatment level is replicated in an incomplete block design,  $\lambda$  is the number of times each treatment level occurs with every other treatment level in the same block,  $t$  is the number of levels of the treatment factor,  $k$  is the number of experimental units in a block, and  $b$  is the number of blocks, then the following requirements must be met in order for that design to be a BIB design.

$$b \geq t$$

*Eq. 7-1*

$$Tr = bk$$

*Eq. 7-2*

$$\lambda(t - 1) = r(k - 1)$$

*Eq. 7-3*

I do not ;These relations can be used to find the minimum number of blocks required for a BIB design. Since  $r$  and  $\lambda$  must be integers, by *Eq. 7-3* we see that  $\lambda(t - 1)$  must be divisible by  $k - 1$ . If  $t = 6$  and  $k = 3$ , as in the taste test panel,  $5\lambda$  must be divisible by 2. The smallest integer  $\lambda$  for which this is satisfied is  $\lambda = 2$ . Therefore, it may be possible to find a BIB with  $\lambda = 2$ ,  $r = \frac{10}{2} = 5$ ,  $b = \frac{6 \times 5}{3} = 10$ . The function `BIBsize` in the R package `daewr` provides a quick way of finding values of  $\lambda$  and  $r$  to satisfy *Eq. 7-1* to *Eq. 7-3* when given the number of levels of the treatment factor  $t$  and the block size  $k$ . For example, in the code below the function is called with  $t = 6$  and  $k = 3$ .

```
library(daewr)
BIBsize(6, 3)
```

Possible BIB design with b= 10 and r= 5 lambda= 2

Fisher (1940) showed that even though *Eq. 7-1–Eq. 7-3* are satisfied for some combination of  $t$ ,  $b$ ,  $r$ ,  $\lambda$ , and  $k$ , a corresponding BIB may not exist. If a BIB does exist, Kiefer (1958) and Kshirsager (1958) have shown that it is  $D$ -optimal, thus it can be found using the R package `AlgDesign` that was described in Section 6.5.2. The `optBlock` function in that package can find BIB designs. For example, the code below searches for a BIB with  $b = 10$  blocks of  $k = 3$  experimental units per block and  $t = 6$  levels of the treatment factor. The option `blocksizes=rep(3,10)` specifies 10 blocks of size 3, and the option `withinData=factor(1:6)` specifies six levels of one factor.

```
library(AlgDesign)
BIB <- optBlock(~ ., withinData = factor(1:6),
blocksizes = rep(3, 10))
```

The object `BIB` created by `optBlock` contains a data frame `BIB$design` with one column that contains a list of treatment factor levels for each

of the 10 blocks in order. Creating the design again with the `optBlock` function may result in a different but equivalent BIB.

```
des <- BIB$rows
dim(des) <- NULL
des <- matrix(des, nrow = 10, ncol = 3, byrow = TRUE,
dimnames = list(c("Block1", "Block2", "Block3",
"Block4", "Block5", "Block6", "Block7", "Block8",
"Block9", "Block10"), c("unit1", "unit2",
"unit3")))
des
```

	unit1	unit2	unit3
Block1	1	3	6
Block2	2	3	4
Block3	1	4	6
Block4	1	2	5
Block5	4	5	6
Block6	1	3	5
Block7	2	3	6
Block8	1	2	4
Block9	3	4	5
Block10	2	5	6

According to this plan the three experimental units in the first block would receive treatment levels 1, 5, and 6, the experimental units in the second block would receive treatment levels 2, 3, and 4, and so forth. By inspection, it can be seen that each level of the treatment factor is repeated  $r = 5$  times in this design, and that every treatment level occurs within the same block with every other treatment level  $\lambda = 2$  times. Thus, we are assured that this is a balanced incomplete block design.

Once a balanced incomplete block (BIB) design is found, the levels of the treatment factor within each block should be randomized to the experimental units within that block, as illustrated for the RCBD in *Section 4.2*.

### 7.3. Analysis of Incomplete Block Designs

The model for an incomplete block design is

$$y_{ij} = \mu + b_i + \tau_j + \varepsilon_{ij}, \quad \text{Eq. 7-4}$$

which is identical to the model for a randomized complete block design given in **Section 4.3**. However, the analysis is slightly different due to the missing observations.

### 7.3.1. Example – Taste Test

Consider the data from a taste panel experiment reported by Moskowitz (1988), shown in **Table 7-1**. This experiment is a BIB with  $t = 4$  levels of the treatment factor or recipe, and block size  $k = 2$ . Thus each panelist tastes only two of the four recipes in a random order and assigns a category scale score. Category scales are commonly used in assessing food likes or dislikes and consist of numbers 1 to 10 that represent descriptive categories. Only  $\binom{4}{2} = 6$  blocks or panelists are required for a BIB, but in this experiment that number was doubled in order to increase the power for detecting differences. Thus the first six panelists and the last six are a repeat of the same BIB design. Subjects participating in the taste panel were randomly assigned to panelist numbers and the order of the two recipes tasted by each panelist was randomized.

**Table 7-1. Data from BIB Taste Test**

Panelist	Recipe			
	A	B	C	D
1	5	5	-	-
2	7	-	6	-
3	5	-	-	4
4	-	6	7	-
5	-	6	-	4
6	-	-	8	6
7	6	7	-	-
8	5	-	8	-
9	4	-	-	5
10	-	7	7	-
11	-	6	-	5
12	-	-	7	4

When analyzing the data from an incomplete block design, the marginal treatment means are not unbiased estimators of the estimable effects  $\mu + \tau_i$ . For example, in **Table 7-1** the marginal mean for recipe A could

be biased low by the fact that it was not tasted by panelists 4, 5, 6, 10, 11, and 12 who seem to rate recipes higher. Likewise, the noncentrality factor for the sequential sums of squares for treatments (or recipes) may contain block effects as well as treatment effects. The solution to these problems is the same as shown in [Section 3.5.3](#) for analyzing data from factorial designs with an unequal number of replicates per cell. The least squares means are used rather than marginal means, and the type III or adjusted sums of squares for treatments should be used. The `Anova` function in the `car` package can be used to get the type III or adjusted sums of squares for recipes as shown in [Section 3.5.3](#), or the `lm` function can be used with the `recipe` term ordered last in the model as shown in the code below. `lm` will not produce the type III or adjusted sums of squares for blocks or panelists, but since differences in panelists are not of interest they are not needed.

```
library(daewr)
mod1 <- aov(score ~ panelist + recipe, data = taste)
summary(mod1)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)						
panelist	11	19.333	1.7576	2.301	0.1106						
recipe	3	9.125	3.0417	3.982	0.0465 *						
Residuals	9	6.875	0.7639								
	---										
Signif. codes:	0	'***'	0.001	'**'	0.01	'*'	0.05	.	0.1	' '	1

The  $F_{3,9}$  value for testing recipes was 3.982, which is significant at the  $\alpha = 0.05$  significance level. The adjusted or least squares means can be calculated using the `predict` command and the `tapply` function as shown in [Section 3.5.3](#), but it can be done with less coding using the `lsmeans` package. The code below using `lsmeans` computes the adjusted means, their standard errors and confidence intervals, and makes Tukey comparisons of all pairs of means.

```
library(lsmeans)
lsmeans(mod1, pairwise ~ recipe, adjust = ("tukey"))

$lsmeans
  recipe lsmean      SE df lower.CL upper.CL
  A       5.46 0.418  9     4.51     6.40
  B       6.21 0.418  9     5.26     7.15
```

C	6.83	0.418	9	5.89	7.78
D	4.83	0.418	9	3.89	5.78

Results are averaged over the levels of: panelist  
 Confidence level used: 0.95

```
$contrasts
contrast estimate    SE df t.ratio p.value
A - B     -0.750 0.618 9  -1.214  0.6342
A - C     -1.375 0.618 9  -2.225  0.1882
A - D      0.625 0.618 9   1.011  0.7472
B - C     -0.625 0.618 9  -1.011  0.7472
B - D      1.375 0.618 9   2.225  0.1882
C - D      2.000 0.618 9   3.236  0.0421
```

Results are averaged over the levels of: panelist  
 P value adjustment: tukey method for comparing a family of 4 estimates

The results show the average score for recipe *C* is significantly higher than the score for recipe *D* at the  $\alpha = 0.05$  level, but that no other differences are significant at the  $\alpha = 0.05$  level. Another thing that should be noticed in the output is the fact that the standard error of the differences in means is the same for all pairs of means. This is a result of the fact that a BIB design was used.

### 7.3.2. Determining the Number of Replicates

A rough estimate of the number of replicates of each treatment level,  $r = bk/t$  or the number of blocks,  $b = tr/k$  required for adequate power for detecting a practical difference in treatment means in a BIB design can be determined using the following strategy. If an estimate of  $\sigma^2$ , the variance of heterogeneous experimental units, and  $\Delta$ , the size of a practical difference in two treatment means are available, then use the method of **Section 3.5.2** to determine the number of replicates,  $r_{crd}$ , required for a completely randomized design.

If blocking the heterogeneous experimental units into small blocks of size  $k$  is expected to reduce the variance of experimental units within the blocks by a percentage equal to  $100 \times \left(1 - \frac{1}{RE}\right)$ , then following **Section 4.6**, the number of replicates required for the blocked design would be  $r = r_{crd}/RE$  and the number of blocks of size  $k$  in an

incomplete block would be  $b = tr/k$ .

#### 7.4. BTIB and PBIB Designs

Sometimes a BIB design requires more blocks or experimental units than are available or needed to obtain adequate power for detecting practical differences in treatment means. In this situation, the number of blocks and experimental units can be reduced by relaxing the requirements that: (1) each treatment level be equally replicated and (2) that it appears within a block with every other treatment level the same number of times. By relaxing these requirements, each treatment level will occur more frequently in a block with some treatment levels than it will with other treatment levels. Therefore some pair-wise comparisons of treatment means will have smaller standard errors than others.

In other situations, a BIB design cannot be used because physical constraints of experimentation prevent some treatment levels from being tested together within the same block. For example, an experiment was conducted to compare the reading of a portable home use blood pressure monitor to other automatic blood pressure monitors in supermarket pharmacies in order to determine if readings from the portable monitor were biased high. The experimental unit was the pressure in a subject's veins at the time it was measured, and the treatment factor levels were the monitors used to make the measurement.

Blood pressure is known to vary widely from person to person and within a person at different times throughout a day. Blood pressure is most consistent in one person within a short period of time. Therefore the experimental units were blocked into homogeneous groups by subject and time. The portable blood pressure monitor could be carried into a store and a subject's blood pressure could be checked within a short period of time by both the automatic monitor within the store and the portable monitor. However, the monitors from two stores could not be compared within a block, or short period of time, because the stores were physically separate. Driving between stores could completely change a subject's blood pressure, so the incomplete block design shown in **Table 7-2** was utilized. The response (diastolic blood pressure)

is shown in the table.

**Table 7-2. Incomplete Block Design with Blood Pressure Monitors**

Block	Treatment			
	Portable Monitor	Store A Monitor	Store B Monitor	Store C Monitor
1=(subject 1, time 1)	85	77	-	-
2=(subject 2, time 1)	80	75	-	-
3=(subject 1, time 2)	89	-	73	-
4=(subject 2, time 2)	80	-	70	-
5=(subject 1, time 3)	78	-	-	76
6=(subject 2, time 3)	80	-	-	70

Here we can see that treatment level 1 (portable monitor) appears in a block with every other treatment level, but the other treatment levels never appear together in a block. The code to analyze the data using R `aov` function and the `lsmeans` function is shown below.

```
library(daewr)
modm <- aov( pressure ~ Block + Treatment,
             data = BPmonitor)
library(lsmeans)
lsmeans(modm, pairwise ~ Treatment, adjust = ("tukey"))
```

In the resulting table of comparisons of means, shown at the top of the next page, it can be seen that the standard errors of the differences between the portable monitor (*P*) mean and the means for the other monitors (*A*, *B*, and *C*) are smaller than the standard errors of the comparisons between *A* and *B*, *A* and *C*, and *B* and *C*.

```
$lsmeans
Treatment lsmean   SE df lower.CL upper.CL
A          75.5 2.75  3    66.7    84.3
B          69.0 2.75  3    60.2    77.8
C          76.0 2.75  3    67.2    84.8
P          82.0 1.23  3    78.1    85.9
```

Results are averaged over the levels of: Block  
Confidence level used: 0.95

```
$contrasts
contrast estimate   SE df t.ratio p.value
```

A - B	6.5	4.26	3	1.525	0.5212
A - C	-0.5	4.26	3	-0.117	0.9993
A - P	-6.5	3.01	3	-2.157	0.3110
B - C	-7.0	4.26	3	-1.642	0.4744
B - P	-13.0	3.01	3	-4.313	0.0670
C - P	-6.0	3.01	3	-1.991	0.3564

Results are averaged over the levels of: Block

P value adjustment: tukey method for comparing a family of 4 estimates

The design for the blood pressure monitor experiment is a special case of an incomplete block design that Bechhofer and Tamhane (1981) have called a BTIB (balanced with respect to test treatments). In these designs one treatment level is designated as the control level and there is more interest in comparing each of the other treatment levels with the control than there is in comparing the other treatment levels. In a BTIB design each treatment must appear the same number of times ( $\lambda_0$ ) in a block with the control treatment, and each test treatment must occur the same number of times ( $\lambda_1$ ) in a block with every other test treatment. This results in a design that is more efficient in comparing each treatment with the control but less efficient in comparisons among the other treatment levels. One way to form a BTIB design with  $t$  levels of the treatment factor and block size  $k$  is to combine a control level to each block of a BIB design with  $t - 1$  levels of the treatment factor and block size  $k - 1$ .

The BTIB design, described in the last paragraph, is a special case of a **partially balanced incomplete block** design (or PBIB) with two associate classes. In these designs each pair of treatments are either first associates or second associates. First associates occur together in a block  $\lambda_1$  times, and second associates occur together in a block  $\lambda_2$  times, where  $\lambda_1 > \lambda_2$ . The standard error of the difference in treatment means for first associates is smaller than the standard error of the difference in means for second associates. There are multiple ways of creating PBIB designs. Bose et al. (1954) have published tables of some of the most useful plans. Jarrett and Hall (2012) have described a class of PBIB designs called generalized cyclic incomplete block designs, which have good statistical properties and are easy to create. Generalized cyclic

incomplete block designs with block size  $k$  and  $b = t$  blocks can be created following the steps listed below.

1. To form a generalized cyclic design with  $b = t$ ,
  - a. Start with a subset of  $k$  treatment factor levels as the initial block.
  - b. Add 1 (modulo  $t$ ) to each treatment level in the initial block to form the next block.
  - c. Continue adding blocks until you have  $t$  blocks.

To illustrate this, consider creating an incomplete block design with  $t = 6$  and  $k = 3$ . The BIB design with the smallest number of blocks for this combination is found (solving [Eq. 7-1](#) to [Eq. 7-3](#) to be  $b = 10$ ). A generalized cyclical incomplete block design for testing  $t = 6$  levels of the treatment factor can be found with  $b = t = 6$  blocks. To find a design with six blocks, following the steps above, start with a subset of  $k = 3$  levels of the treatment factor to be tested in the initial block, that is,

$$(1 \ 2 \ 4)$$

Next add one to each treatment level to get the treatment levels in the next block, that is,

$$(2 \ 3 \ 5)$$

Continue this modulo 6 (i.e., 7 modulo 6 = 1, etc.) to form the following.

Block	Treatments
1	1    2    4
2	2    3    5
3	3    4    6
4	4    5    1
5	5    6    2
6	6    1    3

The treatment levels in each block would be randomized to the three experimental units in each block. The function `design.cyclic` in the R package `agricolae` can create generalized cyclic designs with  $t$  blocks of size  $k$ . For example, the code below shows the commands to create a generalized cyclic design with  $t = 6$ ,  $k = 3$ . The first argument for `design.cyclic` is a vector containing the levels of the treatment factor.

The second argument is the block size,  $k$ , and the third argument is the number of replicates,  $r$ , of each treatment level. In this case since there are six blocks of three experimental units there is a total of 18 experimental units and each treatment level will be assigned to three, making  $r = 3$ .

```
library(agricolae)
treat <- c(1, 2, 3, 4, 5, 6)
des <- design.cyclic(treat, k = 3, r = 3)
```

```
cyclic design
Generator block basic:
1 2 4
Parameters
=====
treatmeans : 6
Block size : 3
Replication: 3
```

The function `design.cyclic` generates and randomizes the plan. In this example of a PBIB design, each treatment level has one first associate with  $\lambda_1 = 2$  and four second associates with  $\lambda_2 = 1$ . The object created by `design.cyclic` has two components. The component `des$book` contains the design as shown below.

```
des$book
```

	plots	group	block	treat
1	101	1	1	6
2	102	1	1	2
3	103	1	1	5
4	104	1	2	6
5	105	1	2	4
6	106	1	2	3
7	107	1	3	5
8	108	1	3	1
9	109	1	3	4
10	110	1	4	6
11	111	1	4	1
12	112	1	4	3
13	113	1	5	4
14	114	1	5	1
15	115	1	5	2
16	116	1	6	3

17	117	1	6	2
18	118	1	6	5

The analysis of PBIB designs is the same as the examples shown for BIB designs. The type III sums of squares for treatment and the least squares means should be used. The model and assumptions are the same as for the RCB design, and the assumptions of normality and homogeneity of experimental error variance can be checked with the residual plots described in **Section 2.4**.

## 7.5. Row Column Designs

Latin-square designs with two independent blocking factors were described in Chapter 4. These designs could increase the precision in detecting differences among treatments by adjusting for variability in experimental units in two ways. However, the restriction on Latin-square designs was that the number of levels of the row blocking factor, the number of levels of the column blocking factor, and the number of levels of the treatment factor all had to be equal. This restriction may be impractical in some situations.

In Chapter 4, an experiment was described where the purpose was to study the effect of shelf facing on the sales of toothpaste in drugstores. In that example, four levels of the treatment factor (shelf facings), four levels of the row blocking factor (stores), and four levels of the column blocking factor (week of sales) were used. If the researchers desired to expand their study to test eight different shelf facings instead of four, they could easily increase the number of levels of the row blocking factor and include eight stores. However, increasing the number of weeks would prolong the study and could be undesirable.

An alternate design called a Row Column design or RCD utilizes a complete block design in the column blocks, but an incomplete block design in the row blocks. This type design can also be created and randomized by the function `design.cyclic` in the R package `agricolae`. The code below illustrates how this can be done. The addition of the argument `rowcol=TRUE` causes `design.cyclic` to create a row column design. Since there are eight levels of the treatment factor and column block size  $k = 4$ , there will be  $r = 4$  replicates of each

treatment level in the design. The argument `seed = 1` fixes the random seed so the same design can be produced in repeat runs.

```
library(agricolae)
treat <- c(1, 2, 3, 4, 5, 6, 7, 8)
RCD <- design.cyclic(treat, k = 4, r = 4,
                      rowcol = TRUE, seed = 1)
```

```
cyclic design
Generator block basic:
1 2 4 8
```

```
Parameters
=====
treatmeans : 8
Block size : 4
Replication: 4
```

This code will create a design with eight row blocks for stores, and four column blocks for weeks. The model and analysis of RCDs is identical to the model and analysis of Latin-square designs described in Chapter 4, with the exception that type III treatment sums of squares and least squares treatment means should be used due to the fact that the row blocks are incomplete and do not contain all levels of the treatment factor.

## 7.6. Confounded $2^k$ and $2^{k-p}$ Designs

When the experimental units are heterogeneous and can be grouped into blocks, frequently there are too few experimental units per block to accommodate all the treatment combinations in  $2^k$  factorial or  $2^{k-p}$  fractional factorial design. Therefore, it is impossible to use complete block designs, like those described in [Section 4.6](#). One solution is to use an incomplete block design described in the first part of this chapter with  $t = 2^k$ . However, this usually results in many more blocks than are necessary. The effect sparsity principle and hierarchical ordering principle tell us that it is unlikely that all interactions (especially those above order 3) will be important. If we are willing to sacrifice the ability to estimate some interactions,  $2^k$  factorial designs can be blocked in a minimum number of blocks as long as the number of experimental units in each block is a power of 2.

For example, a  $2^3$  factorial design has eight treatment combinations. By confounding the three-factor interaction with blocks, this design can be run in two blocks of size 4 as shown in **Table 7-3** below.

In this design, the ABC interaction effect is completely confounded between the two blocks and cannot be estimated. However, since interactions between block and treatment effects are assumed negligible, all the other main effects and two-factor interactions are not confounded. We can detect the significant effects from an experiment like this using graphical methods illustrated in **Section 3.8.1** and Chapter 6.

**Table 7-3.  $2^3$  in Two Blocks of Size 4**

A	B	C	Block=	ABC
-	-	-	1	-
+	-	+	1	-
-	+	+	1	-
+	+	-	1	-
-	-	+	2	+
+	-	-	2	+
-	+	-	2	+
+	+	+	2	+

### 7.6.1. Confounding $2^k$ Designs

In general a  $2^k$  factorial can be run in blocks of size  $2^q$  by choosing  $k - q$  interaction contrasts to confound with blocks. These interactions are called block defining contrasts. When a  $2^k$  is run in blocks of size  $2^q$ , there will be  $\frac{2^k}{2^q} = 2^{k-q}$  blocks. Therefore, there will be  $2^{k-q} - 1$  degrees of freedom for blocks. These  $2^{k-q} - 1$  degrees of freedom are accounted for by the  $k - q$  defining block contrasts and all their generalized interactions.

As an example consider confounding a  $2^4$  factorial in blocks of size  $4 = 2^2$ . Since  $k = 4$  and  $q = 2$ ,  $k - q = 2$  defining block contrasts must be chosen. If the three-factor interactions ABD and BCD are chosen as the block defining contrasts, their generalized interaction  $ABD(BCD) = AC$  will also be confounded with blocks, accounting for the 3 degrees of freedom for blocks. The `FrF2` function in the R package `FrF2` can find

this design as shown in the code below.

```
library(FrF2)
Bhelm <- FrF2( 16, 4, blocks = c("ABD", "BCD"),
               alias.block.2fis = TRUE, randomize = FALSE)
```

The design shown on the next page is called a completely confounded blocked factorial or CCBF. In practice the treatment combinations within each block would be randomized to experimental units in that block by removing the option `randomize = FALSE`.

`Bhelm`

```
run.no run.no.std.rp Blocks A B C D
1       1      1.1.1     1 -1 -1 -1
2       2      6.1.2     1 -1  1 -1  1
3       3     12.1.3     1  1 -1  1  1
4       4     15.1.4     1  1  1  1 -1
run.no run.no.std.rp Blocks A B C D
5       5      3.2.1     2 -1 -1  1 -1
6       6      8.2.2     2 -1  1  1  1
7       7     10.2.3     2  1 -1 -1  1
8       8     13.2.4     2  1  1 -1 -1
run.no run.no.std.rp Blocks A B C D
9       9      4.3.1     3 -1 -1  1  1
10     10      7.3.2     3 -1  1  1 -1
11     11      9.3.3     3  1 -1 -1 -1
12     12     14.3.4     3  1  1 -1  1
run.no run.no.std.rp Blocks A B C D
13     13      2.4.1     4 -1 -1 -1  1
14     14      5.4.2     4 -1  1 -1 -1
15     15     11.4.3     4  1 -1  1 -1
16     16     16.4.4     4  1  1  1  1
class=design, type= FrF2.blocked
NOTE: columns run.no and run.no.std.rp are annotation, not
part of the data frame
```

The analysis of a confounded blocked factorial is similar to the analysis of an unreplicated  $2^k$  design as described in [Section 3.6](#). As an example consider the experiment conducted by Lephart, et.al (2013) (DOD IG, 2013). The researchers were interested in the effects of the four factors shown in [Table 7-4](#) upon advanced combat helmets (ACH), which we examine in [Section 4.4](#).

**Table 7-4. Factors for ACH Penetration Experiment**

Factor	Levels	
	(-)	(+)
A—enviro	Ambient	Cold
B—sides	Left	Right
C—shots	One	Two
D—size	Medium	Large

ACH is designed to be resistant to a 9mm Full Metal Jacketed Round Nose (FMJ RN) Remington bullet penetration with a nominal mass of 124 grains. The experiment consisted of firing 240 shots at 48 helmets with the factors in **Table 7-4**. He drew a  $10 \times 10$  grid on a white plate with a permanent marker, and his response was the number of clean grid squares after soaking the plate with baked-on spaghetti sauce in the dishwasher.

The experimental unit was the helmets, and in order to generalize the conclusions, the researchers included four blocks which consisted of combinations of environmental conditions: ambient, hot, cold, and sea water. In this experiment, it would have been possible to run a complete block factorial design where all combinations of the treatment factor levels from **Table 7-4** were included in each block. This would have taken a total of  $4 \times 16 = 64$  experiments. However, considering the hierarchical ordering principle, some interactions were confounded in order to reduce the total number of experiments. Using the same design shown earlier, *ABD* and *BCD* and their generalized interaction *ABD(BCD) = AC* were confounded with blocks, resulting in 16 experiments or plates grouped into 4 blocks with 4 experimental units in each block. The design and response (number of clean grid squares) is shown in **Table 7-8**.

**Table 7-5. Design matrix the Advanced Combat Helmet**

Size (Block)	Ambient		Hot		Cold		Seawater	
	Shot 1	Shot 2	Shot 1	Shot 2	Shot 1	Shot 2	Shot 1	Shot 2
Small	Cr	B	L	F	Cr	B	L	F
	L	F	Cr	B	R	F	Cr	B
	Cr	B	R	F	Cr	B	R	F
Medium	R	F	Cr	B	R	F	Cr	B
	Cr	B	R	F	Cr	B	L	F
	L	F	Cr	B	L	F	Cr	B

	Cr	B	R	F	Cr	B	R	F
Large	R	F	Cr	B	L	F	Cr	B
	Cr	B	L	F	Cr	B	L	F
X-Large	L	F	Cr	B	L	F	Cr	B
	Cr	B	L	F	Cr	B	R	F
	R	F	Cr	B	R	F	Cr	B

**Table 7-6. Blocks for Dishwashing Experiment**

9 mm RTP shell	Ambient 70° F	Hot 160° F	Cold -60° F	Immerse in Seawater, test at 70° F	Totals
DOT&E test protocol sample size	60 shots 12 helmets	60 shots 12 helmets	60 shots 12 helmets	60 shots 12 helmets	240 shots 48 helmets

The **Clopper-Pearson** method is used to calculate the allowable number of penetrations out of a given sample size. When applied against the required parameters (90/90 and 240 shots), the Clopper-Pearson method yields 17 allowable penetrations.

**Table 7-7. DOT&E LAT RTP for Lot Acceptance Testing**

9 mm RTP shell	Lot Size	Sample Size	Accept	Reject
	91-150	25 shots 5 helmets	0	1
	151-500	40 shots 8 helmet	1	2
	501-1,200	65 shots 13 helmets	1	2
	1,200+	65 shots 13 helmets	1	2

To analyze the data, the factorial model was fit that included all main effects and interactions. The R code to do this using the `lm` function are shown below.

```
y <- c(0, 0, 12, 14, 1, 0, 1, 11, 10, 2, 33, 24, 3, 5,
41, 70)
Bhelm <- add.response(Bhelm, response = y)
Hel_mod1 <- lm( y ~ Blocks + A * B * C * D,
data = Bhelm)
```

Since  $ABD$ ,  $BCD$ , and  $AC$  were confounded with blocks the `lm` function assigns the value of NA for their effects. Because there were no replicates in the experiment, a half-normal plot of the effects was used

to determine the significant effects and interactions. The code below produces the half-normal plot. The second and third statements select the non-block effects that do not have the value of NA, and the last statement adds a reference line to the plot.

```
effects <- coef(Hel_mod1)
effects <- effects[5:19]
effects <- effects[ !is.na(effects) ]
library(daewr)
halfnorm(effects, names(effects), alpha = .25)
```

The half-normal plot of effects is shown in *Figure 7-1*.

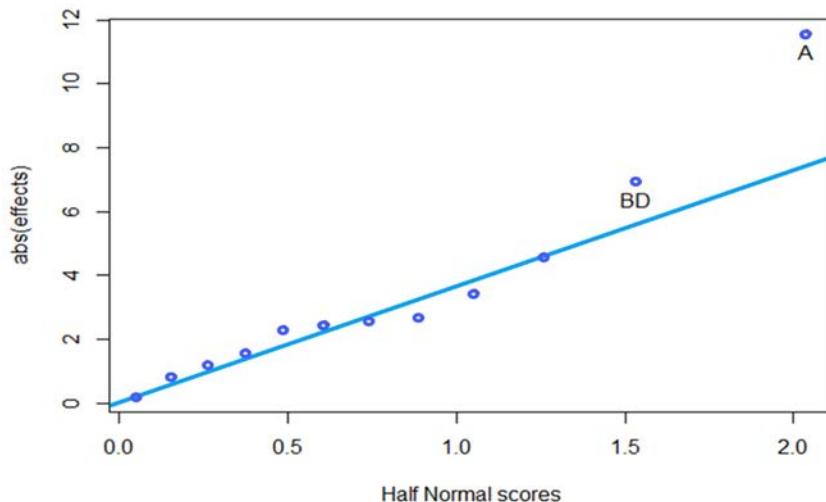
*Table 7-8. Design and Data for ACB Experiment*

A	B	C	D	Block	y
-	-	-	-	1	0
+	+	+	-	1	14
-	+	-	+	1	0
+	-	+	+	1	12
+	-	-	-	2	33
-	+	+	-	2	2
+	+	-	+	2	24
-	-	+	+	2	10
+	+	-	-	3	11
-	-	+	-	3	1
+	-	-	+	3	1
-	+	+	+	3	0
-	+	-	-	4	5
+	-	+	-	4	41
-	-	-	+	4	3
+	+	+	+	4	70

In *Figure 7-1* we can see that main effect A (upper-right), the environmental condition, and possibly the BD interaction between side strength and shot number appear to be significant. The environmental condition main effect was easy to interpret. The means for this factor revealed that 23.125 more shot penetrations occurred with a seawater environment than with cold water.

To interpret the *BD* interaction an interaction plot or table of means is required. *Figure 7-2* shows the interaction plot. Here it can be seen that increasing the exposure time in the Seawater environment from 1 day

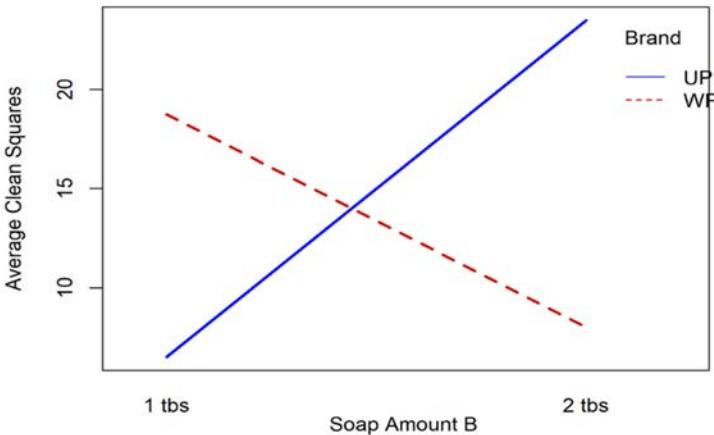
to 2 days increases the number of penetrations in the ACH, while increasing the exposure time from 1 day to 2 days has the opposite effect and decreases the number of penetrations.



**Figure 7-1. Half-Normal Plot of Absolute Regression Coefficients from ACH Experiment**

According to the effect heredity principle, it would be unusual to have an interaction between two factors that do not have significant main effects. This, along with the fact that the *BD* interaction does not appear to be significant using the graphical techniques of Box and Meyer (1986a), Lawson et al. (1998), or Lenth (1989) would justify ignoring this unusual finding. Therefore, the only thing that seems to have a significant effect on the penetrability of the ACH is the Seawater environment..

Blocks (not shown in the normal plot) also accounted for a large proportion of the sums of squares. If the block term were not included in the model, none of the factor effects would be determined significant. If experimental units had been restricted to one environmental condition and one exposure time, the conclusions regarding the penetrability would only apply to that environment and exposure time.,



*Figure 7-2. Interaction Plot for Soap Amount  $\times$  Soap Brand*

### 7.6.2. Confounding in $2^{k-p}$ Designs

A  $2^{k-p}$  fractional factorial can also be easily blocked by confounding interactions if the block size is a power of two. For example, consider the  $2^{5-2}$  fractional factorial shown in **Section 6.3**. In **Table 7-9** the eight runs in this design are split into two blocks by using the BC interaction as the block defining contrast. When there is only one block defining contrast in a full factorial design, only that interaction is confounded with the difference in blocks. However, in a one-fourth fractional factorial, each effect that can be estimated is aliased with three other effects due to the fractionation.

*Table 7-9. Blocked  $2^{5-2}$  Design*

A	B	C	D	E	Block=	BC
-	-	-	+	+	2	+
+	-	-	-	-	2	+
-	+	-	-	+	1	-
+	+	-	+	-	1	-
-	-	+	+	-	1	-
+	-	+	-	+	1	-
-	+	+	-	-	2	+
+	+	+	+	+	2	+

In the example above the generators for the quarter fraction were  $D=AB$  and  $E = AC$ , resulting in the defining relation for the fraction  $I = ABD = ACE = BCDE$ . Multiplying through the defining relation for

the fractional factorial by the block defining contrast, we see that  $BC = ACD = ABE = DE$  and that four interactions are actually confounded with the blocks.

In general, when a  $2^{k-p}$  fractional factorial is blocked into blocks of size  $2^q$ , there will be  $\frac{2^{k-p}}{2^q} = 2^{k-p-q}$  blocks.  $k - p - q$  interactions must be chosen as block defining contrasts. The block defining contrasts and all their generalized interactions will account for the  $2^{k-p-q} - 1$  degrees of freedom for blocks, and each interaction that is confounded will also be aliased with  $2^{p-1}$  other interactions due to the fractionization. These designs are called completely confounded blocked fractional factorial or *CCBFF*.

To illustrate the design and analysis of a confounded  $2^{k-p}$  design, consider the experiment described by Abt, et. al. The purpose was to explore the effects of the Army 101<sup>st</sup> Airborne Division Eagle Tactical Athlete Program, comprise of eight factors shown in **Table 7-10** on reduction of musculoskeletal injury. They used a 1/16th fractional factorial design to identify variables affecting the ability highly, physically active soldiers resist musculoskeletal injury in the line of duty. This was a screening experiment with the purpose of identifying significant main effects that would be studied later in further experiments.

The generators for the  $2^{8-4}$  fractional design were  $E = BCD$ ,  $F = ACD$ ,  $G = ABC$ , and  $H = ABD$ , resulting in the defining relation

$$\begin{aligned} I &= BCDE = ACDF = ABCG = ABDH = ABEF = ADEG = ACEH \\ &= BDFG = BCFH = CDGH = CEFG = DEFH = BEGH = AFGH \\ &= ABCDEF GH \end{aligned}$$

**Table 7-10. Factors and Levels for Injury Prevention Experiment**

Factor	Description	Levels	
		-	+
A	Health Diet	None	Free Access
B	Frequency of weighing	Once per 3 day	Once per day
C	Physical Therapist	No	Yes
D	Nutritionist	No	Yes
E	Standard Training	Yes	No

F	ETAP Program	Locked	Free
G	Available Food	Free access	80 percent
H	Available Water	80 percent	Free access

This was a resolution IV fraction and the eight main effects were confounded with strings of three-factor and higher-order interactions. In addition, there were seven strings of estimable confounded two-factor interactions listed below.

$$\begin{aligned}
 & AB + CG + DH + EF \\
 & AC + BG + DF + EH \\
 & BC + AG + DE + FH \\
 & AD + BH + CF + EG \\
 & BD + AH + CE + FG \\
 & CD + AF + BE + GH \\
 & CH + AE + BF + DG
 \end{aligned}$$

The experimenters desired to block the experiments into eight blocks of  $2^1 = 2$  runs each. This would allow them to prevent biases from uncontrolled variables, such as time of year, over the duration of their experiments. Doing this would require choosing  $k - p - q = 8 - 4 - 1 = 3$  block defining contrasts. These block defining contrasts, along with their generalized interactions, would account for the  $8 - 1 = 7$  degrees of freedom for blocks.

The block defining contrasts that were chosen were  $AB + CG + DH + EF$ ,  $AC + BG + DF + EH$ , and  $AD + BH + CF + EG$ . The two-factor generalized interactions are  $AB(AC) = BC$  (or  $BC + AG + DE + FH$ ),  $AB(AD) = BD$  (or  $BD + AH + CE + FG$ ), and  $AC(AD) = CD$  (or  $CD + AF + BE + GH$ ). Finally, the three-factor generalized interaction can be identified by substituting  $BH$  for  $AD$ , since  $AD$  is confounded with  $BH$  to get  $AB(AC)(AD) = AB(AC)(BH) = CH$ , or  $CH + AE + BF + DG$ . Therefore, using the three block defining contrasts  $AB + CG + DH + EF$ ,  $AC + BG + DF + EH$ , and  $AD + BH + CF + EG$  actually confounds all of the seven confounded strings of two-factor interactions (shown above) with blocks.

The researchers were willing to give up information on all two-factor interactions to prevent biases because this was a screening experiment where they were satisfied to get information about the relative

importance of the main effects. The design and results are shown in **Table 7-11**.

**Table 7-11. Design and Results for Injury Prevention Experiment**

Block	Factors								weight
	A	B	C	D	E	F	G	H	
1	+	-	-	-	-	+	+	+	9
1	-	+	+	+	+	-	-	-	0
2	-	-	+	+	-	-	+	+	9.25
2	+	+	-	-	+	+	-	-	4.9
3	+	-	+	-	+	-	-	+	8.8
3	-	+	-	+	-	+	+	-	4.35
4	-	-	-	+	+	+	-	+	0
4	+	+	+	-	-	-	+	-	7.43
5	-	+	+	-	-	+	-	+	5.35
5	+	-	-	+	+	-	+	-	9.9
6	+	+	-	+	-	-	-	+	2.6
6	-	-	+	-	+	+	+	-	7.43
7	-	+	-	-	+	-	+	+	6.8
7	+	-	+	+	-	+	-	-	3.93
8	+	+	+	+	+	+	+	+	10.2
8	-	-	-	-	-	-	-	-	4.87

### 7.6.3. Objective

The study objective was to validate the Eagle Tactical Athlete Program (ETAP) to modify suboptimal strength, performance, and Army Physical Fitness Test variables.

### 7.6.4. Design

The design was a randomized controlled trial. The experiment was performed university-operated, military human performance research laboratory.

### 7.6.5. Participants

A total of 57 soldiers of the 101<sup>st</sup> Airborne Division (Air Assault) participated (Experimental- N: 30, age:  $25.0 \pm 5.2$  years, height:  $173.4 \pm$

8.3 cm, mass:  $76.6 \pm 11.3$  kg, Control-  $N: 27$ , age:  $25.0 \pm 5.8$  years, height:  $175.6 \pm 8.5$  cm, mass:  $76.5 \pm 11.6$  kg) participated.

### 7.6.6. Interventions

Pre- and post-test measurements were captured for strength, performance, and Army Physical Fitness Test variables. Subjects were randomly assigned to an experimental or control group. The experimental group performed an eight-week clinical trial of ETAP, which was based on the results from 21 months of laboratory data collected on soldiers of the 101<sup>st</sup> Airborne Division. ETAP followed a sports medicine periodized training model and included specific modalities designed to improve athleticism. The periodized training program was also developed to specifically address and maximize each athletic and skill-related performance component to ensure the tactical athletes are a viable force for deployment into the demands of the current conflict. The control group performed standard physical training according to FM 3-22.20. This trial was designed to induce adaptations in variables known to contribute to injury and limit performance.

### 7.6.7. Main Outcome Measures

Knee, shoulder, and torso strength, body fat, anaerobic power and capacity, performance tests, and the Army Physical Fitness Test. Two-way repeated measures ANOVA tests were used to analyze the dependent variables.

The two treatment combinations within each block were randomized to the soldiers in that block. The code to create this blocked fractional factorial using the `FrF2` function in the R package `FrF2` is shown below.

```
library(FrF2)
Bff <- FrF2(16, 8, generators = c("BCD", "ACD", "ABC",
"ABD"),
blocks = c("AB", "AC", "AD"), randomize = FALSE)
weight <- c(0.0, 9.0, 5.35, 9.90, 4.35, 8.8, 6.8, 3.93,
         9.25, 4.9, 7.43, 2.6, 0.0, 7.43, 4.87, 10.2)
add.response(Bff, response = weight)
```

The `generators = option` specifies the generators (interactions confounded with added factors) for the 1/16<sup>th</sup> fractional factorial, and

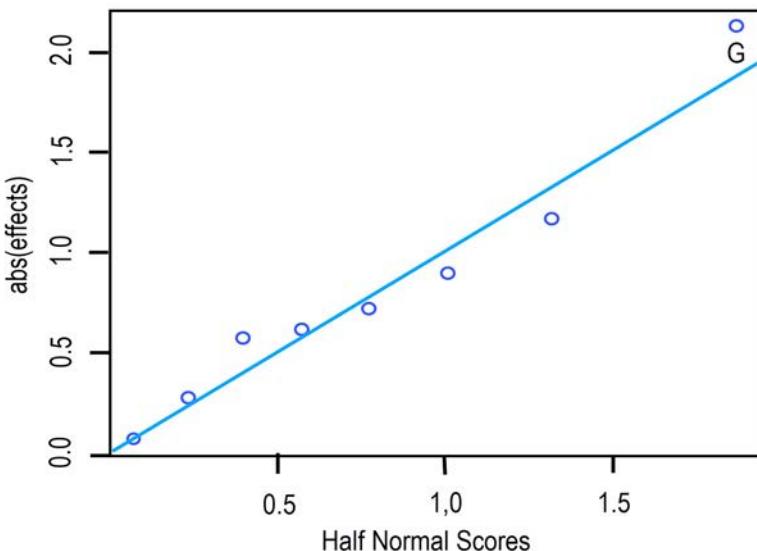
the `blocks = option` specifies the block defining contrasts.

To analyze the data a model involving only the main effects  $A - H$  and the block effects can be fit, since they account for all the degrees of freedom. The R code to fit this model is shown below.

```
injury <- lm(weight ~  
Blocks + A + B + C + D + E + F + G + H, data = Bff)
```

Since this was a saturated model with zero degrees of freedom for the error term, the significance of effects could be accessed using the graphical techniques described earlier. The code to make the half normal plot of the effects in **Figure 7-3** is shown below. The reference line was again added manually.

```
effects <- coef(injury)  
effects <- effects[ 9:16 ]  
library(daewr)  
halfnorm(effects, names(effects), alpha = .15)
```



**Figure 7-3. Half-Normal Plot of Absolute Regression Coefficients from Injury Prevention Experiment**

From the half-normal plot it is clear that the only factor having a significant effect on the weight was  $G$ —available food. The calculated

effect showed that the weight was on the average 4.24kg higher for soldiers given free access to food compared to the treatment group whose food intake was restricted to 80% of normal Army rations. This was the conclusion that the authors of the article reached.

### 7.6.8. Criteria for Choosing Block Defining Contrasts

Care must be exercised when choosing the block defining contrasts for  $2^k$  full factorial or  $2^{k-p}$  fractional factorial designs. To illustrate why, consider the following two examples.

**Example 1** consists of blocking a  $2^5$  factorial into 8 blocks of size  $2^2 = 4$ . To do this  $k - q = 5 - 2 = 3$  block defining contrasts must be chosen. Suppose the interactions  $ABC$ ,  $CDE$ , and  $ABCDE$  are chosen as the block defining contrasts. Their two-factor generalized interactions  $ABC(CDE) = ABDE$ ,  $ABC(ABCDE) = DE$ ,  $CDE(ABCDE) = AB$ , and their three-factor generalized interaction  $ABC(CDE)(ABCDE) = ABC(AB) = C$  are also confounded, accounting for the seven degrees of freedom for blocks. The block defining contrasts were all three-factor interactions and above, yet one of their generalized interactions is main effect  $C$  that we would not want to confound with blocks. A better choice would be to choose  $ACE$ ,  $BCE$ , and  $ABCD$  as block defining contrasts. Their generalized interactions include no main effects.

**Example 2** consists of blocking a  $2^{6-2}$  fractional factorial in four blocks of size  $2^2 = 4$ . To do this  $k - p - q = 6 - 2 - 2 = 2$  block defining contrasts must be chosen. If the generators for the fractional factorial are  $E = ABC$  and  $F = ABD$ , the defining relation for the fractional factorial is  $I = ABCE = ABDF = CDEF$ . If the block defining contrasts chosen are  $BDE$  and  $ACDE$ , they and their generalized interaction  $BDE(ACDE) = ABC$  will account for the three degrees of freedom for blocks. However, because of the fractionation we see that all of the following aliases also become confounded with blocks.

$$BDE = ACD = AEF = BCF$$

$$ACDE = BD = BCEF = AF$$

$$ABC = E = CDF = ABDEF$$

Again we can see that this choice of generators results in confounding the main effect  $E$  with blocks. This was an unintended result. A better

choice of block defining contrasts would be  $ACD$  and  $AB$ , which will not result in any main effects being confounded with blocks.

There are many choices for block defining contrasts and fractional factorial generators, and each will result in a different set of interactions being confounded with blocks and a different alias structure for the design. Some choices will be better than others and will result in fewer low-order interactions being confounded with blocks and main effects. However, finding the best generators and block defining contrasts for a particular design problem is not a simple task. Fortunately, statisticians have provided tables that show choices that are optimal in certain respects.

Box et al. (1978) provide tables for block defining contrasts that will result in a minimal number of low-order interactions being confounded with blocks in a blocked  $2^k$  design. Wu and Hamada (2000) provide a more extensive table that was generated by the algorithm described by Chen et al. (1993). In some cases, such as the  $2^6$  blocked into 16 blocks of size 4, Wu and Hamada's tables provide a set of blocks defining contrasts that are better (in the sense that fewer two-factor interactions are confounded with blocks) than the contrasts shown in Box, Hunter, and Hunter's tables.

It is more difficult to find fractional factorial generators and block defining contrasts for  $2^{k-p}$  fractional factorial designs, because of the combination of confounding due to the block effects and fractionization. However, Sun et al. (1997) provide an extensive catalog of block defining contrasts for  $2^k$  designs and generators for  $2^{k-p}$  designs along with the corresponding block defining contrasts that will result in best designs with regard to one of several quality criteria such as **estimability order**. They say that a design with *estimability of order e* is one in which all factorial effects of order  $e$  or less are estimable; that is they are not confounded with blocks or factorial effects of order less than  $e + 1$ . They state that there is no single best design, but the choice depends upon the situation.

When not specified by the user, the function `FrF2` in the R package `FrF2` uses the block defining contrasts from Sun et al.'s (1997) catalog

to create blocked  $2^k$  designs. For  $2^{k-p}$  `FrF2` uses the function `blockpick` or `blockpick.big`, which often finds designs isomorphic to those in Sun *et al.*'s (1997) catalog. For example, the code to create a  $2^{6-1}$  design, blocked into four blocks of eight experimental units each with using `FrF2` is shown on the next page.

```
Blocked.design <- FrF2(32, 6, blocks = 4,
                      alias.block.2fis = TRUE, randomize = FALSE)
summary(Blocked.design)
```

The summary report shows that the  $2^{6-1}$  design was created generator  $F = ABCDE$  and the block defining contrasts  $AB$  and  $AC$ . This is a resolution VI fractional factorial, and if this were run in a completely randomized design, all main effects and two-factor interactions would be estimable assuming four-factor and higher-order interactions were negligible. A design like this might be used if the experimenter was interested in estimating two-factor interactions. However, if experimental units are heterogeneous, more power or precision for detecting factorial effects and interactions could be achieved by sacrificing some interactions and confounding them with blocks. The block defining contrasts chosen by `FrF2` are both two-factor interactions. The generalized interaction  $AB(AC) = BC$  accounts for the third degree of freedom for blocks, and due to the fractionization each interaction confounded with blocks has an alias that is also confounded with blocks. In this case, the defining relation for the fractional factorial is  $I = ABCDEF$  and we see that  $AB = CDEF$ ,  $AC = BDEF$ , and  $BC = ADEF$ . So in reality three two-factor interactions and three four-factor interactions will be confounded with blocks and lost using this design.

By removing the option `alias.block.2fis=TRUE` as shown below, `FrF2` creates a resolution IV fractional factorial with defining relation  $I = ABCF$  and block defining contrasts  $ABD$  and  $ACE$ .

```
Blocked.design <- FrF2(32, 6, blocks = 4,
                      randomize = FALSE)
summary(Blocked.design)
```

Multiplying each block defining contrast and their generalized

interaction by the defining relation, it can be seen that  $ABD = DCF$ ,  $ACE = BEF$ , and  $BCDE = ADEF$  are confounded with blocks. Thus no two-factor interaction is confounded with blocks, but two-factor interactions are confounded with other two-factor interactions due to the fact that the design is resolution IV.

The first  $2^{6-1}$  design in four blocks presented in Wu and Hamada's table gives another alternative. It uses  $F = ABCDE$  as the generator of the half-fraction and  $ACD$ ,  $BCD$  as the block defining contrasts (Wu & Hamada, 2000). This plan only confounds one two-factor interaction with blocks and is therefore slightly better than the first design created by the `FrF2` function above. Having the fraction generator and the block defining contrasts available in Wu and Hamada's table, this design can be created in with `FrF2` by specifying the generators and blocks like the Injury Prevention Experiment shown in **Section 7.6.2**.

## 7.7. *Confounding 3-Level and p-Level Factorial Designs*

If every factor in a factorial has three levels we call the design a symmetric  $3^k$  design. These designs can be run in blocks (CCBF) using the same strategy as shown in **Section 0** by confounding portions of interactions with blocks. Confounded  $3^k$  designs can only be run in blocks of size  $3^q$  where  $q < k$ . To illustrate the method of confounding, consider a  $3^2$  design with factors  $A$  and  $B$ , each with three levels. Since  $k = 2$ , this design can only be blocked in blocks of size three, resulting in three blocks. In order to prevent main effect  $A$  and  $B$  from being completely confounded with blocks, the two-factor interaction  $AB$  should be confounded with blocks. However, there are  $(3 - 1) \times (3 - 1) = 4$  degrees of freedom for the two-factor interaction while there are only  $3 - 1 = 2$  degrees of freedom for blocks. Therefore, only part of the interaction need be confounded with blocks. The four degrees of freedom for  $AB$  can be partitioned into two degrees of freedom contrasts  $A + B$  and  $A + 2B$ , and one of these can be used to define the blocks.

If the levels of the factors  $A$  and  $B$  are represented symbolically by 0, 1 and 2, the levels of the two contrasts  $A + B$  modulo 3 and  $A + 2B$

modulo 3, are shown below. The levels of these two contrasts can each be seen to have two degrees of freedom because they divide the treatment combinations into three groups similar to the way the levels of main effects  $A$  and  $B$  divide the treatment combinations into three groups. Both of these contrasts are orthogonal to main effects  $A$  and  $B$  since all three levels of each main effect are represented within each level of both contrasts. Therefore if either of these contrasts is used to define blocks neither main effect will be confounded with blocks. The R code below shows how a design created with the `gen.factorial` function can be blocked using the `mod` function in `daewr` confounding blocks with the  $A + B$  contrast.

A	B	A+B	A+2B
0	0	0	0
0	1	1	2
0	2	2	1
1	0	1	1
1	1	2	0
1	2	0	2
2	0	2	2
2	1	0	1
2	2	1	0

```
library(AlgDesign)
Blockdes <- gen.factorial(3, nVars = 2,
                           center = FALSE, varNames = c("A", "B"))
Block <- 1 + mod((Blockdes$A - 1) + (Blockdes$B - 1), 3)
Blockdes <- cbind(Block, Blockdes)
```

The model, represented in the R `aov` notation for this design, would be `Block+A+B`. The  $AB$  interaction, that is confounded with blocks, is left out of the model just as it would have been in the confounded  $2^k$  and  $2^{k-p}$  designs. Main effect  $A$  would have two degrees of freedom, main effect  $B$  would have two degrees of freedom, blocks would have two degrees of freedom, and the error would have two degrees of freedom. For this design to be useful the interaction  $AB$  would be assumed negligible.

In general when confounding  $3^k$  experiments in blocks of size  $3^q$ ,  $k - p$  block defining contrasts must be chosen. They and all of their generalized interactions will also be confounded with blocks. It would be difficult to choose a set of blocks defining contrasts by trial and error that will result in the fewest low-order interactions being confounded with blocks. Wu and Hamada (2000) give tables of block defining contrasts and design generators for  $3^k$  and  $3^{k-p}$  designs blocked into blocks of size  $3^q$  that will result in the maximum number of estimable effects. These tables were determined using the algorithm described by Chen et al. (1993). The designs listed in their tables can be created using the `mod` functions as shown in the example above.

Cook and Nachtsheim (1989) developed a different algorithm to block an existing design or create a blocked design from a list of candidates by maximizing the block D-efficiency,  $D_S$ . The `optBlock` function in the R package `AlgDesign` uses a different algorithm but the same fundamental idea as Cook and Nachtsheim (1989) to find blocked designs. For example, the code below creates a  $3^4$  design in nine blocks of nine.

```
library(AlgDesign)
Blockdes <- gen.factorial(3, nVars = 2,
                           center = FALSE, varNames = c( "A", "B" ))
Block <- 1 + mod((Blockdes$A - 1) +
                  (Blockdes$B - 1), 3)
Blockdes <- cbind(Block, Blockdes)
library(AlgDesign)
Blockdes <- gen.factorial(3, nVars = 4,
                           factors = "all", varNames = c("A", "B", "C", "D"))
Blockdes <- optBlock( ~ A + B + C + D + A:B + A:C +
                      A:D+B:C + B:D + C:D, withinData = Blockdes,
                      blocksizes = c(rep(9, 9)), criterion = "D")
```

The model statement ( $A + B + C + D + A:B + A:C + A:D + B:C + B:D + C:D$ ) defines the terms the experimenter would like to estimate. In this example all main effects and two-factor interactions should be estimable. With this specification three and four-factor interactions can be completely or partially confounded with the 9–1 degrees of freedom for blocks.

$p^k$  designs where  $p$  is a prime number can also be blocked using the mod function as shown above, but in practice,  $3^k$  or  $3^{k-p}$  designs and  $p^k$  or  $p^{k-s}$  designs are rarely used because it is unusual for all factors in a factorial design to have the same number of levels, unless the number of levels is two.  $2^k$  and  $2^{k-p}$  designs are commonly used in practice because experimenters reduce the number of levels of each factor to two (by choosing the two levels they feel will exhibit the maximum difference) in order to reduce the total number of experiments. In cases where the number of levels of some factors cannot be reduced to two, because they represent a discrete set of alternatives such as machine settings, a mixed level, or asymmetric factorial results. The principles shown in this section, to block  $3^k$  or  $p^k$  experiments by confounding portions of interactions, can be extended to the mixed level factorials as shown in the next section.

## 7.8. Blocking Mixed Level Factorials and OAs

A mixed level or asymmetric factorial can be represented as  $s_1^{m_1} \times s_2^{m_2} \times \cdots \times s_\gamma^{m_\gamma}$  involving  $n = \sum_{i=1}^\gamma m_i$  factors where  $m_i$  factors each has  $s_i$  levels. For example, a  $2^3 \times 3^2 \times 4^1 \times 6^1$  is a factorial with three factors with two levels, two factors with three levels, one factor with four levels, and one factor with six levels. The number of levels of every factor in a mixed level factorial is either a prime number or a product of prime numbers. If the number of the  $i$ th factor  $s_i = \prod_{l=1}^m p_l$  where  $p_l$  are prime numbers, then the levels of the factor  $s_i$  can be represented by the combination of levels of  $m$  pseudo factors each with a prime number of levels. For example, in the  $2^3 \times 3^2 \times 4^1 \times 6^1$  factorial the first three factors ( $A$ ,  $B$ , and  $C$ ) have two levels, where two is a prime number. The fourth and fifth factors ( $D$  and  $E$ ) have three levels, where three is a prime number. The sixth factor ( $F$ ) has four levels, and  $4 = 2 \times 2$  is the product of two prime numbers. Finally, the seventh factor ( $G$ ) has six levels, and  $6 = 2 \times 3$  is also a product of prime numbers. The levels of the four- and six-level factors  $F$  and  $G$  can be represented as combinations of the levels of two- and three-level pseudo factors  $f_1, f_2, g_1$ , and  $g_2$  as shown below.

F	$f_1$	$f_2$	G	$g_1$	$g_2$
0	0	0	0	0	0
1	1	0	1	1	1
2	0	1	2	0	2
3	1	1	3	1	0
			4	0	1
			5	1	2

### 7.8.1. Blocking Mixed Level Factorials

Since the levels of each factor in a mixed level factorial can be represented by a combination of the levels of pseudo factors with a prime number of levels, the entire factorial can be represented by a  $p_1^{n_1} \times p_2^{n_2} \times \cdots \times p_g^{n_g}$  factorial, where all the fact have a prime number of levels. For example,  $2^3 \times 3^2 \times 4^1 \times 6^1$  factorial can be represented a  $2^6 \times 3^3$  factorial. The  $p_i^{n_i}$  are called sub-experiments, and the classical method of confounding a mixed level or asymmetric factorial is to apply the method described in the last section to each sub-experiment and then combine the blocks from each sub-experiment to form the blocks of the entire **completely confounded blocked factorial (CCBF)**. This method will result in the greatest efficiency for estimating the effects and interactions that are not confounded with blocks in that they will be completely orthogonal to blocks.

Asymmetric factorials of the form  $s_1^{m_1} \times s_2^{m_2} \times \cdots \times s_\gamma^{m_\gamma}$  can be blocked into  $b$  blocks of size  $k$ . However, to avoid confounding or partially confounding any main effect, the block size  $k$  must be divisible by all the prime numbers  $p_1 - p_r$  that represent the number of levels of the factors or number of levels of the pseudo factors used to represent a factor.

To illustrate the classical method of blocking a mixed level factorial, consider blocking the 36 treatment combinations of a  $2^2 \times 3^2$  factorial. This factorial can be represented as a product of two sub-experiments: a  $2^2$  comprised of two factors  $A$  and  $B$  each at two levels; and a  $3^2$  comprised of two factors  $C$  and  $D$  each at three levels. Since the levels of all factors are prime, the block size must be divisible by both two and three. The blocks can be found by confounding within each sub-experiment. Since there are only two factors in each sub-experiment, and it would be undesirable to confound main effects with blocks, there

is only one interaction that can be confounded in each sub-experiment.

Confounding  $A + B$  in the  $2^2$  sub-experiment results in two blocks of size 2. Confounding the  $C + D$  contrast in the  $3^2$  sub-experiment results in three blocks of size 3. The combination of each block from the first sub-experiment with each block of the second sub-experiment results in blocks of size  $k = 6$ , which is divisible by both 2 and 3. The interactions confounded in the complete factorial will be  $AB$ , two degrees of freedom from  $CD$  and two degrees of freedom from the product  $ABCD$ . The main effects  $A, B, C, D$  and the  $AC, AD, BC, BD$  two-factor interactions and the  $ABC, ABD, ACD$ , and  $BCD$  three-factor interactions are not confounded with blocks and are estimable.

The code below shows how to create this design in R. First, the `fac.design` function from the `DoE.base` package is used to create the  $2^2 \times 3^2$  comprised of all possible combinations of the two sub-experiments.

```
Mixfac <- fac.design(nlevels = c(2, 2, 3, 3),
                      factor.names = c("A", "B", "C", "D"),
                      randomize = FALSE)
```

creating full factorial with 36 runs ...

Next, `mod` function from `daewr` package is used to create the block indicators.

```
library(daewr)
blk1 <- mod(as.numeric(Mixfac$A) +
             as.numeric(Mixfac$B), 2) + 1
blk2 <- mod(as.numeric(Mixfac$C) +
             as.numeric(Mixfac$D), 3) + 1
Block <- as.factor((blk1 - 1) * 3 + blk2 )
```

The columns in the design created by `fac.design` are factor objects in R, and they must be converted to numeric objects using the R function `as.numeric` in order to use them as arguments in the `mod` function to create the block indicators `blk1` and `blk2` in the two sub-experiments. The block indicator (`blk1`) for the  $2^2$  sub-experiment takes the values 1 and 2. The block indicator (`blk2`) in the  $3^2$  sub-experiment takes the values 1, 2, and 3. The statement `Block <- as.factor((blk1 - 1) * 3 +`

`blk2`) combines the two block indicators into one. When `blk1=1` and `blk2=1, Block=1`; when `blk1=1` and `blk2=1, Block==2`, and so forth.

Finally, the block indicators are combined with the  $2^2 \times 3^2$  design and the runs are sorted by blocks.

```
BlMixfac <- cbind(Block,Mixfac)
BlMixfac <- BlMixfac[order(BlMixfac$Block), ]
```

The first four blocks of the design are shown horizontally below.

```
BlMixfac
```

Block	A	B	C	D	Block	A	B	C	D	Block	A	B	C	D	Block	A	B	C	D
1	2	1	2	1	2	1	2	3	1	3	1	1	1	1	4	1	2	2	1
1	2	2	1	2	2	1	1	2	2	3	2	1	3	2	4	1	1	1	2
1	1	1	3	3	2	1	2	1	3	3	2	2	2	3	4	2	2	2	1
1	1	1	2	1	2	2	1	3	1	3	2	2	1	1	4	2	1	1	2
1	1	2	1	2	2	2	2	2	2	3	1	1	3	2	4	2	1	1	2
1	2	2	3	3	2	2	1	1	3	3	1	2	2	3	4	1	2	3	3

This design will allow estimation of all the terms in the model  $A + B + C + D + A:B + A:D + B:C + B:D$ , but the interactions  $A:B$  and  $C:D$  are confounded with blocks and cannot be estimated.

When the levels of one or more factors in a mixed level factorial are a product of prime powers and can be represented by the combination of levels of pseudo factors, no interactions among pseudo factors that represent the same factor can be confounded in any sub-experiment. If any interaction among pseudo factors that represents a factor is confounded, then that main effect will also be confounded. For example, consider blocking the 72 combinations of factor levels in a  $3 \times 4 \times 6$  factorial. Factor  $A$  has three levels, factor  $B$  has four levels, and can be represented by all combinations of two two-level pseudo factors  $b_1$  and  $b_2$ , and factor  $C$  has six levels that can be represented by all combinations of a two-level pseudo factor  $c_1$  and a three-level pseudo factor  $c_2$ . Using the pseudo factors, the  $3 \times 4 \times 6$  factorial can be represented by a  $2^3 \times 3^2$  factorial in prime level factors and prime level pseudo factors. The block size must be divisible by the prime numbers 2 12 may be possible.

The first sub-experiment is a  $2^3$  composed of two-level pseudo factors  $b_1, b_2, c_1$  and the second sub-experiment is a  $3^2$  composed of factor  $A$  and pseudo factor  $c_2$ . The first sub-experiment can only be blocked into 2 blocks of 4 in order to avoid confounding the  $b_1 + b_2$  interaction and therefore the  $B$  main effect. Thus the three-factor interaction  $b_1 + b_2 + c_1$  must be confounded with blocks in the first sub-experiment. In the second sub-experiment the  $A + c_2$  contrast of the  $AC$  interaction must be confounded to create three blocks of three. The combination of each block from the first sub-experiment combined with each block from the second sub-experiment results in six blocks of 12 treatment combinations, and this is the only block size that is possible without confounding a main effect.

The interactions confounded with the five degrees of freedom for blocks in the combined factorial will be  $BC$  (with one degree of freedom) from the first sub-experiment, two degrees of freedom from  $AC$  from the second sub-experiment, and two degrees of freedom from the product  $ABC$  (i.e.,  $b_1 + b_2 + c_1 \times (A + c_2)$ ). The R code below illustrates how this design can be created using `fac.design`. First, the `fac.design` function is used to create the full factorial in factors and pseudo factors.

```
> library(DoE.base)
> Mixfac <- fac.design(nlevels = c(2, 2, 2, 3, 3),
  factor.names = c("b1", "b2", "c1", "A", "c2"),
  randomize = FALSE)
```

**creating full factorial with 72 runs ...**

Next, the pseudo factor interaction  $b_1 \times b_2 \times c_1$  is confounded with the block indicator `blk1` in the  $2^3$  sub-experiment, and the  $A \times c_2$  interaction is confounded with the block indicator `blk2` in the  $3^2$  sub-experiment.

```
> library(daewr)
> blk1 <- mod(as.numeric(Mixfac$b1) +
  as.numeric(Mixfac$b2) +
  as.numeric(Mixfac$c1), 2) + 1
blk2<-mod(as.numeric(Mixfac$A) +
  s.numeric(Mixfac$c2), 3) + 1
```

Finally, the block indicators are combined to create the block factor, the pseudo factors are combined to create the factor levels, and all are combined and sorted by the block numbers as shown below.

```
Block <- as.factor((blk1 - 1) * 3 + blk2 )
B <- (as.numeric(Mixfac$b1) - 1) * 2 +
      as.numeric(Mixfac$b2)
C <- (as.numeric(Mixfac$c1) - 1) * 3 +
      as.numeric(Mixfac$c2)
BlMixfac<-cbind(Block, A = Mixfac$A,
                  B = as.factor(B), as.factor(C))
BlMixfac <- BlMixfac[order(Block), ]
```

The first block of the design is shown on the next page.

### BlMixfac

	Block	A	B	C
[1,]	1	2	3	1
[2,]	1	2	2	1
[3,]	1	2	1	4
[4,]	1	2	4	4
[5,]	1	1	3	2
[6,]	1	1	2	2
[7,]	1	1	1	5
[8,]	1	1	4	5
[9,]	1	3	3	3
[10,]	1	3	2	3
[11,]	1	3	1	6
[12,]	1	3	4	6

The model that can be fit to the data resulting from this experiment is  $A + B + C + A:B$  since none of the terms in this model are confounded with blocks.

Even though this design is optimal for estimating the parameters in the model, it may not be the best in all situations since the two-factor interactions  $AC$  and  $BC$  cannot be estimated. Since there are only five degrees of freedom for blocks, it would appear that there should be a way to confound part of the  $2 \times 3 \times 5 = 30$  degrees of freedom for the three-factor interaction  $ABC$  with blocks and leave all the two-factor interactions estimable. In fact, if you are willing to sacrifice some of the efficiency in estimating the terms in the model, a better blocked design can be found using a  $D$ -optimal search than can be found using the classical method.

Before the days of modern computers and software packages like R, it was necessary to completely confound some interactions with blocks, using the classical method, so that other interactions would be left orthogonal to blocks. In that way, the analysis of the data could be completed by hand using the ANOVA sum of squares formulas for balanced data. However, with availability of programs like the R [lm](#) function, sums of squares are computed using matrix operations as shown in Chapters 2 and 3, and it is no longer necessary for each term in the model to be completely orthogonal to blocks in order to compute the ANOVA and  $F$ -tests.

The model for a blocked factorial experiment can be written in matrix notation as

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\tau} + \mathbf{Z}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \quad \text{Eq. 7-5}$$

where  $\mathbf{y}$  is the  $n \times 1$  vector of responses,  $\boldsymbol{\tau}$  is the vector of estimable treatment effects and interactions, and  $\boldsymbol{\beta}$  is the vector of block effects. One optimality criterion that has been proposed for blocked designs is the  **$D_s$  criteria** (see Atkinson et al., (2007)). A  **$D_s$  optimal design** is one that minimizes the covariance matrix of the least squares estimator for  $\boldsymbol{\tau}$  or equivalently maximizes the determinant of

$$\mathbf{X}'\mathbf{Q}\mathbf{X} \quad \text{Eq. 7-6}$$

where

$$\mathbf{Q} = \mathbf{I} - \mathbf{Z}(\mathbf{Z}'\mathbf{Z})^{(-1)}\mathbf{Z}' \quad \text{Eq. 7-7}$$

Designs where blocks are orthogonal to treatment effects, or  $\mathbf{X}'\mathbf{Z} = \mathbf{0}$ , are 100%  $D_s$ -efficient.

Applying the classical method separately to symmetric sub-experiments results in designs that have known confounding patterns and are 100%  **$D_s$ -efficient** for estimating the effects and interactions,  $\boldsymbol{\tau}$ , that are not confounded with the block differences (since they will be orthogonal to blocks). However, in practical situations, use of the classical approach does not provide much flexibility in the choice of block size or in the choice of interactions to be confounded with block differences. Since the sub-experiments are often defined in terms of pseudo factors, interactions of interest often become confounded with blocks.

Cook and Nachtsheim's (1989) more general computer algorithm for creating blocked designs and the similar algorithm available in the `optBlock` function of the `AlgDesign` package can find better blocked designs for mixed level factorials. These algorithms begin with a nonsingular starting design, then sequentially exchange treatment combinations assigned to one block with those assigned to other blocks in order to maximize  $|X'QX|$ . The designs resulting from this algorithm may not be 100% Ds-efficient for estimating  $\tau$ , but greater choices of block sizes and estimable interactions are possible.

The next example illustrates the use of the `optBlock` function. First the `gen.factorial` function in the `AlgDesign` package creates a candidate set composed of a  $3 \times 4 \times 6$  factorial. The model statement in the `optBlock` function call specifies that all two-factor interactions be estimable. The `blocksizes=c(rep(12,6))` option requests six blocks of size 12, and the `criterion="D"` requests that `optBlock` find a design that is optimal in the sense that it maximizes the determinant. The six blocks of the design produced here are stored in `bdes$Blocks` and are shown side by side on the next page.

```
des <- gen.factorial(levels = c(3, 4, 6), factors =
  'all', varNames = c("A", "B", "C"))
bdes <- optBlock(~ A + B + C + A:B + A:C + B:C,
  withinData = des, blocksizes = c(rep(12, 6)),
  criterion = "D")
bdes$Blocks
```

\$B1			\$B2			\$B3					
	A	B	C	A	B	C	A	B	C		
2	2	1	1	15	3	1	2	6	3	2	1
4	1	2	1	20	2	3	2	7	1	3	1
9	3	3	1	22	1	4	2	14	2	1	2
13	1	1	2	28	1	2	3	37	1	1	4
17	2	2	2	33	3	3	3	41	2	2	4
24	3	4	2	35	2	4	3	48	3	4	4
27	3	1	3	38	2	1	4	52	1	2	5
32	2	3	3	42	3	2	4	57	3	3	5
34	1	4	3	43	1	3	4	59	2	4	5
66	3	2	6	61	1	1	6	63	3	1	6
67	1	3	6	65	2	2	6	68	2	3	6

71	2	4	6		72	3	4	6		70	1	4	6
\$B4					\$B5					\$B6			
	A	B	C			A	B	C			A	B	C
11	2	4	1		1	1	1	1		3	3	1	1
25	1	1	3		8	2	3	1		5	2	2	1
29	2	2	3		12	3	4	1		10	1	4	1
36	3	4	3		18	3	2	2		16	1	2	2
39	3	1	4		19	1	3	2		21	3	3	2
44	2	3	4		40	1	2	4		23	2	4	2
46	1	4	4		45	3	3	4		26	2	1	3
50	2	1	5		47	2	4	4		30	3	2	3
54	3	2	5		51	3	1	5		31	1	3	3
55	1	3	5		53	2	2	5		49	1	1	5
64	1	2	6		58	1	4	5		56	2	3	5
69	3	3	6		62	2	1	6		60	3	4	5

`optBlock` does not produce a unique design, therefore running this code again may result in a different design with similar properties. The six blocks of the design produced here are stored in `bdes$Blocks` and are shown side by side on the next page.

The design can be stored in a form convenient for analysis using the following code.

```
Block <- c(rep(1:6, each = 12))
bdesign <- cbind(Block, bdes$design)
```

Although not 100% efficient (implying all terms in the model are completely orthogonal to blocks), like the design found using the classical method, at least all of the two-factor interactions are estimable. If the code is modified by changing the option on the blocks statement from `blocksizes=c(rep(12,6))` to `blocksizes=c(rep(6,12))`, the `optBlock` function finds a  $3 \times 4 \times 6$  factorial blocked in 12 blocks of six that still allows estimation of all two-factor interactions. The  $D$ -efficiency for the treatments in this design was low, but this is far better than could be accomplished with the classical method. The only way the classical method could block the  $3 \times 4 \times 6$  factorial into blocks of size 6 would be to confound part of the main effect for factor  $B$ . If reducing the block size to six reduces the variance of the experimental units within a block, the sacrifice in efficiency will be worthwhile.

When using the  $D_s$ -optimal approach to finding a confounded block design for a mixed level factorial, it is no longer required that the block size be divisible by all the prime numbers that represent factor levels in the sub-experiments. If the block size that is most convenient for reducing variability of experimental units within a block is not divisible by the number of levels of all factors, it may prevent finding a design where all factors and interactions of interest are orthogonal to blocks. However, a design found using the  $D_s$ -optimal search will usually not have all effects in the model orthogonal to blocks anyway. As long as the determinant is not zero, the terms specified in the model will be estimable. Due to the non-orthogonality of these designs, the type III sums of squares and least squares means should be used when analyzing data.

### 7.8.2. Blocking Orthogonal Arrays, Orthogonal Main Effect Plans, and Nearly Orthogonal Main Effect Plans

**Orthogonal array designs**, orthogonal main effect plans, and nearly orthogonal plans discussed in [Section 1.8](#) can also be blocked into CCBFF designs in order to reduce the variation of experimental units within blocks or to allow the list of experiments to be completed on different days or with different batches of raw materials, and so on. One way to accomplish this is to include the block factor as one of the factors when initially creating a design using the `oa.design` function in the `DoE.base` package. The code below, that is similar to what was shown in [Section 6.7](#), searches for an appropriate orthogonal array for creating a fraction of  $41 \times 61 \times 23 \times 32$  design blocked into three blocks.

```
library("DoE.base")
show.oas(factors = list(nlevels = c(4, 6, 2, 3),
                        number = c(1, 1, 3, 3)))
```

The results show there are several 72-run orthogonal arrays for this purpose. Since they are orthogonal, all main effects will be orthogonal to each other and the block factor. The code below shows how to create the 72-run  $41 \times 61 \times 23 \times 32$  design in three blocks of 24.

```

library("DoE.base")
fnames =c ("A", "B", "C", "D", "E", "F", "G", "Block")
BlockOA <- oa.design(nlevels =
  c(4, 6, 2, 2, 3, 3, 3), factor.names = fnames,
  seed=104, nruns = 72)
BlockOA <- BlockOA[order(BlockOA$Block), ]

```

Since there are only 18 degrees of freedom needed for estimating main effects and the block factor, this design leaves 56 degrees of freedom for error. If the experimenter were willing to sacrifice orthogonality, a much smaller blocked design can be found. For example, the code below creates a 72-run orthogonal array as a list of candidates and then uses the `optBlock` function to create a 24-run subset blocked into three blocks of 8.

```

library(DoE.base)
library(AlgDesign)
fnames <- c("A", "B", "C", "D", "E", "F", "G")
cand <- oa.design(nlevels = c(4, 6, 2, 2, 2, 3, 3),
  factor.names = fnames, randomize = TRUE,
  seed = 104, nruns = 72)
bdes <- optBlock(~ A + B + C + D + E + F + G,
  withinData = cand, blocksizes = c(rep(8, 3)),
  criterion = "D")

```

The first block is shown in part of the output below.

<code>bdes\$Blocks</code>
---------------------------

\$B1
A B C D E F G
4 2 1 2 1 1 3 2
5 1 6 2 2 1 3 3
6 1 3 1 2 2 2 2
28 3 2 2 1 2 2 3
40 3 6 1 1 1 2 1
45 3 5 1 1 2 1 2
50 4 1 1 1 1 2 3
70 3 3 2 2 1 1 3

## 7.9. Partially Confounded Blocked Factorial (PCBF)

One of the main purposes for running a factorial experiment with few

factors is to estimate all interactions. However, if a design is confounded into blocks, we will lose the ability to estimate some interactions. Yet, including a few additional blocks in the design will allow us to estimate all main effects and interactions using the method of **partial confounding**. This method consists of confounding one or more effects or interactions in one set of blocks, and confounding different effects or interactions in additional sets of blocks (or replicates). By combining all the replicates, all effects and interactions will be estimable, although not orthogonal to blocks.

For example, if we were to perform a study on the effect of two levels of  $A$ =calcium supplements and two levels of  $B$ =potassium supplements upon the blood pressure of hypertensive subjects, a 2<sup>2</sup> factorial experiment would be performed in order to estimate the two main effects and the interaction. However, if the experiments were blocked into two blocks of two experimental units (e.g., two pairs of identical twins) by confounding  $AB$ , this interaction would be lost. One way to remedy the problem would be to include four additional blocks, confounding main effect  $A$  with the difference in two additional blocks, and main effect  $B$  with the difference in two more blocks. Combining the six blocks, both main effects and their interaction would be estimable. This design is shown in **Table 7-12**, and this general class of designs is called **partially confounded blocked factorials** or **PCBF**.

We could then fit the model **Block + A + B + A:B** to data from the combined set of blocks. We would use the type III sums of squares and least squares means to analyze the data, since the effects are not completely orthogonal to blocks.

If we allow each effect and interaction in the model to be confounded an equal number of times in different sets of blocks, as in the example shown above, Butler (2006) shows the design will have favorable properties and the maximum treatment  $D$ -efficiency for estimating the factorial effects. Therefore, we could create a design like this using Cook and Nachtsheim's algorithm or the similar algorithm available in the **optBlock** function of the **AlgDesign** package. The example below shows how this could be done.

**Table 7-12. Partially Confounded  $2^2$  in 6 Blocks of 2**

Block	A	B	
1	-	-	Rep 1 confounds
1	+	+	
2	-	+	AB
2	+	-	
3	-	-	Rep 2 confounds
3	-	+	
4	+	-	A
4	+	+	
5	-	-	Rep 3 confounds
5	+	-	
6	-	+	B
6	+	+	

```
library(AlgDesign)
Blockdes <- gen.factorial(2, nVars = 2, factors = "all",
                           varNames = c("A", "B"))
Blockdes <- optBlock(~ A + B + A:B, withinData =
  Blockdes, blocksizes = c(rep(2,6)), criterion = "D")
```

For mixed level factorial plans, Das (1960) has provided a method for constructing balanced confounded designs where (1) the information recovered for each degree of freedom for any partially confounded interaction is the same, and (2) any contrast of a partially confounded interaction is estimable independently of any contrast of another partially confounded interaction. The information recovered for the  $i$ th degree of freedom in the **Eq. 7-5** is calculated as  $\frac{c_{ii}}{c'_{ii}}$ .  $c_{ii}$  is the diagonal of  $(\mathbf{X}'\mathbf{X})^{-1}$  matrix corresponding to a particular single degree of freedom, and  $\sigma^2 c'_{ii}$  is the variance of  $\hat{\tau}^i$  in a design where the treatment effects are orthogonal to blocks.  $c'_{ii}$  is the diagonal  $(\mathbf{X}'\mathbf{Q}\mathbf{X})^{-1}$ , and  $\sigma'^2 c'_{ii}$  is the variance of  $\hat{\tau}^i$  in the partially confounded design where  $Q$  is defined in **Eq. 7-7**. In partially confounded designs  $c'_{ii} > c_{(ii)}$ , but  $\sigma'^2$  should be much smaller than  $\sigma^2$  due to the fact that the experimental units are more homogeneous within the blocks of reduced size.

Constructing a design using Das's method consists of converting the

asymmetrical factorial into a fraction of a symmetrical factorial. The partial confounding is performed in replicates of the symmetrical factorial, and then each replicate is reduced by fractionation. Lawson et al. (2009) show how Das's method can be implemented in SAS with proc plan and data step programming. They also provide a SAS macro for generating Das's balanced confounded designs. Creating designs by this method will allow all interactions in the model to be estimated and will result in a design with equal precision for each single degree of freedom of partially confounded effects and interactions. One disadvantage for using this method is that the total number of treatment combinations,  $N = s_1^{m_1} \times s_2^{m_2} \times \dots \times s_\gamma^{m_\gamma}$ , must always be divisible by the block size. This can require a large number of blocks in some cases.

The `optBlock` function can also be used to find a partially confounded mixed level factorial that will allow estimation of all interactions, and there is no restriction on the block size. Using this method can sometimes produce a balanced confounded design like Das's method, and in other cases it will find an approximately balanced design with more choices for the block size and total number of runs. Lawson et al. (2009) compare the properties of designs created by this method to designs created by Das's method. The example below shows the use of the `AlgDesign` functions `gen.factorial` and `optBlock` to construct a partially confounded  $3 \times 2^2$  factorial blocked in 12 blocks of 4.

```
desf <- gen.factorial(c(2, 2, 3), factors = "all",
                      varNames = c("A", "B", "C"))
Blockdes <- optBlock(~ A*B*C, withinData = desf,
                     blocksizes=c(rep(4, 12)), criterion = "D")
```

As an example of the design and analysis of a partially confounded factorial, consider an experiment performed by Dossett et al. (2007). They were investigating methods of storing apple slices in brown bag lunches. The problem was that the apple slices in a sack lunch turn brown and look unappetizing before lunchtime. They wanted to compare the effects of dipping the slices in different treatment solutions prior to storage and to compare the effects of storing them in different storage containers to see if they could find conditions to reduce the amount of browning. **Table 7-13** shows the factors and levels. They

thought that different varieties of apples might brown at different rates and therefore wanted to block by apple variety. Their apple slicer cut the apples into six slices, therefore all  $4 \times 3 = 12$  treatment combinations could not be tested within each block or apple. Since they were interested in the possible interaction of their treatment factors, they ran a partially confounded design.

*Table 7-13. Levels of Factors for Apple Slice Browning Experiment*

Factor		
Factor Level	A=Pretreatment Solution	B=Storage Container
0	none	none—open air
1	weak lemon juice	Ziploc bag
2	salt water	Tupperware
3	baking soda water	—

The R code below can be used to create a  $4 \times 3$  factorial design in 4 blocks of size 6 using `gen.factorial` and `optBlock`. The blocks represented four different varieties of apples: namely Fuji, Braeburn, Red Delicious, and Granny Smith. The experimental units would be the six slices from each apple, and these could be randomly assigned to one of the treatment combinations designated for each block.

```
des23 <- gen.factorial(c(4, 3), factors = "all",
                        varNames = c("A", "B"))
Blockdes <- optBlock(~ A*B, withinData = des23,
                      blocksizes = c(rep(6, 4)), criterion = "D")
```

Dossett et al. (2007) actually used a slightly different procedure to generate the design, but their design had similar properties. After storing their treated apple slices for an hour and forty-five minutes, each slice was compared to photographs of an apple slice at various stages of browning and assigned a rating between 1 and 11. The lowest rating was for the least amount of browning and the highest was for the most. All three team members independently rated the slices, and the response was the average rating.

Table 7-14 shows the results.

**Table 7-14. Blocked  $4 \times 3$  Factorial Design and Results for Apple Slice Browning**

Block 1			Block 2			Block 3			Block 4		
A	B	rating									
0	0	7.33	2	0	1.00	3	2	10.33	2	2	1.00
2	1	1.67	1	0	3.33	1	0	2.00	3	0	8.33
0	2	6.67	2	2	1.00	2	1	2.33	1	1	4.33
1	1	1.33	0	1	8.67	0	2	7.00	1	2	1.33
2	0	1.67	0	0	8.33	3	1	3.67	0	1	3.33
3	0	8.00	3	2	4.00	1	2	6.00	3	1	9.33

This design and the results are stored in the data frame `apple` in the `daewr` package. The example below shows how to access and analyze this data.

```
library(car)
modf <- lm(rating ~ Block + A + B + A:B, data = apple,
            contrasts = list(A = contr.sum, B = contr.sum,
            Block = contr.sum))
```

The Anova table is shown below.

```
Anova(modf, type = "III")
```

```
Anova Table (Type III tests)

Response: rating
          Sum Sq Df F value    Pr(>F)
(Intercept) 522.48  1 72.7428 1.323e-05 ***
Block        3.01   3  0.1396   0.9338
A           145.96  3  6.7740   0.0110 *
B           2.21   2  0.1535   0.8599
A:B         7.73   6  0.1795   0.9755
Residuals   64.64  9
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can see that the only thing significant was factor *A*, the treatment solution. The least squares means for the different treatment solutions can be found as shown below.

```
library(lsmeans)
lsmeans(modf, pairwise ~ A, adjust = ("tukey"))
```

```
$lsmeans
A lsmean    SE df lower.CL upper.CL
0   6.97 1.12  9    4.445     9.50
1   2.97 1.12  9    0.444     5.50
2   1.53 1.12  9   -0.998     4.05
3   7.19 1.12  9    4.668     9.72
```

Results are averaged over the levels of: Block, B  
 Confidence level used: 0.95

```
$contrasts
contrast estimate    SE df t.ratio p.value
A0 - A1      4.001 1.61  9    2.484  0.1295
A0 - A2      5.443 1.55  9    3.518  0.0276
A0 - A3     -0.222 1.61  9   -0.138  0.9990
A1 - A2      1.442 1.61  9    0.896  0.8074
A1 - A3     -4.223 1.55  9   -2.729  0.0901
A2 - A3     -5.666 1.61  9   -3.518  0.0276
```

Results are averaged over the levels of: Block, B P value  
 adjustment: tukey method for comparing a family of 4 estimates

Using the student Tukey's HSD procedure as described in **Section 2.8.2**, we find that dipping the apple slices in salt water reduces browning the most, but the amount of browning for slices dipped in lemon juice was not significantly worse. The results are summarized by the underlines in **Table 7-15**. The experimenters recommended further studies varying the concentration of lemon juice to see if they could improve the results and eliminate the aftertaste left by salt water.

**Table 7-15. Least Squares Means for Factor A**

(means underlined by the same line are not significantly different)

Salt Water	Lemon Juice	None	Baking Soda
1.53	2.97	6.97	7.19

## 7.10. Review of Important Concepts

When experimental units are heterogeneous and can be grouped into smaller blocks of more homogeneous experimental units, a blocked design should be used. When the number of experimental units per block or block size is smaller than the number of levels of the treatment factor or combinations of levels of treatment factors in a factorial design,

an incomplete block design should be used.

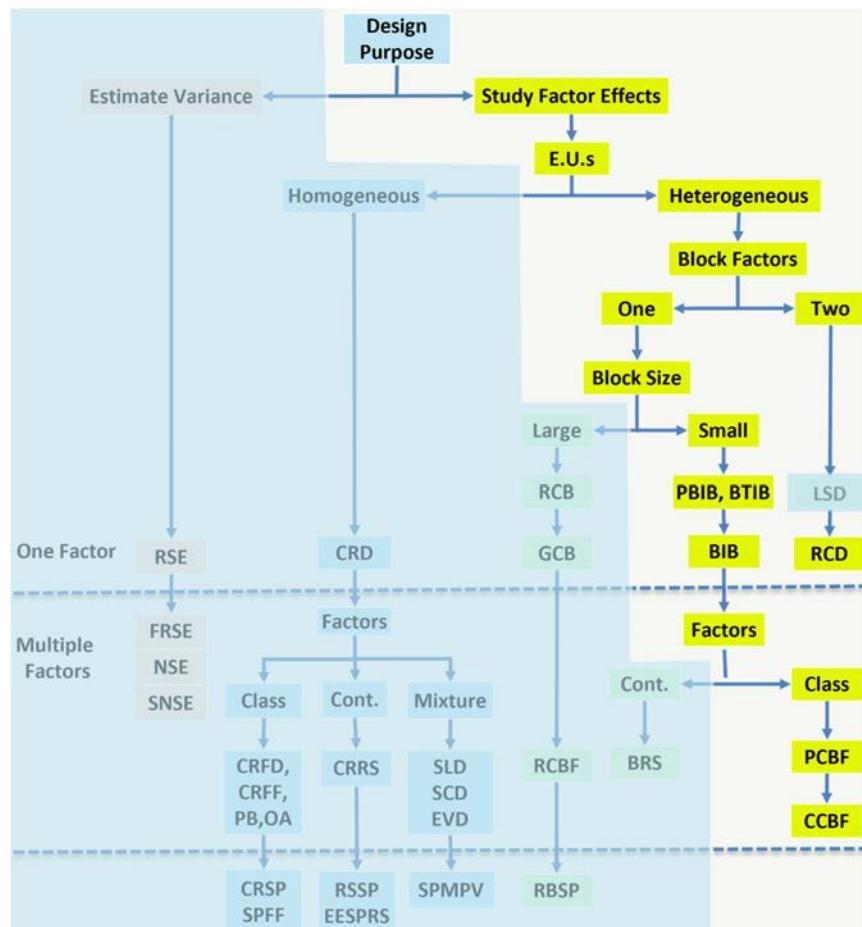
When there is only one treatment factor, there is a choice between two types of incomplete block designs. **Balanced incomplete block (BIB)** designs require that every treatment level occurs in a block an equal number of times with every other treatment level. BIB designs can be created with the

BIB designs can be created with the `optBlock` function in the `AlgDesign` package. The advantage of these designs is that the precision (or standard error) of the difference in every possible pair of treatment means will be equal. The disadvantage is that many blocks and experimental units may be required to achieve the balance. The other alternative design for one treatment factor is the **partially balanced incomplete block (PBIB)** designs.

The advantage of PBIB designs is that the total number of blocks and experimental units required can be reduced. The disadvantage is that the standard error of differences in pairs of treatment means will not be constant. There are many different methods of creating PBIB designs, and some of the more useful designs have been tabulated. One type of PBIB called a BTIB (balanced with respect to test treatments) can be easily created from a BIB design. This design is useful when there is more interest in comparing one treatment level (such as a control) to all other treatment levels than there is in comparisons among the other treatment levels. Another class of PBIB designs that can be easily created using the `design.cyclic` function in the `agricolae` package are called **generalized cyclic designs**.

Latin-square designs introduced in Chapter 4 have two orthogonal blocking factors and contain a complete block design in both the row blocks and `columnfunction` in the `AlgDesign` package. The advantage of these designs is that the precision (or standard error) of the difference in every possible pair of treatment means will be equal. The disadvantage is that many blocks and experimental units may be required to achieve the balance. The other alternative design for one treatment factor is the **partially balanced incomplete block (PBIB)** designs.

**Latin-square designs** introduced in Chapter 4 have two orthogonal blocking factors and contain a complete block design in both the row blocks and column blocks. If an incomplete block design is required in either the row or column blocks, a **row column design (RCD)** can be utilized. **Figure 7-4** illustrates when these designs should be used in relation to the designs presented in other chapters.



**Figure 7-4. Design Selection Roadmap**

When experimenting with multiple factors and the block size is not large enough to accommodate all possible treatment combinations, there are two alternative methods for creating an incomplete block design. The first method is to completely confound some interactions with blocks in

a **completely confounded blocked factorial (CCBF)** design, or a **completely confounded blocked fractional factorial (CCBFF)** design. The advantage of these designs is that the total number of blocks and experimental units can be minimized. The disadvantage is that some interactions will be completely confounded with blocks and will be inestimable. The other method is to use a partially confounded blocked factorial (PCBF) design.

## 8. Response Surface Designs

---

### 8.1. Introduction

In a response surface experiment, the independent variables or factors can be varied over a continuous range. The goal is to determine the factor settings that produce a maximum or minimum response or to map the relationship between the response and the factor settings over this contiguous factor space. As a practical matter, if we want to know a lot about the relationship between the factors and the response, it will require many experiments. For that reason, response surface designs are rarely conducted with more than six factors.

Response surface experiments are normally used at the last stage of experimentation. The important factors have already been determined in earlier experiments, and at this stage of experimentation the purpose is to describe in detail the relationship between the factors and the response. It is usually known or assumed that a simple linear model, even with interactions, is not good enough to represent that relationship. In order to locate maximums or minimums in the response as a function of the factor settings, at least three levels of each factor should be utilized.

Response surface methods generally refer to a complete package of statistical design and analysis tools that are used for the following three steps.

1. Design and collection of data to fit an equation to approximate the relationship between the factors and response.
2. Regression analysis to fit a model to describe the data.
3. Examination of the fitted relationship through graphical and numerical techniques.

Response surface methods have found considerable use in industry especially in chemical processes where the reaction yield or cost of production can be optimized as a function of the settings of controllable process factors. Since their origin these designs have also found successful applications in food science, engineering, biology, psychology, textiles, education, and many other areas.

## 8.2. Fundamentals of Response Surface Methodology

### 8.2.1. Empirical Quadratic Model

In response surface methods, the relationship between the response ( $y$ ) and the factor settings ( $x$ 's) is assumed to be a nonlinear equation given by  $y = f(x) + \varepsilon$ .

For two factors or independent variables we could write this equation as

$$y = f(x_1, x_2) + \varepsilon \quad \text{Eq. 8-1}$$

where  $f$  is a nonlinear function and  $\varepsilon$  is the random effect of experimental error. When  $f$  is unknown it can be approximated near the point  $(x_{10}, x_{20})$  using the two-term Taylor series approximation, that is,

$$\begin{aligned} & f(x_1, x_2) \approx f(x_{10}, x_{20}) + (x_1 - x_{10}) \frac{\partial f(x_1, x_2)}{\partial x_1} \Big|_{x_1=x_{10}, x_2=x_{20}} \\ & + (x_2 - x_{20}) \frac{\partial f(x_1, x_2)}{\partial x_2} \Big|_{x_1=x_{10}, x_2=x_{20}} \\ & + \frac{(x_1 - x_{10})^2}{2} \frac{\partial^2 f(x_1, x_2)}{\partial x_1^2} \Big|_{x_1=x_{10}, x_2=x_{20}} \\ & + (x_2 - x_{20}) \frac{\partial^2 f(x_1, x_2)}{\partial x_2^2} \Big|_{x_1=x_{10}, x_2=x_{20}} \\ & + \frac{(x_1 - x_{10})(x_1 - x_{10})^2}{2} \frac{\partial^2 f(x_1, x_2)}{\partial x_1 \partial x_2} \Big|_{x_1=x_{10}, x_2=x_{20}} \end{aligned} \quad \text{Eq. 8-2}$$

which leads to a **general quadratic equation** of the form

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{12} x_1 x_2 + \varepsilon \quad \text{Eq. 8-3}$$

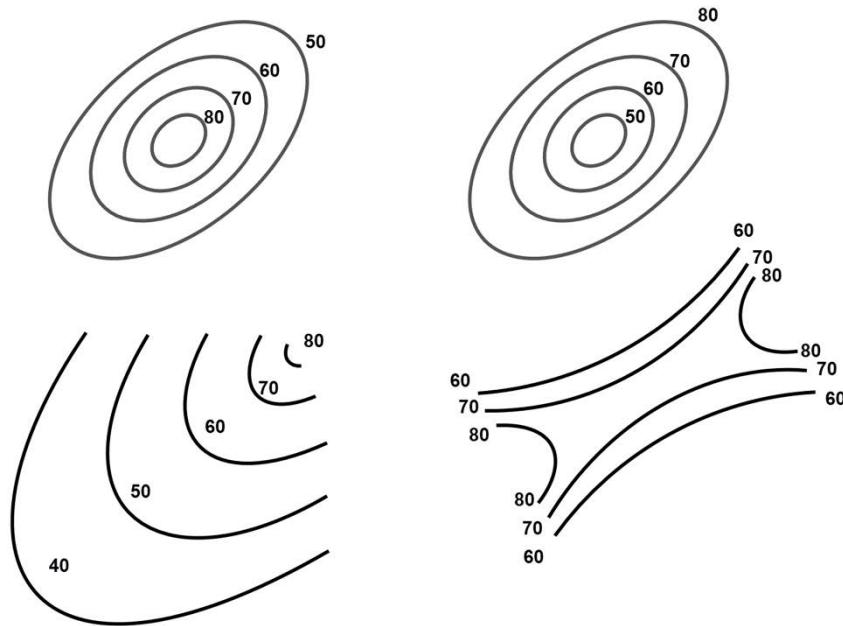
where  $\beta_1 = \frac{\partial f(x_1, x_2)}{\partial x_1} \Big|_{x_1=x_{10}, x_2=x_{20}}$ , and so on. If the region of interest is of moderate size, this general quadratic equation will provide a good approximation to  $f$  and can be used for interpolation within this region. The general quadratic equation is quite flexible and with appropriate

coefficients it can describe a wide variety of surfaces such as hilltops, valleys, rising or falling ridges, or saddle points as shown with contour plots in Figure 10.1.

With  $k$  factors or independent variables the general quadratic equation can be written in the form

$$y = \beta_0 + \sum_{i=1}^k \beta_i x_i + \sum_{i=1}^k \beta_{ii} x_i^2 + \sum_{i < j} \sum_{j < k} \beta_{ij} x_i x_j + \varepsilon \quad \text{Eq. 8-4}$$

and unless the function  $f$  is known, this equation forms the basis of response surface methods. Response surface designs were created to provide data to approximate this equation, and mathematical tools were created to explore the fitted surface represented by this equation.



*Figure 8-1. Surfaces That Can Be Described by General Quadratic Equation*

## 8.2.2. Design Considerations

The first requirement of a response surface design is that it should provide data that will allow estimation of the coefficients in **Eq. 8-4**. This model has  $1 + 2k + k(k - 1)/2$  coefficients, thus any response surface

design must have at least three levels for each factor (to allow estimation of quadratic terms) and at least  $1 + k + k(k - 1)/2$  total runs.  $3^k$  factorial designs have three levels for every factor. However, Box and Wilson (1951) showed that they were less satisfactory as a response surface design than an alternative design they called the central composite design. The central composite will be the first standard response surface design we will discuss in the next section.

Normally the factors or independent variables ( $x_i$ ) in the general quadratic **Eq. 8-4** are coded and scaled as shown in **Sections 3.8.1** and **3.8.2** so that the experimental region is a hyper-cube or hyper-sphere with radius  $R$  in the coded space. This general quadratic model can be written in matrix terms as

$$\mathbf{y} = \mathbf{x}\mathbf{b} + \mathbf{x}'\mathbf{B}\mathbf{x} + \varepsilon \quad \text{Eq. 8-5}$$

where  $\mathbf{x}' = (1, x_1, x_2, \dots, x_k)$ ,  $\mathbf{b}' = (\beta_0, \beta_1, \dots, \beta_k)$ , and the symmetric matrix

$$\mathbf{B} = \begin{pmatrix} \beta_{11} & \beta_{12}/2 & \dots & \beta_{1k}/2 \\ \square & \beta_{22} & \ddots & \beta_{2k}/2 \\ \square & \square & \ddots & \square \\ \square & \square & \square & \beta_{kk} \end{pmatrix}$$

When fitting a linear regression model of the form  $\mathbf{y} = \mathbf{x}\mathbf{b}$ , the design points are chosen to minimize the variance of the fitted coefficients  $\mathbf{b} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$ . Since the variance covariance matrix of estimated regression coefficients is  $\sigma^2(\mathbf{X}'\mathbf{X})^{-1}$ , this means the design points should be chosen such that the  $(\mathbf{X}'\mathbf{X})$  matrix is diagonal, or that the design is orthogonal like the  $2^k$  designs discussed in Chapter 3 and the  $2^{k-p}$  designs discussed in Chapter 6. When the  $(\mathbf{X}'\mathbf{X})$  matrix is diagonal, the diagonal elements of  $(\mathbf{X}'\mathbf{X})^{-1}$  will be minimized.

On the other hand, when fitting the general quadratic model, the primary purpose is not to understand the mechanism of the underlying relationship between the response and the factors. Therefore, the specific coefficients in the general quadratic model are of less importance. What is more important in a response surface study is to develop a prediction equation with the eventual goal of determining the

optimum operating conditions. Thus, the variance of a predicted value at  $\mathbf{x}$ , that is given by the equation  $Var[\hat{y}(\mathbf{x})] = \sigma^2 \mathbf{x}'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}$  is of more importance than the variance of the fitted coefficients.

Since it is not known before the experiment is conducted where the optimum will lie in the design region, one desirable property of a response surface design is to have the variance of a predicted value nearly the same everywhere in the design region. In this way, the precision in predicting the optimum will not depend upon its unknown position in the design region. The first step in equalizing the variance of prediction over the design region is to find a design that is **rotatable** (Myers & Montgomery, 2002). A rotatable design is one in which the variance of the predicted value at the point  $\mathbf{x}$  is only a function of the distance from the design origin to  $\mathbf{x}$ . Box and Wilson's central composite design and other standard response surface designs have this property.

Box and Hunter (1957) showed that certain rotatable designs could be modified to have **uniform precision**, which means the variance of a predicted value is the same at the origin and at the radius of one in the coded design region. A design with uniform precision is close to having the variance of a predicted value equal throughout the design region. Many standard response surface designs have this property or are close to having this property. When a standard response surface design cannot meet the needs of an experimenter, a computer algorithm can be used to construct a special purpose design. It would be reasonable to use a design that is *D*-optimal (see **Section 6.5.2**) for **Eq. 8-4** as a response surface design. However, the *D*-optimality criterion minimizes the variance of model coefficients. The *I*-optimality criterion minimizes the average variance of a predicted value, and therefore it is more appropriate for finding a response surface design. Programs, such as **rov** function in the R package **AlgDesign**, can be used for creating I-optimal and D-optimal designs. Giovannitti-Jensen and Myers (1989) and Myers et al. (2004) describe a variance dispersion plot which can be used to visually evaluate whether any response surface design is close to having the properties of rotatability and uniform precision.

In addition to providing a good distribution of the variance of a predicted value, two other desirable properties of response surface designs are (1)

to provide an estimate of the “pure” experimental error so the adequacy of the general quadratic model can be checked, and (2) to allow for blocking so that an experimenter can begin with a linear design and add a second block to estimate curvature if necessary. To estimate pure error, at least one design point must be replicated. When blocking, the first block would consist of a design such as a  $2^k$  or  $2^{k-p}$  augmented with center points that allow for checking the adequacy of the linear model. If the linear model is adequate, no further experimentation is required. If the linear model is not adequate, a second block of experiments can be added which will allow estimation of the quadratic coefficients in the general quadratic model.

### 8.3. Standard Designs for Second Order Models

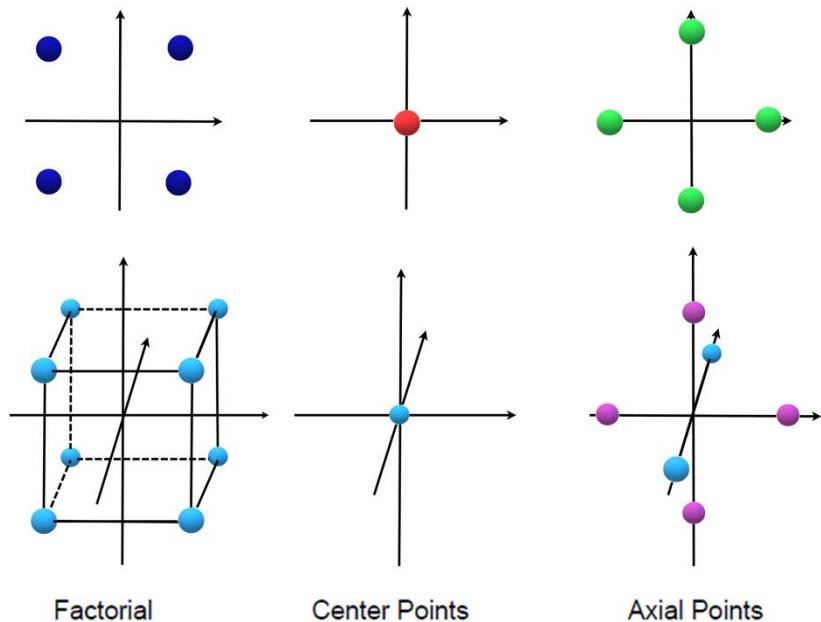
This section presents some of the more popular **completely randomized response surface** (CRRS) designs.

#### 8.3.1. Central Composite Design (CCD)

Box and Wilson’s (1951) **central composite design** (CCD) consists of a  $2^k$  factorial or  $2^{k-p}$  fractional factorial design augmented with center points and axial points as shown in Figure 10.2. The  $2^k$  or *resolution V*  $2^{k-p}$  design allows estimation of the linear and linear-by-linear terms in the general quadratic model. The addition of the center points and axial points allow estimation of the quadratic terms. By choosing the distance from the origin to the axial points ( $\alpha$  in coded units) equal to  $\sqrt[4]{F}$  where  $F$  is the number of points in the factorial portion of the design, a CCD will be rotatable. By choosing the correct number of center points, the CCD will have the uniform precision property.

As an example of a CCD, consider the results of the experiment described by Kamon et al. (1999) in Table 10.1. The experiment was conducted to find the optimal formulation of a two-component admixture to improve the **workability** of cement grouts. The factors were the water-to-cement ratio, the percent of black liquor added, and the percent of SNF added. The left side of the table<sup>9</sup> shows the coded factors  $x_1 - x_3$ . The axial points were chosen at  $\pm 1.68 = \sqrt[4]{8}$  in coded units to make the design rotatable. The coded levels are found as (*actual level* –

*center value)/(half – range).* For example,  $x_2 = (BlackLiq. - 0.150) \sim 0.03$ . The actual factor levels on the right side of the table can be obtained from the coded values on the left side of the table by solving  $actual\ level = (half - range) \times x_i + center\ value$ .



*Figure 8-2. CCD in Two and Three Dimensions*

*Table 8-1. CCD in Coded and Actual Units for Cement Workability Experiment*

Run	x1	x2	x3	Water/ Cement	Black Liq.	SNF	y
1	-1	-1	-1	0.330	0.120	0.080	109.5
2	1	-1	-1	0.350	0.120	0.080	120.0
3	-1	1	-1	0.330	0.180	0.080	110.5
4	1	1	-1	0.350	0.180	0.080	124.5
5	-1	-1	1	0.330	0.120	0.120	117.0
6	1	-1	1	0.350	0.120	0.120	130.0
7	-1	1	1	0.330	0.180	0.120	121.0
8	1	1	1	0.350	0.180	0.120	132.0
9	0	0	0	0.340	0.150	0.100	117.0
10	0	0	0	0.340	0.150	0.100	117.0
11	0	0	0	0.340	0.150	0.100	115.0
12	-1.68	0	0	0.323	0.150	0.100	109.5
13	1.68	0	0	0.357	0.150	0.100	132.0

14	0	-1.68	0	0.340	0.100	0.100	120.0
15	0	1.68	0	0.340	0.200	0.100	121.0
16	0	0	-1.68	0.340	0.150	0.066	115.0
17	0	0	1.68	0.340	0.150	0.134	127.0
18	0	0	0	0.340	0.150	0.100	116.0
19	0	0	0	0.340	0.150	0.100	117.0
20	0	0	0	0.340	0.150	0.100	117.0

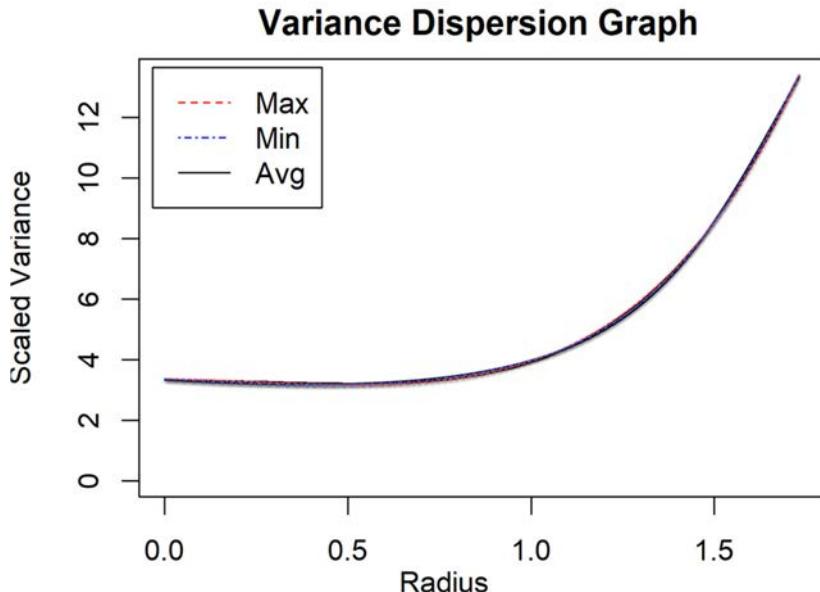
By including six center points, the design has uniform precision, and the variance of a predicted value will be the same at the origin  $(0, 0, 0)$ , in coded factor levels, and at any point at radius 1 in coded units. Box and Hunter (1957) have tabulated the number of center points required to make a central composite design uniform precision for various values of  $k = \text{number of factors}$ .

(1989) The R code below calls the `ccd` function from the R package `rsm` (Lenth, Lenth (2009)) to create a uniform precision central composite design in three factors. The factor columns from the resulting design object `rotd` are copied into the data frame `rotdm` and the `Vdgraph` function from the R `Vdgraph` package (Lawson (2013a), Lawson (2013b)) is then called to create Myers et al.'s (2004) variance dispersion graph for this design that is shown in **Figure 8-3**. The variance dispersion graph plots the maximum, minimum, and average scaled variance of a predicted value ( $NV ar(\hat{y}(x))/\sigma^2$ ) as a function of the distance from the origin of the design. When the design is not rotatable, there will be three distinct lines on the graph. The closer together the lines, the closer the design is to being rotatable. In **Figure 8-3** the lines overlap showing that the design is rotatable. It can also be seen that the value of the scaled variance of a predicted value at the origin is nearly equal to the variance at radius 1.0, indicating the design has uniform precision.

```
library(rsm)
rotd <- ccd(3, n0 = c(4,2), alpha = "rotatable",
randomize = FALSE)
rotdm <- rotd[, 3:5]
library(Vdgraph)
Vdgraph(rotdm)
```

The experiments in **Table 8-1** should be run in a random order, but they are listed in a non-random order to illustrate another useful aspect of

central composite designs. This design can be run as a sequence of two blocks. In some applications this is a desirable approach. The first eight runs in the table represent a full  $2^3$  factorial. Runs 9–11 are center points at the mid-level of each factor. If these 11 experiments were completed in the first block, the data could be analyzed to see if a linear model



*Figure 8-3. Variance Dispersion Graph for Uniform Precision CCD in Three Factors*

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_{12} x_1 x_2 + \beta_{13} x_1 x_3 + \beta_{23} x_2 x_3 \quad \text{Eq. 8-6}$$

is adequate to represent the data. If so, runs number 12–20 need not be completed. If the linear model is not adequate to represent the data, the second set of experiments can be completed, and a block variable added to the model to account for any changes in background variables that occurs between the first and second set of experiments.

### 8.3.2. Box-Behnken Design

Whereas the CCD requires five levels ( $-\alpha$ ,  $-1$ ,  $0$ ,  $+1$ ,  $+\alpha$ ) for each factor, Box and Behnken (1960) developed some three level designs that will allow estimation of the general quadratic model. These designs consist of 22 factorials in each pair of factors with all other factors held

constant at their mid-level plus a few center points. No Box-Behnken design exists for only two factors. An example of a **Box-Behnken** design in three coded factors is shown in **Table 8-2**.

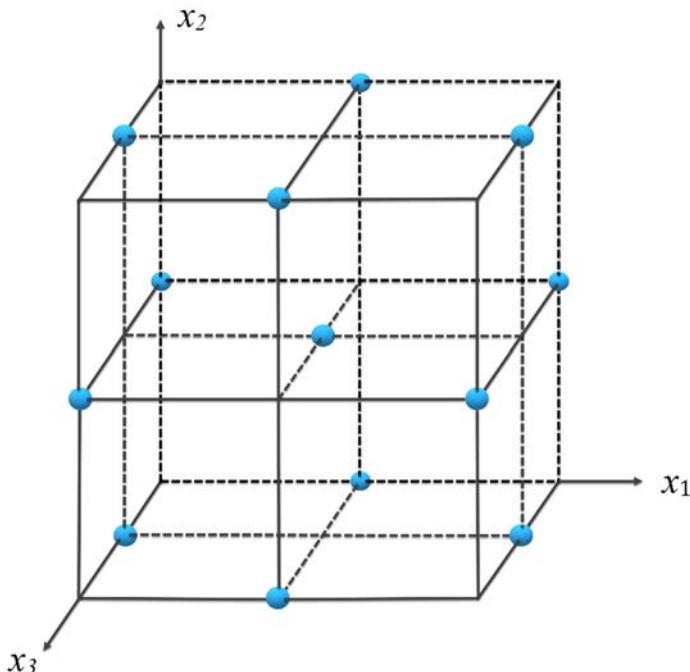
**Table 8-2. Box-Behnken Design in Three Factors**

run	x1	x2	x3
1	-1	-1	0
2	1	-1	0
3	-1	1	0
4	1	1	0
5	-1	0	-1
6	1	0	-1
7	-1	0	1
8	1	0	1
9	0	-1	-1
10	0	1	-1
11	0	-1	1
12	0	1	1
13	0	0	0
14	0	0	0
15	0	0	0

Box-Behnken designs have two advantages over CCDs. The first advantage is that they only require that factors be varied over three levels. This may make experimentation less costly if actual prototypes are being constructed in the experimentation. The second advantage is that they usually (except for the five-factor case) require fewer total runs than the central composite design. For example, the three-factor CCD required 20 runs whereas the three-factor Box-Behnken design only required 15 runs.

A disadvantage of a Box-Behnken design compared to a central composite design is that it cannot be built up in two steps beginning with a  $2^k$  design. This can be visualized in **Figure 8-4**, which diagrams a three-factor Box-Behnken design. A  $2^3$  design consists of runs at all corners of the cube as shown in **Section 3.8.1**, but as can be seen in **Figure 8-4** the Box-Behnken design does not include those points. Therefore a Box-Behnken design should be used if the experimenter is reasonably sure that a linear model will not adequately represent the relationship

between the factors and the response, and he or she wants to save a few runs. Another possible disadvantage is that the factors have only three levels. While this may be less costly when building prototypes, having three levels leaves nothing to check the adequacy of the quadratic model.

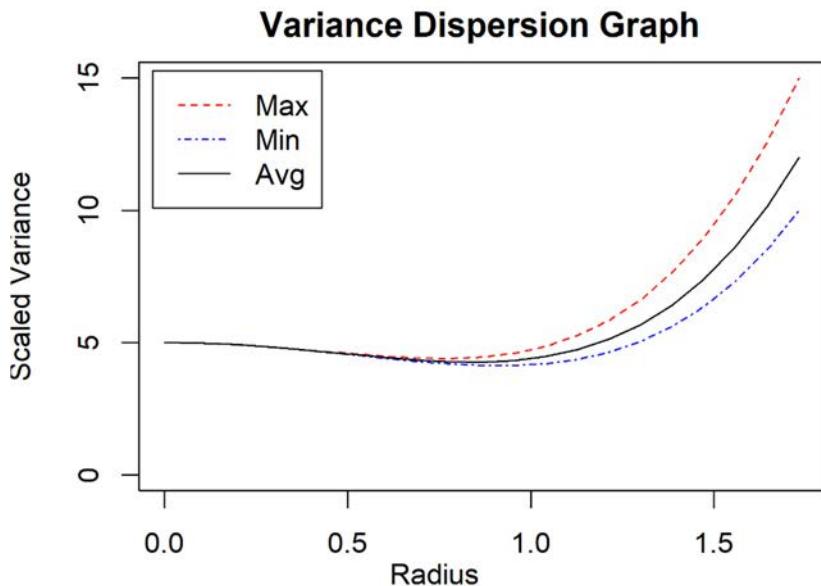


*Figure 8-4. Graphical Representation of Three-Factor Box-Behnken Design*

**Figure 8-5** shows the variance dispersion graph for the three-factor Box-Behnken design. Since the lines for the minimum, maximum, and average scaled variance of a predicted value do not overlap as in **Figure 8-4**, the Box-Behnken design is not rotatable. But, since the lines are very close together within the coded experimental region to a radius of 1.0, the Box-Behnken design is close enough to being rotatable and having uniform precision for practical use. This is true for Box-Behnken designs with  $k = 3$  to 6.

Another graph that is useful evaluating the variance of predicted values from a response surface design is the fraction of design space plot (Zahran, Anderson-Cook, C. M., & Myers, 2003). This plot shows the

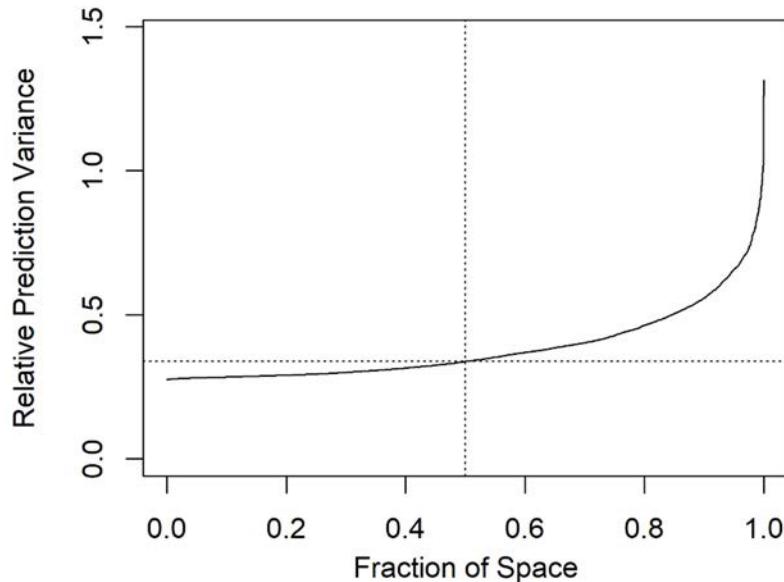
relative variance of a predicted value,  $V ar(\hat{y}(x))/\sigma^2$ , on the vertical axis versus the fraction of points in the design region on the horizontal axis. **Figure 8-6** shows the fraction of design space plot for the Box-Behnken design. From this plot we can see that the relative variance of a predicted value is less than 0.35 in 50% of the design region. This plot can be made with the `FDSPlot` function that is also in the `Vdgraph` package. Variance dispersion graphs are better for evaluating how close a design is to being rotatable and having uniform precision. Fraction of design space plots are generally better for comparing contending designs for a research problem.



**Figure 8-5. Variance Dispersion Graph for Box-Behnken Design in Three Factors**

As an example of a Box-Behnken design, consider the experiment performed by Anderson (2003). He experimented with a scale model trebuchet (medieval missile launcher) that was issued to all engineering students at the South Dakota School of Mines and Technology for hands-on experimentation. He wanted to determine the settings that would allow him to toss an object a given distance to fend off attackers or door-to-door salespeople. He varied three factors on the trebuchet. *A*, the arm length 4 to 8 inches from the counterweight end to the point where the weights were hung; *B*, counterweight 10 to 20 pounds; and *C*,

missile weight 2 to 3 ounces. The Box-Behnken design in actual factor levels is shown in **Table 8-3** along with the response (the distance the missile flew).



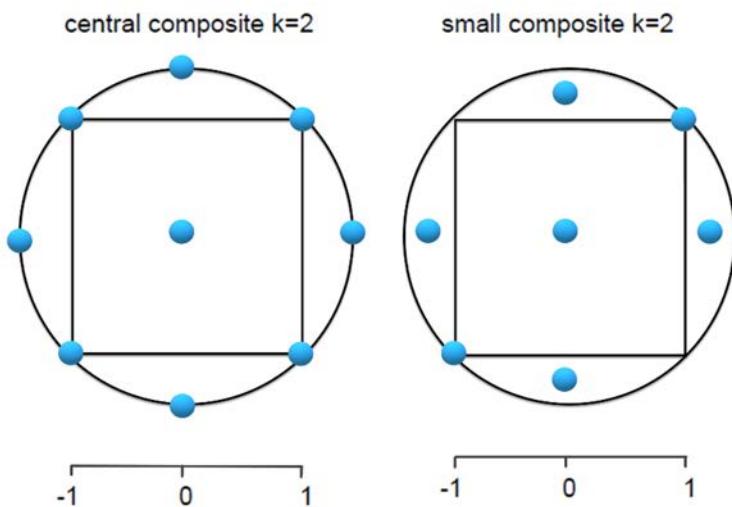
**Figure 8-6. Fraction of Design Space Plot for Box-Behnken Design in Three Factors**

**Table 8-3. Box-Behnken Design Results for Trebuchet Experiments**

run	A	B	C	y
1	4	10	2.5	33
2	8	10	2.5	85
3	4	20	2.5	86
4	8	20	2.5	113
5	4	15	2	75
6	8	15	2	105
7	4	15	3	40
8	8	15	3	89
9	6	10	2	83
10	6	20	2	108
11	6	10	3	49
12	6	20	3	101
13	6	15	2.5	88
14	6	15	2.5	91
15	6	15	2.5	91

### 8.3.3. Small Composite Design

When the cost of each experiment is high, and an experimenter is willing to compromise some of the desirable properties of a response surface design in order to reduce the run size, a small composite design can be utilized. In a small composite design the  $2^k$  or *resolution V*  $2^{k-p}$  part of the central composite design is replaced with a *resolution III*  $2^{k-p}$ . In the central composite design, the  $2^k$  or *resolution V*  $2^{k-p}$  was included to allow estimation of all linear main effect terms and linear by linear two-factor interactions. Hartley (1959) has shown that when center points and axial points are included, two-factor interactions can be estimated with a *resolution III fractional factorial*. Westlake (1965) obtained additional small composite designs by substituting irregular fractions for the factorial portion of the design, and Draper (1985) and Draper and Lin (1990) substituted Plackett-Burman designs for the factorial portion to come up with even more small composite designs.

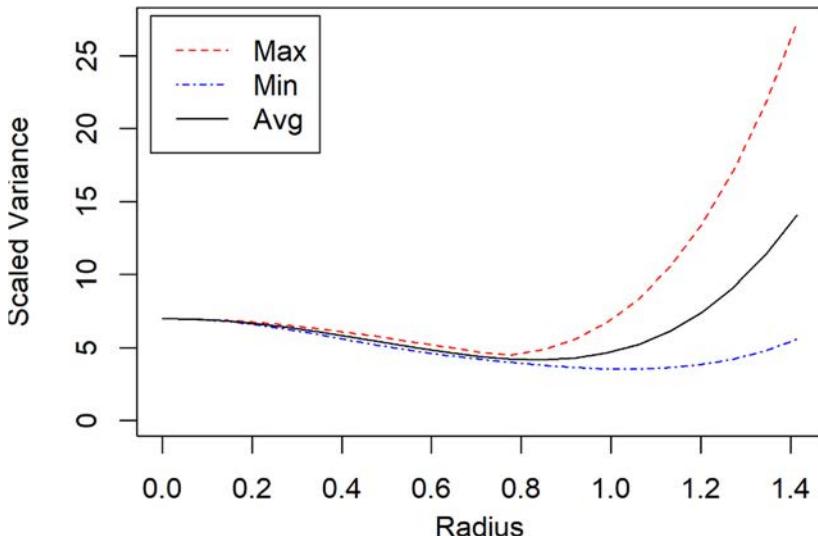


**Figure 8-7. Graphical Comparison of CCD and Small Composite (with  $I = AB$ ) for  $k=2$**

**Figure 8-7** is a graphical comparison of the central composite and small composite design for  $k = 2$  factors. The central composite design has five center points for a total of 13 runs, while the small composite design has one center point with a total of 7 runs.

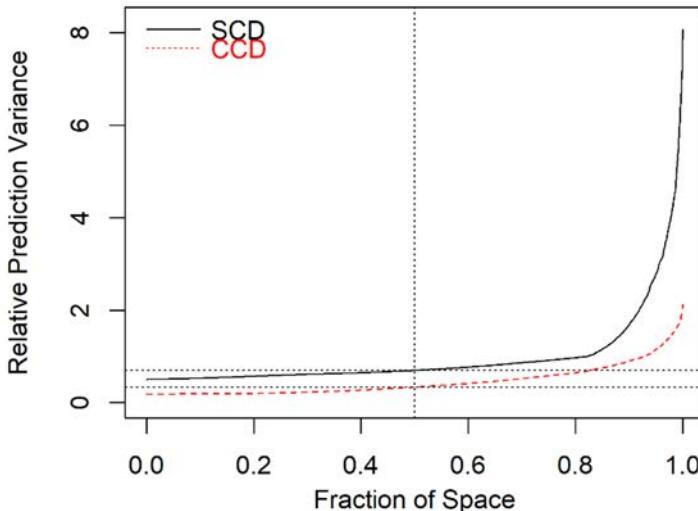
**Figure 8-8** is the variance dispersion graph for the small composite design with  $k = 2$ . This shows the design is near rotatable within a coded radius of 0.8, but not uniform precision since the scaled prediction

variance is larger in the center of the design than at a coded radius of 0.8. The variance dispersion graph for the uniform precision central composite design with  $k = 2$  would look very similar to **Figure 8-3**.



**Figure 8-8. Variance Dispersion Graph of Small Composite Design with 2 Factors**

**Figure 8-9** is a comparison of the fraction of design space plots for the central composite and small composite designs for  $k = 2$  factors. This graph is better for comparing the two designs. It can be seen, by reference to the upper horizontal dotted line in the graph, that the relative prediction variance would be less than 0.64 in 50% of the design space if the small composite design were used. On the other hand, the relative prediction variance would be less than 0.64 over more than 80% of the design region if the central composite design were used. The relative variance of prediction is uniformly smaller for the central composite design. The choice between the two designs would be made by weighing the extra precision the central composite design affords to the cost of the six additional experiments it requires.



**Figure 8-9. Comparison of Fraction of Design Space Plots for Small Composite Design and Central Composite Design for Two Factors**

### 8.3.4. Hybrid Design

Roquemore (1976) developed hybrid designs that require even fewer runs than the small composite designs. These designs were constructed by making a central composite design in  $k - 1$  factors and adding a  $k$ th factor so that the  $X'X$  has certain properties and the design is near rotatable. **Table 8-4** shows an example of the Roquemore 310 design.

The design is labeled 310 because it has 3 factors and 10 runs. It can be seen that there is a central composite design in columns 1 and 2. This design is close to rotatable only within a coded radius of 0.5 as shown in **Figure 8-10**, but since it only has 10 runs, there are zero degrees of freedom for estimating the experimental error. Some of the Roquemore hybrid designs leave one degree of freedom for estimating experimental error, but none of them have pure replicates that are required for testing the adequacy of the general quadratic model. Therefore these designs should only be used when the experimenter is confident that the general quadratic model will represent the data well, and there is an independent estimate of experimental error from previous data.

**Table 8-4. Roquemore 310 Design**

Run	x1	x2	x3
-----	----	----	----

1	0	0	1.2906
2	0	0	-0.1360
3	-1	-1	0.6386
4	1	-1	0.6386
5	-1	1	0.6386
6	1	1	0.6386
7	1.736	0	-0.9273
8	-1.736	0	-0.9273
9	0	1.736	-0.9273
10	0	-1.736	-0.9273

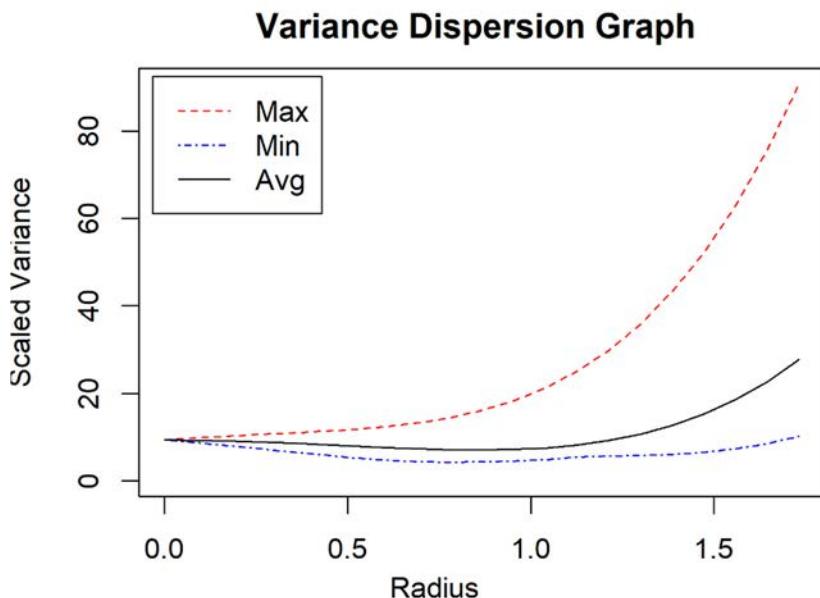


Figure 8-10. Variance Dispersion Graph for Roquemore 310 Design

#### 8.4. Creating Standard Response Surface Designs in R

Standard response surface designs can be created using R packages. The `rsm` package has functions for creating both central composite designs and Box-Behnken designs. This package also has a function for listing various central composite designs available for a specified number of factors. The `Vdgraph` package contains stored data frames that contain small composite and hybrid designs for 2–6 factors.

If a standard response surface design for three factors is desired, the R code below illustrates the call to the function `ccd.pick` in the `rsm` package to show the possible central composite designs available.

```
library(rsm)
ccd.pick(k=3)
```

	n.c	n0.c	blk.s.c	n.s	n0.s	bbr.c	wbr.s	bbr.s	N	alpha.rot	
alpha.orth	1	8	9	1	6	6	1	1	1 29	1.6818	1.680
	1	8	2	1	6	1	1	1	1 17	1.6818	1.673
	2	8	6	1	6	4	1	1	1 24	1.6818	1.690
	3	8	5	1	6	3	1	1	1 22	1.6818	1.664
	4	8	10	1	6	7	1	1	1 31	1.6818	1.699
	5	8	8	1	6	5	1	1	1 27	1.6818	1.658
	6	8	3	1	6	2	1	1	1 19	1.6818	1.705
	7	8	7	1	6	5	1	1	1 26	1.6818	1.712
	8	8	4	1	6	2	1	1	1 20	1.6818	1.632
	9	8	4	1	6	3	1	1	1 21	1.6818	1.732
	10	8	4	1	6	3	1	1			

To create one of the designs listed in the output of the `ccd.pick` function, the `ccd` function in the `rsm` package can be used as shown below.

```
library(rsm)
ccd.up <- ccd(y ~ x1 + x2 + x3, n0 = c(4,2),
alpha = "rotatable", randomize = FALSE)
```

This code produces the 9th design that has `n0.c=4`, and `n0.s=2`. This design has uniform precision as can be visualized by its variance dispersion graph that can be produced with the command `Vdgraph(ccd.up[, 3:5])`.

The design `ccd.up` produced in the above code has coded factor levels that vary between -1.68 and +1.68. If the coding option is used, as shown in the code below, the resulting listing of the design shows the actual factor levels.

```
ccd.up <- ccd(y ~ x1 + x2 + x3, n0=c(4,2),
alpha="rotatable", coding=list(x1~(Temp-150)/10,x2~(Press-
50)/5, x3~(Rate-4/1)),randomize=FALSE)
head(ccd.up)
```

	run.order	std.order	Temp	Press	Rate	y	Block
1	1	1	140	45	3	NA	1
2	2	2	160	45	3	NA	1
3	3	3	140	55	3	NA	1
4	4	4	160	55	3	NA	1
5	5	5	140	45	5	NA	1
6	6	6	160	45	5	NA	1

Data are stored in coded form using these coding formulas ...

```
x1 ~ (Temp - 150)/10
x2 ~ (Press - 50)/5
x3 ~ (Rate - 4/1)
```

However, they are still stored in coded levels for computational purposes when fitting a model to the design and examining the fitted surface. Other options for alpha are "spherical" to place axial points the same distance from the origin as corners of the design, and "faces" to place axial points at high and low factorial levels, sometimes called a face-centered cube design.

The function `bbd` in the `rsm` package generates Box-Behnken designs in three to seven factors. For example, the code below produces the design shown in Table 10.3.

```
library(rsm)
Treb<-bbd(y ~ x1 + x2 + x3, randomize = FALSE, n0 = 3,
Coding = list(x1 ~ (A-6)/2, x2 ~ (B-15)/5,
x3 ~ (C-2.5)/.5))
```

The `rsm` package does not have functions to create hybrid designs or small composite designs directly but the `Vdgraph` package has the most common designs stored as data frames that can be recalled. Table 10.5 lists the designs available as data frames in this package. The Hex2 is a rotatable near uniform precision design in two factors that contains only nine runs. These runs are situated on the vertices of a hexagon in a two-dimensional space.

*Table 8-5. Reduced Run Response Surface Designs in Package Vdgraph*

#### **Small Composite Designs:**

Data Frame Name	Description
SCDDL5	Draper and Lin's Design for 5-factors
SCDH2	Hartley's Design for 2-factors

- SCDH3 Hartley's Design for 3-factors
- SCDH4 Hartley's Design for 4-factors
- SCDH5 Hartley's Design for 5-factors
- SCDH6 Hartley's Design for 6-factors

### **Hybrid Designs:**

Data Frame Name	Description
D310	Roquemore's hybrid design D310
D311A	Roquemore's hybrid design D311A
D311B	Roquemore's hybrid design D311B
D416A	Roquemore's hybrid design D416A
D416B	Roquemore's hybrid design D416B
D416C	Roquemore's hybrid design D416C
D628A	Roquemore's hybrid design D628A

### **Hexagonal Design:**

Data Frame Name	Description
Hex2	Hexagonal Design in 2-factors

A design can be recalled as shown in the example below.

```
library(Vdgraph)
data(D310)
D310
```

	x1	x2	x3
1	0.0000	0.000	1.2906
2	0.0000	0.000	-0.1360
3	-1.0000	-1.000	0.6386
4	1.0000	-1.000	0.6386
5	-1.0000	1.000	0.6386
6	1.0000	1.000	0.6386
7	1.7636	0.000	-0.9273
8	-1.7636	0.000	-0.9273
9	0.0000	1.736	-0.9273
10	0.0000	-1.736	-0.9273

The data frames in **Vdgraph** have the factors named  $x_1 - x_k$ , and coded factor levels. To make a listing of the design in random order and actual factor levels use the commands similar to those shown below.

```
des <- transform(D310, Temp=10*x1+150, Press=5*x2+50,
```

```
Rate = x3+ 4)  
des[sample(1:10),4:6]
```

	Temp	Press	Rate
2	150.000	50.00	3.8640
10	150.000	41.32	3.0727
5	140.000	55.00	4.6386
1	150.000	50.00	5.2906
9	150.000	58.68	3.0727
4	160.000	45.00	4.6386
7	167.636	50.00	3.0727
6	160.000	55.00	4.6386
3	140.000	45.00	4.6386
8	132.364	50.00	3.0727

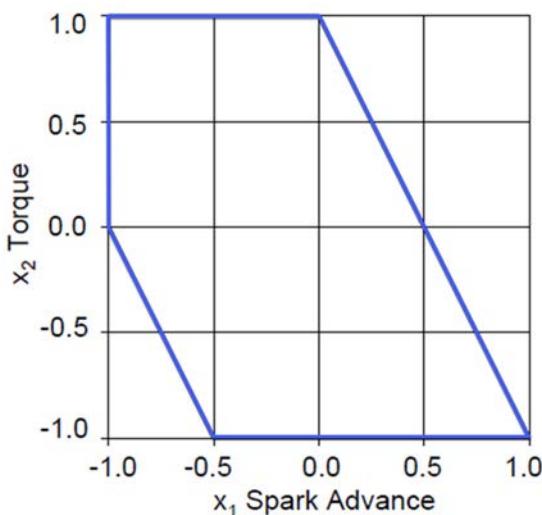
## 8.5. Non-Standard Response Surface Designs

Some design situations do not lend themselves to the use of standard response surface designs for reasons such as (1) the region of experimentation is irregularly shaped, (2) not all combinations of the factor levels are feasible, (3) there is a non-standard linear or a nonlinear model. In these situations, standard response surface designs will not be appropriate. One way to construct a response surface design in these cases is to use a computer algorithm to construct an I-optimal design.

For an example of the first situation where the region of experimentation is irregularly shaped, Atkinson et al. (2007) describe an experiment to investigate the performance of an internal combustion engine. Two factors under study were the spark advance and the torque. Both are independently variable, so the coded design region is a square. However, for factor combinations outside the pentagonal region shown in **Figure 8-11**, it is likely the engine would not run or would be seriously damaged.

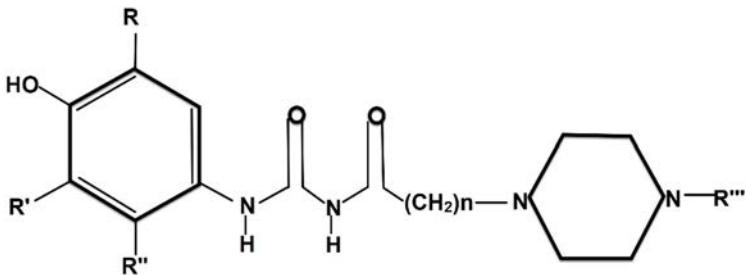
Therefore the experimental region is irregular, and the standard two-factor response surface designs would not fit in this region. A response surface design can be constructed for this problem by defining candidate points that fall within a grid over the experimental region (shown in **Figure 8-11**). This can be done in R, and the `optFederov` function in the `AlgDesign` package can be used to select an I-optimal subset for fitting the general quadratic model.

As an example of the second situation where not all possible combinations of the factor are feasible, consider a typical study in QSAR (Quantitative Structure Activity Relation) used in drug design. In this type of study, when a lead compound is discovered with desirable bioactivity, many derivatives of this compound are considered in an attempt to find one that will increase the desired effect. For example, **Figure 8-12** shows the general structure of *hydroxyphenylureas*, which have been shown to be active as antioxidants.



**Figure 8-11. Experimental Region for Engine Experiment**

The compound has four receptor sites ( $R$ ,  $R'$ ,  $R''$ , and  $R'''$ ) (shown in the figure) where different substituent atoms or molecules can be attached to modify the basic structure. With many possible substituent molecules that can be added at each receptor site, there is a large library of candidates that could be synthesized and tested for activity. Table 10.6 is an illustrative list that shows 36 different possible *hydroxyphenylureas* from a longer list considered by Deeb et al. (2008).



**Figure 8-12. General Structure of Hydroxyphenylureas**

This list of candidates is stored in a data frame in the R package `daewr` and can be recalled with the command below.

```
library(daewr)
data(qsar)
```

For each possible variant compound in the library there are several physical chemistry descriptors that can be calculated.

**Table 8-6. Library of Substituted Hydroxyphenylurea Compounds**

Compound	R	R'	R''	R'''	HE	DMz	SOK
1	H	H	H	CH <sub>3</sub>	-12.221	-0.162	64.138
2	H	H	H	CH <sub>2</sub> Ph	-14.015	-0.068	88.547
3	H	H	H	Ph	-14.502	0.372	85.567
4	H	H	H	2CH <sub>3</sub> OC <sub>6</sub> H <sub>4</sub>	-14.893	1.035	96.053
5	H	OCH <sub>3</sub>	H	CH <sub>3</sub>	-12.855	1.091	74.124
6	H	OCH <sub>3</sub>	H	CH <sub>2</sub> Ph	-14.628	1.115	99.002
7	H	OCH <sub>3</sub>	H	Ph	-15.123	1.554	96.053
8	H	OCH <sub>3</sub>	H	2CH <sub>3</sub> OC <sub>6</sub> H <sub>4</sub>	-15.492	2.221	106.607
9	H	OC <sub>2</sub> H <sub>5</sub>	H	CH <sub>3</sub>	-11.813	1.219	77.02
10	H	OC <sub>2</sub> H <sub>5</sub>	H	CH <sub>2</sub> Ph	-13.593	1.188	101.978
11	H	OC <sub>2</sub> H <sub>5</sub>	H	Ph	-14.088	1.621	99.002
12	CH <sub>3</sub>	OC <sub>2</sub> H <sub>5</sub>	H	2CH <sub>3</sub> OC <sub>6</sub> H <sub>4</sub>	-14.46	2.266	109.535
13	CH <sub>3</sub>	H	CH <sub>3</sub>	CH <sub>3</sub>	-8.519	-0.56	71.949
14	CH <sub>3</sub>	H	CH <sub>3</sub>	CH <sub>2</sub> Ph	-10.287	-0.675	96.6
15	CH <sub>3</sub>	H	CH <sub>3</sub>	Ph	-10.798	-0.134	96.62
16	CH <sub>3</sub>	H	CH <sub>3</sub>	2CH <sub>3</sub> OC <sub>6</sub> H <sub>4</sub>	-11.167	0.418	104.047
17	H	H	H	CH <sub>3</sub>	-12.245	-0.609	67.054
18	H	H	H	CH <sub>2</sub> Ph	-13.98	-0.518	91.546
19	H	H	H	Ph	-14.491	-0.561	88.547
20	H	H	H	2CH <sub>3</sub> OC <sub>6</sub> H <sub>4</sub>	-14.888	-1.478	99.002
21	H	OCH <sub>3</sub>	H	CH <sub>3</sub>	-11.414	-1.888	77.02
22	H	OCH <sub>3</sub>	H	CH <sub>2</sub> Ph	-13.121	-1.692	101.978
23	H	OCH <sub>3</sub>	H	Ph	-13.66	-1.893	99.002

24	H	OCH <sub>3</sub>	H	2CH <sub>3</sub> OC <sub>6</sub> H <sub>4</sub>	-14.012	-2.714	109.535
25	H	OC <sub>2</sub> H <sub>5</sub>	H	CH <sub>3</sub>	-10.029	-1.891	79.942
26	H	OC <sub>2</sub> H <sub>5</sub>	H	CH <sub>2</sub> Ph	-11.74	-1.652	104.977
27	H	OC <sub>2</sub> H <sub>5</sub>	H	Ph	-12.329	-1.902	101.978
28	OCH <sub>3</sub>	OC <sub>2</sub> H <sub>5</sub>	H	2CH <sub>3</sub> OC <sub>6</sub> H <sub>4</sub>	-12.637	-2.762	112.492
29	OCH <sub>3</sub>	OCH <sub>3</sub>	H	CH <sub>3</sub>	-12.118	-2.994	81.106
30	OCH <sub>3</sub>	OCH <sub>3</sub>	H	CH <sub>2</sub> Ph	-13.892	-2.845	106.299
31	OCH <sub>3</sub>	OCH <sub>3</sub>	H	Ph	-14.456	-2.926	103.23
32	OCH <sub>3</sub>	OCH <sub>3</sub>	H	2CH <sub>3</sub> OC <sub>6</sub> H <sub>4</sub>	-14.804	-3.78	113.856
33	CH <sub>3</sub>	H	CH <sub>3</sub>	CH <sub>3</sub>	-9.209	-0.423	74.871
34	CH <sub>3</sub>	H	CH <sub>3</sub>	CH <sub>2</sub> Ph	-10.97	-0.302	99.603
35	CH <sub>3</sub>	H	CH <sub>3</sub>	Ph	-11.488	-0.453	96.6
36	CH <sub>3</sub>	H	CH <sub>3</sub>	2CH <sub>3</sub> OC <sub>6</sub> H <sub>4</sub>	-11.868	-1.322	107.01

In **Table 8-6** three descriptors are shown: Hydration Energy (HE), Molecular Dipole Moment at the z-direction (DM<sub>z</sub>), and Symmetry Index (SOK). The objective of a QSAR study would be to (1) synthesize a subset of the molecules from the large library and test them to determine their ability to scavenge oxygen FRs as measured by the binding constant  $\log K_{app}$ , and (2) fit a model relating the response  $\log K_{app}$  to several physical chemistry descriptors (in this case, the variables HE, DM<sub>z</sub>, and SOK). Once a model relating the response as a function of the independent variables has been found, the combination of the independent variables that is predicted to give the maximum response can be determined. Compounds in the large library that have values of the physical chemistry descriptors (independent variables) closest to those predicted to result in a maximum response can then be synthesized and tested for activity. In this way variant compounds that have increased activity are discovered.

In QSAR studies, standard response surface designs cannot be used because not all potential combinations of the independent variables are possible. Only a subset of the combinations of independent variables that exist in the large library can be chosen in a subset or experimental design to be tested. Again the `optFederov` function in the `AlgDesign` package can be utilized to create an I-optimal or D-optimal subset of the larger library. In this case, the list of candidates will be all the compounds in the library. The design must consist of a subset of the compounds in the library.

If the general quadratic model will be used to fit the data with three fac-

tors, there are 10 coefficients in the general quadratic model so the number of runs in the design must be at least  $n = 10$ . In the R code below the option `nRepeats=40` instructs the `optFederov` function to make 40 different searches for the D-optimal design with a random starting design and to keep the best results. The option `nTrials=16` instructs the `optFederov` function to make a design with 16 runs. The design will be stored in the data frame `desgn1$design`, the fourth item in the list `desgn1`. The `optFederov` function call is repeated changing criterion option from D to I to create an I-optimal design that is stored in the data frame `desgn2$design`. The last statement calls the `Compare2FDS` function to compare the fraction of design space plots for the two designs.

```
library(daewr)
data(qsar)

library(AlgDesign)
desgn1<- optFederov(~quad(.), data = qsar, nTrials = 15,
center = TRUE, criterion = "D", nRepeats = 40)
desgn2<-optFederov(~quad(.), data = qsar, nTrials = 15,
center = TRUE, criterion = "I",nRepeats=40)
Compare2FDS(desgn1$design, desgn2$design, "D-optimal",
"I-optimal", mod=2)
```

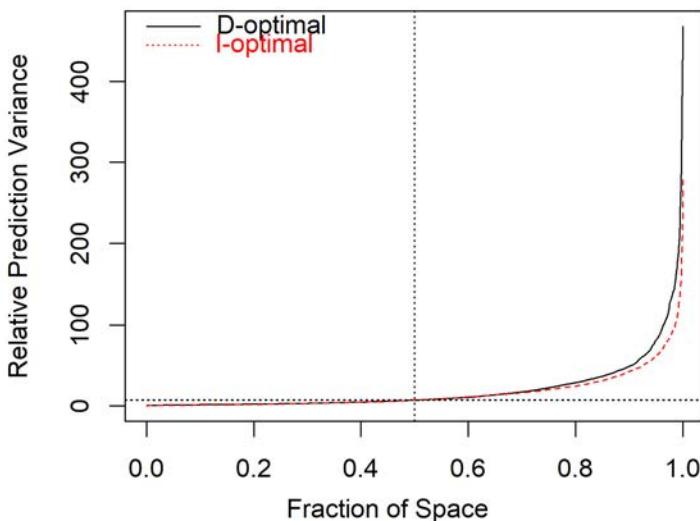
The I-optimal design can be listed as shown on the next page. Once the designs are created with the `optFederov` function, they can be compared with respect to the variance of a predicted value. Although neither design is rotatable, the I-optimal design is slightly better since its relative prediction variance appears uniformly less than the relative prediction variance of the D-optimal design as shown in **Figure 8-13**.

<code>desgn2\$design</code>
-----------------------------

	Compound	HE	DMz	S0K
1	1	-12.221	-0.162	64.138
4	4	-14.893	1.035	96.053
9	9	-11.813	1.219	77.020
12	12	-14.460	2.266	109.535
13	13	-8.519	-0.560	71.949
14	14	-10.287	-0.675	96.600
16	16	-11.167	0.418	104.047

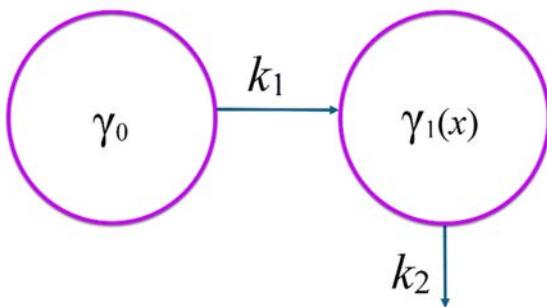
19	19	-14.491	-0.561	88.547
22	22	-13.121	-1.692	101.978
28	28	-12.637	-2.762	112.492
29	29	-12.118	-2.994	81.106
32	32	-14.804	-3.780	113.856
33	33	-9.209	-0.423	74.871
34	34	-10.970	-0.302	99.603
36	36	-11.868	-1.322	107.010

The third situation where a non-standard response surface design should be used is when the model is non-standard or nonlinear. Consider modeling the metabolism of tetracycline (a common antibiotic). **Figure 8-14** shows a diagram of the two-compartment model useful in pharmacokinetics for representing the metabolism of drugs.



**Figure 8-13. Comparison of Fraction of Design Space Plots for D-Optimal and I-Optimal Designs**

When the drug is taken orally,  $\gamma_0$  represents the initial concentration in the gastrointestinal (GI) tract.  $\gamma_1(x)$  represents the concentration in the blood at time  $x$ .  $k_1$  represents the rate constant at which the drug moves from the GI tract to the blood and  $k_2$  is the rate constant at which the drug is eliminated from the blood.



**Figure 8-14. Diagram of Two-Compartment Model for Tetracycline Metabolism**

Based on chemical kinetics, an equation relating the concentration in the blood at time  $x$  can be derived and is shown in **Eq. 8-7**

$$y = \gamma_1(x) = \gamma_0 [e^{-k_1(x-t_0)} - e^{-k_2(x-t_0)}] \quad \text{Eq. 8-7}$$

where  $t_0$  is the dead time.

This is an example of a **mechanistic model** that is derived from physical principles. To determine the parameters in the model ( $\gamma_0$ ,  $k_1$ ,  $k_2$ , and  $t_0$ ) a subject takes an oral dose of the drug, and small blood samples are taken at times  $x_i, i = 1, \dots, n$ . The response, or concentration of drug in the blood  $\gamma_1(x_i)$ , is determined at each sampling time or factor level and **Eq. 8-7** is fit to the data. Once fit, the mechanistic model can be used to predict response values outside the experimental range of  $x_i, i = 1, \dots, n$  since it is based on physical principles. This is not true for the general quadratic **Eq. 8-4** since it is just a Taylor's series approximation of the true equation about some point  $(x)$ .

In this problem there is only one independent variable and one response, and there are four parameters in the model. Since the model is nonlinear in the parameters, a simple design of four equally spaced times (to sample blood) would not be efficient for estimating the parameters. Instead, we can perform an  $l$ -optimal search to determine the design points. This is also a clear example of testing system capability when the system space does not include the entire battlespace.

The first step in creating a  $l$ -optimal design for a nonlinear model is to linearize the model about some point as described by Bates and Watts

(Bates & Watts, 2007).

For example, linearizing  $f(x, \gamma_0, k_1, k_2, t_0)$  about the point  $(\gamma_0^*, k_1^*, k_2^*, t_0^*)$

$$\begin{aligned}
 f(x, \gamma_0, k_1, k_2, t_0) \approx & f(\gamma_0^*, k_1^*, k_2^*, t_0^*) + (\gamma_0 - \gamma_0^*) \frac{\partial f}{\partial \gamma_0} \Big|_{\gamma_0=\gamma_0^*} \\
 & + (k_1 - k_1^*) \frac{\partial f}{\partial k_1} \Big|_{k_1=k_1^*} \\
 & + (k_2 - k_2^*) \frac{\partial f}{\partial k_2} \Big|_{k_2=k_2^*} \quad \text{Eq. 8-8} \\
 & + (t_0 - t_0^*) \frac{\partial f}{\partial t_0} \Big|_{t_0=t_0^*}
 \end{aligned}$$

which is linear in the variables  $\frac{\partial f}{\partial \gamma_0}$ ,  $\frac{\partial f}{\partial k_1}$ ,  $\frac{\partial f}{\partial k_2}$ , and  $\frac{\partial f}{\partial t_0}$  that are all functions of  $x$ . For the compartment model in [Eq. 8-7](#)

$$\begin{aligned}
 \frac{\partial f}{\partial \gamma_0} &= e^{-k_1(x-t_0)} - e^{-k_2(x-t_0)} \\
 \frac{\partial f}{\partial k_1} &= -\gamma_0(x-t_0)e^{-k_1(x-t_0)} \\
 \frac{\partial f}{\partial k_2} &= -\gamma_0(x-t_0)e^{-k_2(x-t_0)} \\
 \frac{\partial f}{\partial t_0} &= \gamma_0 k_1 e^{-k_1(x-t_0)} - \gamma_0 k_2 e^{-k_2(x-t_0)}
 \end{aligned}$$

The strategy is to create a grid of candidates in the independent variable  $x$ , calculate the values of the four partial derivatives using initial guesses of the parameter values at each candidate point, and then use the optFederov function to select a l-optimal subset of the grid. The R code below creates the grid and evaluates the partial derivatives using the initial parameter estimates  $\gamma_0^0 = 2.65$ ,  $k_1^0 = 0.15$ ,  $k_2^0 = 0.72$ , and  $t_0^0 = 0.41$ .

```

k1 <- .15; k2 <- .72; gamma0 <- 2.65; t0 <- 0.41
x <- c(seq(1:25))
dfdk1 <- c(rep(0, 25))
dfdk2 <- c(rep(0, 25))
dfdgamma0 <- c(rep(0, 25))
dfdt0 <- c(rep(0, 25))

for (i in 1:25) {
  dfdk1[i] <- -1 * gamma0 * exp(-k1 * (x[i] - t0))
  *(x[i] - t0)
  dfdk2[i] <- gamma0 * exp(-k2 * (x[i] - t0)) * (x[i] -
  t0)
  dfdgamma0[i] <- exp(-k1 * (x[i] - t0)) - exp( -k2 *
  (x[i] - t0))
  dfdt0[i] <- gamma0 * exp(-k1 * (x[i] - t0)) * k1 -
  gamma0 *
    exp(-k2 * (x[i] - t0)) * k2; }
grid <- data.frame(x, dfdk1, dfdk2, dfdgamma0, dfdt0)

```

In the code on the next page the `optFederov` function is used to create a design consisting of a subset of the grid.

```

library(AlgDesign)
desgn2 <- optFederov(~+dfdk1 + dfdk2 + dfdgamma0 +
dfdt0, data = grid, nTrials = 8, center = TRUE, riterion
= "D",
nRepeats=20)

```

Since there are  $p = 4$  parameters in the model, only four design points will be needed to estimate the parameters, therefore the option `nTrials = 4` is used in the call to the `optFederov` function. An experimenter can then replicate each design point as many times as necessary to get an estimate of experimental error. The code above results in the four distinct times for taking blood samples (1, 2, 5, and 25), and due to the form of the model and initial guesses of the parameters these points are far from equally spaced. Since the `optFederov` function may not always find the I-optimal design, and the fact that I-optimal designs may not be unique, running this code again may result in a slightly different design.

As stated by Bates and Watts (2007), the efficiency of the design for the

nonlinear model will depend on the stage at which a researcher is in the investigation. If the form of the equation is known but not the parameter values, the design created by the `optFederov` function will be called an initial design. After the data is collected from an initial design and the model is fit, the parameter estimates will be an improvement on the initial estimates. Therefore, the initial guesses of the parameter values can be replaced by the estimates and the initial design can be augmented with  $p = 4$  additional design points using the `augment = TRUE` option in the function call similar to the example shown in [Section 6.5.2](#).

## 8.6. Fitting the Response Surface Model with R

This section will describe how to fit a response surface model using R package `rsm` and a nonlinear mechanistic model using the R function `nls`.

### 8.6.1. Fitting a Linear Model and Checking for Curvature

If the factorial portion of a central composite design along with center points is completed in an initial block of experiments as described in [Section 8.3.1](#), the linear [Eq. 8-6](#) should be fit to the data and checked for adequacy. If the linear model is adequate, the axial points and additional center points in the central composite design will not be necessary. To check for adequacy the residual sums of squares from [Eq. 8-6](#) should be partitioned into the portion due to pure replication and the portion due to quadratic departure from the model. This can be easily done using the `rsm` function in the R package `rsm`. For example, the data frame `cement` in the `daewr` package contains the data in [Table 8-1](#). This data frame contains an additional variable `Block` not shown in [Table 8-1](#). `Block=1` for the first 11 runs (factorial + center points) in Table 10.1 and `Block=2` for the last nine runs (axial + center points).

In the R code below, the data is retrieved from the `daewr` package. This data frame was created with the `ccd` function in the package `rsm` therefore the factors are stored internally as the coded factors `x1`, `x2`, and `x3`.

```
library(daewr)
data(cement)
head(cement)
```

	Block	WatCem	BlackL	SNF	y
C1.1	1	0.33	0.12	0.08	109.5
C1.2	1	0.35	0.12	0.08	117.0
C1.3	1	0.33	0.18	0.08	110.5
C1.4	1	0.35	0.18	0.08	121.0
C1.5	1	0.33	0.12	0.12	120.0
C1.6	1	0.35	0.12	0.12	130.0

Data are stored in coded form using these coding formulas ...  
 $x_1 \sim (\text{WatCem} - 0.34)/0.01$   
 $x_2 \sim (\text{BlackL} - 0.15)/0.03$   
 $x_3 \sim (\text{SNF} - 0.1)/0.02$

An analysis is performed on the data in block 1 using R. The formula  $SO(x_1, x_2, x_3)$  in the call to the `rsm` function creates the full quadratic **Eq. 8-4**. However, when the axial points are left out of the central composite design (as they are in block 1), all the quadratic terms become confounded. The test on the  $PQ(x_1, x_2, x_3)$  term in the resulting ANOVA table is a test of the confounded quadratic terms.

```
library(rsm)
grout.lin <- rsm(y ~ SO(x1, x2, x3), data = cement,
                    subset = (Block == 1))
anova(grout.lin)
```

#### Analysis of Variance Table

```
Response: y
          Df Sum Sq Mean Sq F value    Pr(>F)
FO(x1, x2, x3)   3 465.13 155.042 80.3094 0.002307 ***
TWI(x1, x2, x3)  3   0.25   0.083  0.0432 0.985889
PQ(x1, x2, x3)   1  37.88  37.879 19.6207 0.021377 *
Residuals        3   5.79   1.931
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The sums of squares for Linear (labeled  $FO(x_1, x_2, x_3)$  in the output) account simultaneously for the three linear terms in the model. The sums of squares for cross product (labeled  $TWI(x_1, x_2, x_3)$  in the output) account simultaneously for the three interaction terms in the model. The

sums of squares for Quadratic (labeled  $PQ(x_1, x_2, x_3)$  in the output) accounts for the confounded quadratic terms or departure from linearity. Finally, the sums of squares for Residual represents the pure error sums of squares due to replicated center points. When fitting the linear model with interactions (*Eq. 8-6*) with the R function `lm` the sums of squares for Quadratic would be combined with the error sums of squares resulting in four degrees of freedom for error. The  $F$ -test on the quadratic term is a test of the adequacy of the linear model. Since it is significant at the  $\alpha = 0.05$  level in the above table, it indicates that the quadratic departure from the linear model is significant, and thus the linear model (*Eq. 8-6*) is not adequate for representing the first 11 data points in Table 10.1. Therefore, if the experiments shown in Table 10.1 were to be run in two blocks, it would be necessary to run the second block that includes the axial points.

### 8.6.2. Fitting the General Quadratic Model

The `rsm` function is most useful for fitting the general quadratic model and analyzing the fitted surface. For example, consider fitting the general quadratic model to the trebuchet data in *Table 8-3*. The commands to retrieve the data frame from the `daewr` package are shown below.

```
library(daewr)
data(Treb)
head(Treb)
```

```
A  B  C  y
1 4 10 2.5 33
2 8 10 2.5 85
3 4 20 2.5 86
4 8 20 2.5 113
5 4 15 2.0 75
6 8 15 2.0 105
```

```
Data are stored in coded form using these coding formulas ...
x1 ~ (A - 6)/2
x2 ~ (B - 15)/5
x3 ~ (C - 2.5)/0.5
```

The code to fit the quadratic model and a portion of the output are shown below.

```
> library(rsm)
> treb.quad <- rsm(y ~ S0(x1, x2, x3), data = Treb)
> summary(treb.quad)
```

Call:

rsm(formula = y ~ S0(x1, x2, x3), data = Treb)

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	90.00000	1.16905	76.9859	7.006e-09 ***
x1	19.75000	0.71589	27.5880	1.171e-06 ***
x2	19.75000	0.71589	27.5880	1.171e-06 ***
x3	-11.50000	0.71589	-16.0639	1.703e-05 ***
x1:x2	-6.25000	1.01242	-6.1733	0.0016247 **
x1:x3	4.75000	1.01242	4.6917	0.0053768 **
x2:x3	6.75000	1.01242	6.6672	0.0011461 **
x1^2	-9.37500	1.05376	-8.8967	0.0002986 ***
x2^2	-1.37500	1.05376	-1.3048	0.2487686
x3^2	-3.37500	1.05376	-3.2028	0.0239200 *
---				
Signif. codes:	0 ***	0.001 **	0.01 *	0.05 . 0.1 ' '

Multiple R-squared: 0.9975, Adjusted R-squared: 0.9929

F-statistic: 218.9 on 9 and 5 DF, p-value: 5.964e-06

#### Analysis of Variance Table

Response: y

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
F0(x1, x2, x3)	3	7299.0	2433.00	593.4146	8.448e-07
TWI(x1, x2, x3)	3	428.8	142.92	34.8577	0.0008912
PQ(x1, x2, x3)	3	351.5	117.16	28.5759	0.0014236
Residuals	5	20.5	4.10		
Lack of fit	3	14.5	4.83	1.6111	0.4051312
Pure error	2	6.0	3.00		

Stationary point of response surface:

x1	x2	x3
0.9236846	-1.7161183	-2.7698217

Stationary point in original units:

A	B	C
7.847369	6.419409	1.115089

Eigenanalysis:

eigen() decomposition

\$values

```
[1] 1.280298 -3.551452 -11.853845
```

```
$vectors
[,1]      [,2]      [,3]
x1 -0.1236692 0.5238084 0.8428112
x2 0.8323200 -0.4077092 0.3755217
x3 0.5403233 0.7479291 -0.3855551
```

The first table in the output shows the least squares estimates of the coefficients of the coded factors in *Eq. 8-4* along with their standard errors, *t*-values, and *p*-values. Some statisticians advocate dropping the insignificant terms from the model and refitting, but this is not possible with the `rsm` function. Other statisticians advocate retaining all terms in the general quadratic model, whether significant or not, since the individual model terms have no physical meaning. A compromise is to eliminate a model factor if all linear, quadratic, and interaction terms involving that factor are insignificant. By eliminating a factor the response surface model is reduced from *p* dimensions to *p* – 1 dimensions. In this example all linear and interaction terms are significant, so no factors can be eliminated. If a factor had been found to be insignificant, it could be eliminated from the `SO(x1, x2, x3)` formula for the quadratic model statement and the response surface equation would simplify from three dimensions to two.

The second table shows the ANOVA table from the fit. In this example, it can be seen that the linear quadratic and cross product terms in the model are all significant. The error sums of squares are partitioned into the pure error sums of squares and the lack of fit sums of squares. By doing this, a simple *F*-test can be made for the adequacy of the quadratic model. For example, the responses at the three center points in Table 10.3 are 88, 91, and 91. Therefore, the pure sums of squares due to replication are:

$$\sum_{i=1}^3 y_i^2 - \frac{(\sum_{i=1}^3 y_i)^2}{3} = 88^2 + 91^2 + 91^2 - \frac{(88 + 91 + 91)^2}{3} = 6.0$$

The difference in the total error sums of squares (20.5) and the pure error sums of squares is called the lack of fit sums of squares. An *F*-test of the lack of fit sums of squares is a test of the adequacy of the quadratic

model. In this case it is clearly insignificant, indicating that predictions made from the general quadratic model for this experiment can be considered just as accurate as running additional experiments, as long as no lurking variables change before additional experiments can be run.

When the lack of fit test is significant, it indicates that the general quadratic model is not adequate for prediction. This could be due to the fact that the experimental region is so large that the quadratic model does not provide a good approximation to the true response function over the entire region, or due to outliers, or extreme nonlinearity in certain corners of the experimental region where the approximate function does not fit. These conditions can be detected by making residual plots to check the least squares assumptions or by fitting the model using a “robust” technique such as M-estimators, for details and discussion see Lawson (1982).

Block terms can also be included when fitting the quadratic model with the `rsm` function. For example, the experiments on cement grout shown in **Table 8-1** could be completed in two blocks. The first block would consist of the factorial plus center points shown in runs 1–11 and the second block would consist of the axial and center points shown in runs 12–20. The R code to retrieve the data and fit the model are shown below.

```
library(daewr)
data(cement)
grout.quad <- rsm(y ~ Block + S0(x1,x2,x3),
                    data = cement)
summary(grout.quad)
```

Call:

```
rsm(formula = y ~ Block + S0(x1, x2, x3), data = cement)
```

	Estimate	Std. Error	t value	Pr(> t )	
(Intercept)	1.1628e+02	1.0691e+00	108.7658	2.383e-15	***
Block2	4.4393e-01	1.0203e+00	0.4351	0.67375	
x1	5.4068e+00	6.1057e-01	8.8553	9.746e-06	***
x2	9.2860e-01	6.1057e-01	1.5209	0.16262	
x3	4.9925e+00	6.1057e-01	8.1767	1.858e-05	***
x1:x2	1.2500e-01	7.9775e-01	0.1567	0.87895	
x1:x3	-1.4189e-14	7.9775e-01	0.0000	1.00000	

```

x2:x3      1.2500e-01  7.9775e-01  0.1567  0.87895
x1^2       1.4135e+00  5.9582e-01  2.3723  0.04175 *
x2^2       1.3251e+00  5.9582e-01  2.2240  0.05322 .
x3^2       1.5019e+00  5.9582e-01  2.5207  0.03273 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

```

Multiple R-squared: 0.9473, Adjusted R-squared: 0.8887  
F-statistic: 16.17 on 10 and 9 DF, p-value: 0.0001414  
Analysis of Variance Table

Response: y

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Block	1	0.00	0.003	0.0006	0.98068
F0(x1, x2, x3)	3	751.41	250.471	49.1962	6.607e-06
TWI(x1, x2, x3)	3	0.25	0.083	0.0164	0.99693
PQ(x1, x2, x3)	3	71.45	23.817	4.6779	0.03106
Residuals	9	45.82	5.091		
Lack of fit	5	42.49	8.498	10.1972	0.02149
Pure error	4	3.33	0.833		

Stationary point of response surface:

x1	x2	x3
-1.9045158	-0.1825251	-1.6544845

Stationary point in original units:

WatCem	BlackL	SNF
0.32095484	0.14452425	0.06691031

Eigenanalysis:

```

eigen() decomposition
$values
[1] 1.525478 1.436349 1.278634

```

\$vectors

[,1]	[,2]	[,3]
x1 0.1934409	0.8924556	0.4075580
x2 0.3466186	0.3264506	-0.8793666
x3 0.9178432	-0.3113726	0.2461928

The output will be similar to the last example except a block term with one degree of freedom will be included in both tables. The block factor will then account for any average differences between the first and second group of experiments.

### 8.6.3. Fitting a Nonlinear Mechanistic Model

The general quadratic model is linear in the coefficients ( $\beta$ 's), and the  $X'X$  is full rank so the least square estimates are obtained by solving the normal equations through straightforward matrix inversion. When the model is non-linear like the two-compartment model shown in [Eq. 8-7](#), the least squares estimates are more difficult to obtain and must be found by iterative numerical techniques. However, the R function `nls` makes this all transparent to the user.

**Table 8-7. Tetracycline Concentration in Plasma over Time**

Time (hr)	Tetracyclin Conc. ( $\mu\text{g/ml}$ )
1	0.7
2	1.2
3	1.4
4	1.4
6	1.1
8	0.8
10	0.6
12	0.5
16	0.3

For example, to fit the two-compartment model [Eq. 8-7](#) to the data from Wagner (Wagner, 1967) shown in **Table 8-7**, use the following code.

```
library(daewr)
data(Tet)
mod.nln1 <- nls(Conc ~ gamma0 * (exp( -k1 * (Time - t0))
-
exp( -k2 * (Time - t0))), data = Tet,
start = list(gamma0 = 10, k1 = .12, k2 = .5, t0 = .5))
summary(mod.nln1)
```

In this code the data in **Table 8-6** is recalled from the data frame in the daewr package. The formula given in the call to the `nls` function is [Eq. 8-7](#), where Time in the data frame Tet is  $x$  in Equation (10.7). The option `start = list( gamma0=10, k1 = .12, k2 = .5, t0 = .5)` supplies initial estimates of the parameters from which the iterative numerical procedure will begin. The convergence of the numerical procedure can

be sensitive to the initial estimates, and Bates and Watts (2007) give five strategies for obtaining good initial estimates. The `nls` function by default (other options are available) uses the Gauss-Newton method with numerical derivatives to find the least squares estimates.

The resulting output is shown below. The table gives the least squares parameter estimates, their standard errors, t-statistics, and P-values.

```
Formula: Conc ~ gamma0 * (exp(-k1 * (Time - t0)) - exp(-k2 *  
(Time - t0)))
```

Parameters:

	Estimate	Std. Error	t value	Pr(> t )
gamma0	2.64964	0.36446	7.270	0.000770 ***
k1	0.14880	0.01441	10.327	0.000146 ***
k2	0.71575	0.12605	5.678	0.002359 **
t0	0.41224	0.09495	4.342	0.007416 **
---				

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.04482 on 5 degrees of freedom

Number of iterations to convergence: 11

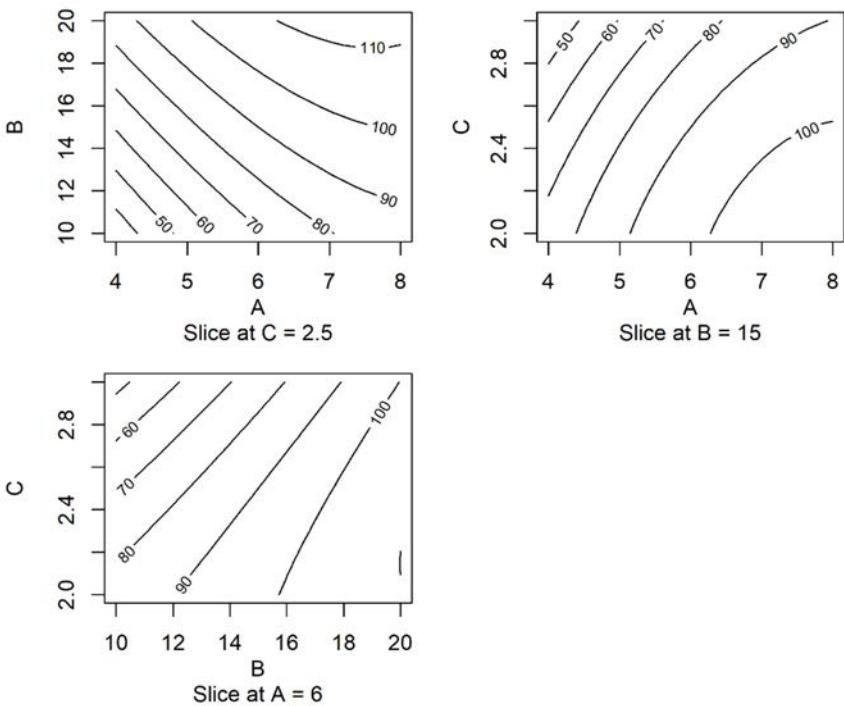
Achieved convergence tolerance: 3.582e-06

## 8.7. Determining Optimum Operating Conditions

### 8.7.1. Contour Plots

Once a response surface model has been fit to the data from an experiment, there are several ways to identify the optimum conditions. If there are only two factors, the simplest method is to make a contour plot or a three-dimensional plot of the response surface. If there are more than two factors in the model, several two-dimensional slices through the experimental region can be made by holding some factors constant and making a contour or 3D plots with respect to the remaining two factors at each slice.

The `rsm` automatically generates a panel of contour plots. The commands below show that by simply adding the `contour(treb.quad, ~ x1+x2+x3)` statement, the panel of contour plots shown in Figure 10.15 is created.

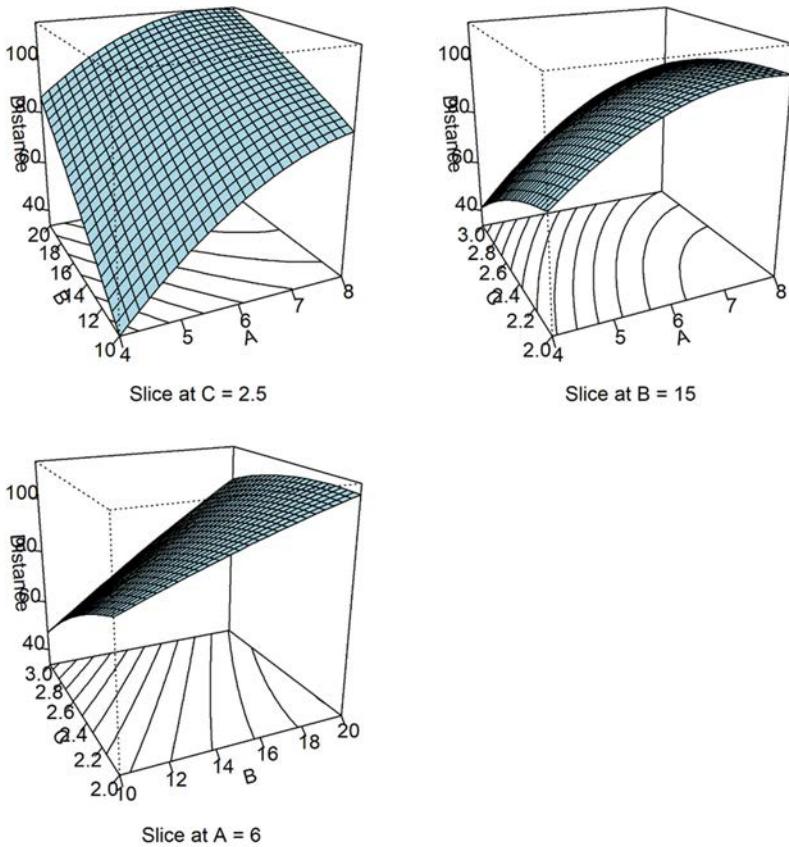


**Figure 8-15. Contour Plots of Predicted Trebuchet Distance**

In this panel of plots, the first plot (upper left) is made at a slice with C: missile weight held constant at its mid-value of 2.5, and factors A: arm length and B: counter weight on the axis. The lines represent contours of predicted values from the quadratic model. The remaining two plots show factors A and C on the axis with factor B held constant at its mid-value of 15, and factors B and C on the axis with A held constant at its mid-value of 6.

By calling the `persp` graphic function rather than the `contour`, as shown below, the panel of three-dimensional surface plots shown in **Figure 8-16** is created. The option `contours=list(z="bottom")` includes the same contour lines as shown in **Figure 8-15**, below the 3D surface.

```
par (mfrow=c(2,2))
persp(treb.quad, ~ x1+x2+x3, zlab="Distance",
contours=list(z="bottom") )
par (mfrow=c(1,1))
```



**Figure 8-16. Three-Dimensional Surface Plots of Predicted Trebuchet Distance**

We can also create individual contour or 3D plots and we can specify the constant values of the factors on the axis as shown in the code below.

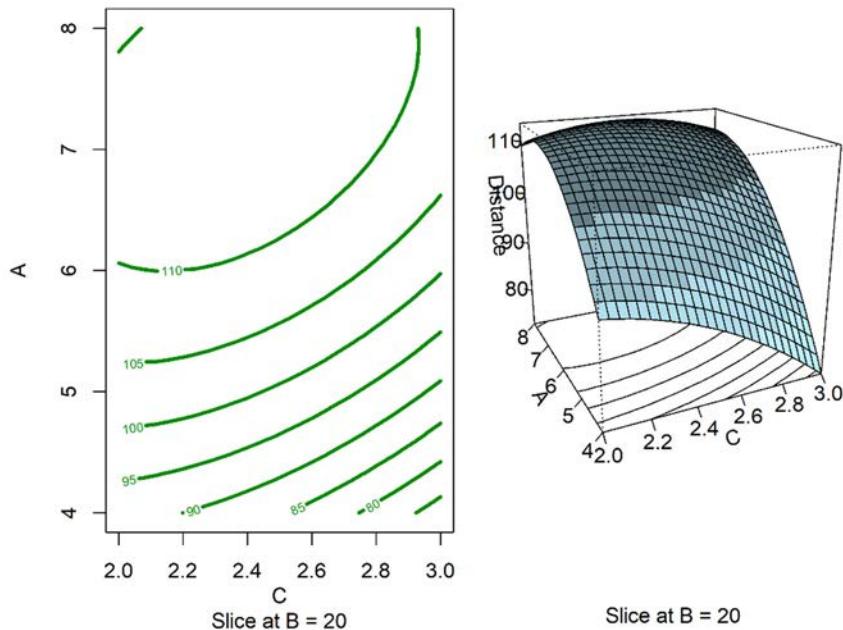
```
par (mfrow=c(1,2))
contour(treb.quad, x1~x3, at=list(x2=1))
persp(treb.quad, x1~x3, at=list(x2=1), zlab="Distance",
contours=list(z="bottom"))
par (mfrow=c(1,1))
```

### 8.7.2. Canonical Analysis

When the response surface is a hilltop or a valley with a distinct maximum or minimum within the experimental region, the exact factor coordinates of the maximum or minimum can be determined by

simultaneously setting the derivatives of the fitted equation with respect to each factor equal to zero and solving the simultaneous set of equations. This is useful when there are more than two factors, and the maximum or minimum would be difficult to identify with multiple contour plots. The vector solution (of factor settings) to the simultaneous set of homogeneous equations can be expressed in matrix terms

as



**Figure 8-17. Contour and 3D Plot of Predicted Trebuchet Distance with Counterweight=20 lb**

$$x_0 = -\hat{B}^{-1}\hat{b}/2 \quad \text{Eq. 8-9}$$

where  $\hat{B}$  and  $\hat{b}$  are the least squares estimates of the matrix and vector of regression coefficients defined in [Eq. 8-5](#). This solution is actually called the stationary point because it could be a maximum, a minimum, or a saddle point as shown in [Figure 8-1](#). To determine what the solution is, it is useful to express the response surface [Eq. 8-5](#) in a canonical form with the origin translated to  $x_0$  and the axis rotated (as shown in [Figure 8-18](#)).

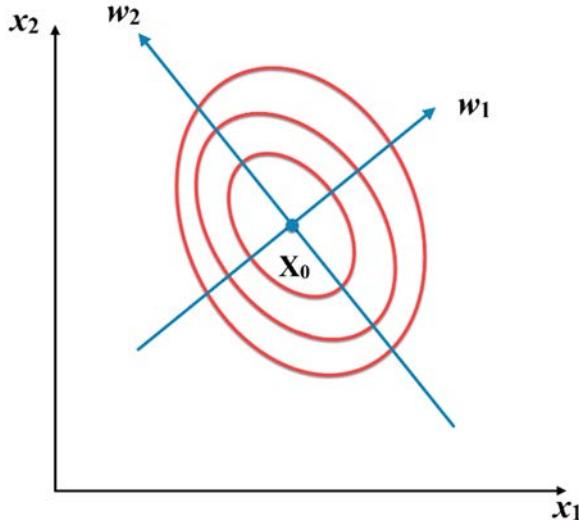
Letting  $z = x - x_0$ , **Eq. 8-5** can be written as

$$\mathbf{y} = \mathbf{y}_0 + z' \mathbf{B} z \quad \text{Eq. 8-10}$$

where  $y_0 = x_0 b + x_0' B x_0$ . This translates the origin to  $x_0$ . Through the or-thogonal rotation  $z = Mw$ , the second order part of **Eq. 8-10** can be reduced to a linear combination of squares of the rotated variables  $w_i$  given by

$$z' \mathbf{B} z = w' M' B M w = \lambda_1 w_1^2 + \lambda_2 w_2^2 + \dots + \lambda_k w_k^2 \quad \text{Eq. 8-11}$$

If all the coefficients  $\lambda_i$  are negative, it indicates the stationary point is a



**Figure 8-18. Representation of Canonical System with Translated Origin and Rotated Axis**

maximum. If all the coefficients are positive, it indicates the stationary point is a minimum, and if there are mixed signs of the coefficients it indicates the stationary point is a saddle point. The matrix  $M$  is the matrix containing the eigenvectors of  $B$  as the columns, and the coefficients  $\lambda_i$  are the eigenvalues of  $B$ .

The `rsm` function automatically calculates the eigenvalues and eigenvectors of the least squares estimates of  $\hat{B}$ . For example, for the cement grout experiment shown in **Table 8-1**, the code below produces

the following table of results after the output summarizing the fitted model.

```
library(daewr)
data(cement)
grout.quad <- rsm(y ~ Block + S0(x1,x2,x3), data =
cement)
summary(grout.quad)
```

Stationary point of response surface:

x1	x2	x3
-1.9045158	-0.1825251	-1.6544845

Stationary point in original units:

WatCem	BlackL	SNF
0.32095484	0.14452425	0.06691031

Eigenanalysis:

```
eigen() decomposition
$values
[1] 1.525478 1.436349 1.278634
```

```
$vectors
[,1]      [,2]      [,3]
x1 0.1934409  0.8924556  0.4075580
x2 0.3466186  0.3264506 -0.8793666
x3 0.9178432 -0.3113726  0.2461928
```

The coordinates of the stationary point are given in both coded and uncoded units. Next the eigenvalues and eigenvectors of  $\hat{B}$  are listed. Since all the eigenvalues are positive, the stationary point is a minimum. Contour plots and 3D surface plots can be made holding factors not on the axis constant at their stationary point values, rather than their center values, by changing the code on page 450 to `contour(treb.quad, ~ x1+x2+x3, at = xs(treb.quad))`.

For this particular problem, knowing the factor settings to achieve a mini-mum response is not useful. The goal of experimentation was to find the factor combinations that would maximize the workability of the cement grout. In general, when seeking a maximum or minimum response within the experimental region, the stationary point will only be useful if (1) the stationary point is within the experimental region, and (2) the stationary point is of the type sought (i.e., maximum or

minimum). If the stationary response is outside the experimental region, or if it is a saddle point or minimum when seeking the maximum, another method must be used to identify the optimum.

### 8.7.3. Ridge Analysis

Another method of finding the optimum within the experimental region is to use ridge analysis (Hoerl, 1959) (1963). This method seeks to find the maximum or minimum of  $y = \mathbf{xb} + \mathbf{x}'\mathbf{Bx}$  subject to the constraint that  $\mathbf{x}'\mathbf{x} = R^2$ , or that the coordinates of the optimum are on a radius  $R$  from the origin in coded units. The solution is obtained in a reverse order using Lagrange multipliers. The resulting optimal coordinates are found to be the solution to the equation

$$(\mathbf{B} - \mu \mathbf{I}_k) \mathbf{x} = -\frac{\mathbf{b}}{2} \quad \text{Eq. 8-12}$$

To solve this system, choose a value for  $\mu$ , insert it into [Eq. 8-12](#), and solve for the vector of coded factor settings  $x$ . Once »the coded factor settings have been found, the radius of the solution is  $R = \sqrt{\sum x_i^2}$ .

Inserting values of  $\mu$  larger than the largest eigenvalue of the matrix  $B$  will result in a solution for the maximum response on the radius  $R$ , and inserting values of  $\mu$  smaller than the smallest eigenvalue of the matrix  $B$  will result in a solution for the minimum response on the radius  $R$ . Trial and error are required to find the solution on a specific radius. However, the steepest function in the rsm package does all the work.

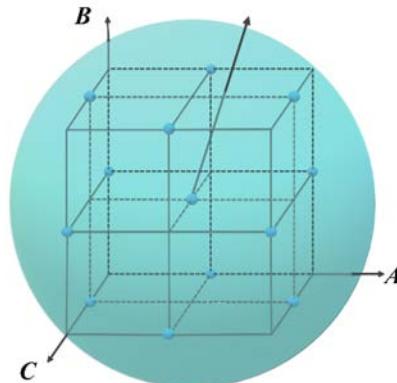
As an example, consider again finding the maximum distance the trebuchet can toss an object within the experimental ranges defined in [Table 8-3](#). This was done approximately using a contour plot in [Section 8.7.1](#). The stationary point analysis, printed by the `rsm` function for this data, showed the stationary point was a saddle point that was outside the experimental region. Therefore, the maximum within the experimental region has to be on the extreme boundary of the experimental region. Since the experimental region for a Box-Behnken design (that was used for this experiment) is a sphere with radius  $\sqrt{2} = 1.412$  in coded units, the call to the steepest function in the code shown

below gives rise to the output list shown below the code.

```
> ridge<-steepest(treb.quad, dist=seq(0, 1.412, by=.1),  
descent=FALSE)  
> ridge
```

ridgedist	x1	x2	x3	A	B	C	yhat
0	0	0	0	6	15	2.5	90
0.1	0.064	0.067	-0.037	6.128	15.335	2.4815	92.909
0.2	0.124	0.139	-0.073	6.248	15.695	2.4635	95.626
0.3	0.18	0.215	-0.105	6.36	16.075	2.4475	98.12
0.4	0.232	0.297	-0.134	6.464	16.485	2.433	100.455
0.5	0.277	0.385	-0.158	6.554	16.925	2.421	102.599
0.6	0.315	0.48	-0.175	6.63	17.4	2.4125	104.59
0.7	0.345	0.58	-0.185	6.69	17.9	2.4075	106.424
0.8	0.368	0.686	-0.185	6.736	18.43	2.4075	108.154
0.9	0.384	0.795	-0.177	6.768	18.975	2.4115	109.783
1	0.393	0.905	-0.161	6.786	19.525	2.4195	111.318
1.1	0.397	1.017	-0.137	6.794	20.085	2.4315	112.817
1.2	0.398	1.127	-0.107	6.796	20.635	2.4465	114.259
1.3	0.395	1.236	-0.073	6.79	21.18	2.4635	115.673
1.4	0.39	1.344	-0.034	6.78	21.72	2.483	117.077

It can be seen that the predicted maximum on a radius R increases as R increases along the path shown in **Figure 8-19**, and that it finally reaches a maximum of 117.077 feet at the boundary of the experimental region (1.412 in coded units).



**Figure 8-19. Path of Maximum Ridge Response through Experimental Region**

At that point the factor levels in uncoded units are shown to be approximately A: arm length = 6.8 inches; B: counterweight = 21.8 lb;

and  $C$ : missile weight = 2.5 oz. A graph of the predicted response and factor coordinates as a function of the distance from the origin (in coded units) as shown in Figure 10.20. This is a common way to present and summarize the results of a ridge analysis.

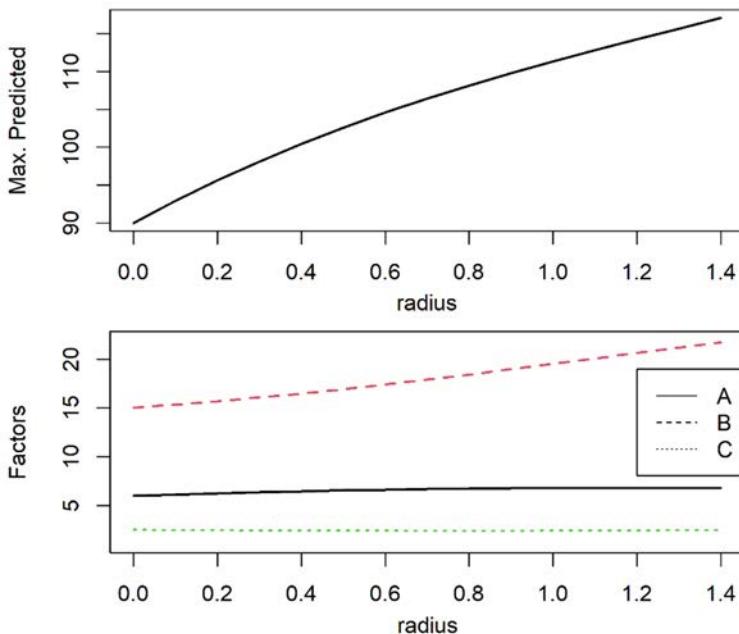


Figure 8-20. Factor Levels to Achieve Maximum Predicted Value

#### 8.7.4. Nonlinear Optimization

Canonical analysis and ridge analysis work well for finding the optimum of a response surface if the model is the general quadratic model and the experimental region is spherical in coded units. If the response surface equation is not a general quadratic model or the experimental region is irregular, the exact optimum can always be found using more general numerical methods. The R `constrOptim` function can minimize a function subject to linear inequality constraints using an adaptive barrier algorithm, and it can be used to quickly find a numerical optimum.

As an example of using `constrOptim` to find a response surface optimum, consider finding the maximum of a nonlinear mechanistic response surface. In a chemical reaction the reactant is transformed to

the product at rate constant.



## References

---

- Addelman, S., & Kempthorne, O. (1979). *Orthogonal main-effect plans*. Aeronautical Research Laboratory Technical Report.
- AlmeidaeSilva, J. B., Lima, U. A., Taqueda, M. E., & Guaragna, F. G. (2003). Use of response surface methodology for selection of nutrient levels for culturing Paecilomyces variotii in eucalyptus hemicellulosic hydrolyzate. *Biosource Technology*, 87, 45-50.
- Amerise, I. L., & Tarsitano, A. (2015). Correction methods for ties in rank correlations. *Journal of Applied Statistics*, 42(12), 2584-2596. doi:10.1080/02664763.2015.1043870
- Anderson, M. (2003). Messing with medieval missile machines. *Stateaser, Sept.* Retrieved from [www.statease.com](http://www.statease.com).
- Apple, D. (2006). *Dish detergent experiment*. BYU Dept. of Statistics.
- Atkinson, A. C., Donev, A. N., & Tobias, R. D. (2007). *Optimum Experimental Designs with SAS*. Oxford : Oxford University Press.
- Bainbridge, T. R. (1965). Staggered nested designs for estimating variance. *Industrial Quality Control*, 22, 12-20.
- Bankes, S. (1993). Exploratory Modeling for Policy Analysis. *Operations Research*, 41(3), 435-449. Retrieved from <https://www.jstor.org/stable/171847>
- Barrios, E. (2009). BsMD: Bayes Screening and Model Discrimination. *R package version 0.6-5.2*.
- Bastable, A. (2006). Tee it high or low. *Golf Magazine, June*, 173-179.
- Bates, D. M., & Watts, D. G. (2007). *Nonlinear Regression and Its Applications*. Hoboken, N.J.: John Wiley & Sons.
- Bates, D. M., & Watts, D. G. (2007). *Nonlinear Regression and Its Applications*. Hoboken, N.J.: John Wiley & Sons.

- Bechhofer, R. E., & Tamhane, A. C. (1981). Incomplete block designs for comparing treatments with a control: General theory. *Technometrics*, 23, 45-57.
- Bennett, C. A., & Franklin, N. L. (1954). *Statistical Analysis in Chemistry and the Chemical Industry*. New York: John Wiley & Sons.
- Bergstra, J., & Bengio, Y. (2012). Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, 13, 281–305.
- Berquist, B., Vanhatalo, E., & Nordenvaad, M. L. (2011). A bayesian analysis of unreplicated two-level factorials using effects sparsity, hierarchy, and heredity. *Quality Engineering*, 23, 152–166.
- Bisgaard, S. (1999). Quality quandries-proposals: A mechanism for achieving better experiments. *Quality Engineering*, 11, 645-649.
- Bose, R. C., Clatworthy, W. H., & Shrikhande, S. S. (1954). *Tables of partially balanced designs with two associate classes*. Technical Bulletin 107. North Carolina Agricultural Experiment Station.
- Box, G. E. (1957). Evolutionary operation: A method for increasing indus-. *Applied Statistics*, 6, 81-101.
- Box, G. E. (1988). Signal-to-Noise Ratios, Performance Criteria, and Transformation. *Technometrics*, 30, 1-40.
- Box, G. E. (1991). George's column: Finding bad values in factorial designs. *Quality Engineering*, 3, 249-254.
- Box, G. E. (1999). Statistics as a Catalyst to Learning by Scientific Method Part II—A Discussion. *Journal of Quality Technology*, 31, 16-29.
- Box, G. E., & Behnken, D. W. (1960). Some new three level designs for the study of quantitative variables. *Technometrics*, 2, 455-460.
- Box, G. E., & Cox, D. R. (1964). An analysis of transformations. *Journal of the Royal Statistical Society, Series B*, 26, 15-20.

- Box, G. E., & Hunter, J. S. (1957). Multifactor experimental designs for exploring response surfaces. *Annals of Mathematical Statistics*, 28, 195-241.
- Box, G. E., & Hunter, J. S. (1961). The  $2k-p$  fractional factorial designs. *Technometrics*, 3, 311-351 and 449-458.
- Box, G. E., & Meyer, R. D. (1986a). An analysis for unreplicated fractional factorials. *Technometrics*, 28, 11-18.
- Box, G. E., & Meyer, R. D. (1986b). Dispersion effects from fractional designs. *Technometrics*, 28, 19-27.
- Box, G. E., & Wilson, K. B. (1951). On experimental attainment of optimum conditions. *Journal of the Royal Statistical Society B*, 13, 1-45.
- Box, G. E., & Wilson, K. G. (1951). On the Experimental Attainment of Optimum Conditions. *Journal of the Royal Statistical Society*, 13, 1-45.
- Box, G. E., Bisgaard, S., & Fung, C. D. (1988). An Explanation and Critique of Taguchi's Contributions to Quality Engineering. *Quality and Reliability Engineering International*, 4, 123–131.
- Box, G. E., Hunter, W. G., & Hunter, J. S. (1978). *Statistics for Experimenters*. New York: John Wiley & Sons.
- Box, J. F. (1978). *R. A. Fisher: The Life of a Scientist*. New York: Wiley.
- Burdick, R., & Graybill, F. (1992). *Confidence Intervals on Variance Components*. New York: Marcel Dekker, Inc.
- Butler, N. A. (2006). Optimal blocking of two-level factorial designs. *Biometrika*, 93, 289–302.
- Casella, G. (2008). *Statistical Design*. New York: Springer.
- Chen, J., Sun, D. X., & Wu, C. F. (1993). A catalogue of two-level and three-level fractional factorial designs with small runs. *International Statistical Review*, 61, 131–145.

- Cochran, W. G., & Cox, G. M. (1950). *Experimental Designs*. New York: John Wiley.
- Cook, R. D., & Nachtsheim, C. J. (1989). Computer-aided blocking of factorial and response surface designs. *Technometrics*, 31, 339–346.
- Daniel, C. (1959). Use of half-normal plots in interpreting factorial two-level experiments. *Technometrics*, 1, 311-341.
- Daniel, C. (1960). Locating outliers in factorial experiments. *Technometrics*, 2, 149-156.
- Daniel, C. (1976). *Applications of Statistics to Industrial Experimentation* (First ed.). New York: John Wiley & Sons.
- Das, M. N. (1960). Fractional replicates as asymmetrical factorial designs. *Journal of the Indian Society of Agricultural Statistics*, 12, 159{174.
- Davies, O. L. (1949). *Statistical Methods in Research and Production*. Edinburg: Oliver and Boyd.
- de Mendiburu, F. (2012). agricolae: Statistical Procedures for Agricultural. *R package version 1.1-2*.
- de Mendiburu, F. (2012). agricolae: Statistical Procedures for Agricultural. *R package version 1.1-2*.
- Derringer, G. C. (1974). An empirical model for viscosity of filled and plasticized elastomer compounds. *Journal of Applied Polymer Science*, 18, 1083-1101.
- Dey, A. (1985). *Orthogonal Fractional Factorial Designs*. New Delhi: Halsted Press.
- DOD IG. (2013). *Advanced Combat Helmet Technical Assessment*. Alexandria, VA: Department of Defense Inspector General.
- Dossett, T., Le, R., & Moore, P. (2007). Apple browning experiment. BYU Dept. of Statistics.

- Draper, N. R. (1963). Ridge analysis of response surfaces. *Technometrics*, 5, 469-479.
- Draper, N. R. (1985). Small composite designs. *Technometrics*, 27, 173-180.
- Draper, N. R., & Lin, D. K. (1990). Small response surface designs. *Technometrics*, 32, 187-194.
- Dunnett, C. W. (1955). A multiple comparison procedure for comparing several treatments with a control. *Journal of the American Statistical Association*, 509, 1096-1121.
- Dykstra, O. (1971). The augmentation of experimental data to maximize  $|X'X|$ . *Technometrics*, 13, 682-688.
- Easterling, R. G. (2004). Teaching experimental design. *The American Statistician*, 244-252.
- Emmerling, A., Petricevic, R., Beck, A., Wang, P., Scheller, H., & Fricke, J. (1995). Relationship between optical transparency and nanostructural features of silica aerogels. *Journal of Non-Crystalline Solids*, 185(3), 240-248.
- Fairchild, K. P. (2011). *Screening designs that minimize model dependence*. Salt Lake City: Brigham Young University.
- Fannin, T. E., Marcus, M. D., Anderson, D. A., & Bergman, H. L. (1981). Use of a fractional factorial design to evaluate interactions of environmental factors affecting biodegradation rates. *Applied and Environmental Microbiology*, 42, 936–943.
- Faraway, J. J. (2015). *Linear Models with R*. Boca Raton, FL: CRC Press.
- Federov, V. V. (1972). *Theory of optimal experiments*. New York: Academic Press.
- Festing, M. F. (2003). Principles: The need for better experimental design. *Trends Pharmacol Sci*, 24, 341-345.
- Fisher, R. A. (1926). The arrangement of field experiments. *Journal of the Ministry of Agriculture of Great Britain*, 503{513}.

- Fisher, R. A. (1935). *The Design of Experiments*. New York: Collier MacMillan Publishers.
- Fisher, R. A. (1940). An examination of possible different solutions of a problem in incomplete blocks. *Annals of Eugenics*, 9, 353-400.
- Fisher, R. A. (1958). *Statistical Methods for Research Workers* (13 th ed.). Edinburgh.: Oliver and Boyd.
- Fisher, R. A. (1966). *The Design of Experiments* (8 ed.). New York: Hafner Publishing Company.
- Fisher, R. A. (n.d.). An examination of possible different problems in incomplete blocks. 9, 353-400.
- Fox, J. (2005). The R commander: A basic-statistics graphical user interface to R. *Journal of Statistical Software*, 14, 1-42.
- Fox, J., & Weisberg, S. (2011). *An R Companion to Applied Regression*. (Second ed.). Thousand Oaks, CA: Sage.
- Fricke, J. (1988). Aerogels — highly tenuous solids with fascinating properties. *Journal of Non-Crystalline Solids*, 100(1), 169-173.
- Genz, A., Bretz, F., Miwa, T. M., Leisch, F., Scheipl, F., & Bornkamp, B. (2012). mvtnorm: Multivariate Normal and t Distributions. *R Package*.
- Giovannitti-Jensen, A., & Myers, R. H. (1989). Graphical assessment of the prediction capability of response surface designs. *Technometrics*, 331, 159-171.
- Groemping, U. (2011a). FrF2: Fractional Factorial designs with 2-level factors. *R package version 1.2-8*.
- Groemping, U. (2011b). RcmdrPlugin.DoE: R Commander Plugin for (industrial) Design of Experiments. *R package version 0.11-5*.
- Groemping, U. (2012). DoE.base: Full factorials, orthogonal arrays and base utilities for DoE packages. *R package version 0.23-2*.

- Hall, M. (1961). Hadamard matrix of order 16. *Jet Propulsion Laboratory Research Summary*, 1, 21–26.
- Hamada, M., & Wu, C. F. (1992). Analysis of designed experiments with complex aliasing. *Journal of Quality Technology*, 24, 130–137.
- Hare, L. B. (1988). In the soup: A case study to identify contributors to variability. *Journal of Quality Technology*, 20, 36-43.
- Hartley, H. O. (1959). Smallest composite design for quadratic response surfaces. *Biometrics*, 15, 611-624.
- Hartley, H. O. (1967). Expectations, variances and covariances of ANOVA mean squares by synthesis. *Biometrics*, 34, 105-114.
- Heffner, T. G., Drawbaugh, R. B., & Zigmond, M. J. (1974). Amphetamine and operant behavior in rats: Relationship between drug effect and control response rate. *Journal of Comparative and Physiological Psychology*, 86, 1031-1043.
- Henderson, L. O., Hannon, W., Smith, S. J., & Cooper, G. R. (1987). An international collaborative study on standardization of apolipoproteins a-i and b. part ii. evaluation of contributions of antisers to among-laboratory. *Clinical Chemistry*, 33, 2250-2256.
- Hinkelmann, K., & Kempthorne, O. (1994). *Design and Analysis of Experiments Volume I*. New York: John Wiley & Sons.
- Hoerl, A. E. (1959). Optimum solution of many variables equations. *Chemical Engineering Progress*, 55, 69-78.
- Hothorn, T., Bretz, F., & Westfall, P. (2008). Simultaneous inference in general parametric models. *Biometrical Journal*, 50(3), 346-363.
- Hsu, J. C. (1984). Constrained simultaneous confidence intervals for multiple comparisons with the best. *Annals of Statistics*, 12, 1136-1144.

- Hunter, G. B., Hodi, F. S., & Eager, T. W. (1982). High-cycle fatigue of weld repaired cast t-6a1-4v. *Metallurgical Transactions,, 13A*, 1589–1594.
- Hunter, J. S. (1983). Let's all beware the latin square. *Quality Engineering, 1*, 195-198.
- Hunter, J. S. (1985). Statistical Design Applied to Product Design. *Journal of Quality Technology, 17*, 210–221.
- Hunter, J. S. (1989). Let's All Beware the Latin Square. *Quality Engineering, 1*, 453–465.
- Ihaka, R., & Gentleman, R. (1996). R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics, 5*, 299-314.
- Jarrett, R. G., & Hall, W. B. (2012). Generalized cyclic incomplete block designs. *Biometrika, 65*, 397–401.
- Johnson, R. T., Hutto, G. T., Simpson, J. R., & Montgomery, W. C. (2012). Designed Experiments for the Defense Community. *Quality Engineering,, 24*, 60–79. doi:DOI: 10.1080/08982112.2012.627288
- Johnston, A. E. (1986). Rothamsted Experimental Station Report. *Rothamsted Soils Division, 137-166;186-191.*
- Jones, B. a. (2010). Alternatives to resolution iv screening designs in 16 runs. *Int. J. Experimental Design and Process Optimisation, 1*, 285–295.
- Jones, B., & Nachtsheim, C. J. (2011). A class of three-level designs for definitive screening in the presence of second-order effects. *Journal of Quality Technology, 43*, 1–15.
- Kackar, R. N. (1985). Off-Line Quality Control, Parameter. *ournal of Quality Technology, 17*, 176-188.
- Kamon, A., Samet, B., Bouaziz, J., & Chaabouni, M. (1999). Application of a rotatable orthogonal central composite design to the

- optimization of the formulation and utilization of a useful plasticizer for cement. *EDP Sciences*, 27, 91-96.
- Kempthorne, O. (1952). *The Design and Analysis of Experiments*. New York: John Wiley & Sons.
- Kerry, S. M., & Bland, J. M. (1998). Analysis of a trial randomized in clusters. *British Journal of Medicine*(316), 54-54.
- Keuls, M. (1952). The use of the studentized range in connection with analysis of variance. *Euphytica*, 1, 112-122.
- Kiefer, J. (1958). On the nonrandomized optimality and randomized nonoptimality of symmetrical designs. *Annals of Mathematical Statistics*, 29, 675-699.
- Kiefer, J. (1961). Optimum Designs in Regression Problems II. *Annals of Mathematical Statistics*, 32, 298–325.
- Kiefer, J., & Wolfowitz, J. (1959). Optimum Designs in Regression Problems. *Annals of Mathematical Statistics*, 30, 271-294.
- Koselka, R. (1996). The New Mantra: MVT. *Forbes*.
- Kshirsager, A. M. (1958). A note on incomplete block designs. *Annals of Mathematical Statistics*, 29, 907-910.
- Kuhfeld, W. F. (2009). *Orthogonal arrays*. website courtesy of SAS institute. Retrieved from <http://support.sas.com//techsup/technote/ts723.html>
- Kutner, M. H., Nachtsheim, C. J., Neter, J., & Li, W. (2004). *Applied Linear Statistical Models*. New York: McGraw Hill.
- Lawson, J. (1982). Applications of robust regression in designed industrial experiments. *Journal of Quality Technology*, 14, 19-33.
- Lawson, J. (2010). *Design and Analysis of Experiments with SAS*. CRC Press: Boca Raton.

- Lawson, J. (2013a). Vdgraph: A package for creating variance dispersion graphs. *The R Journal*, 4(1), 41-44.
- Lawson, J. (2013b). Vdgraph: Variance Dispersion Graphs for Response Surface designs. *R package version 2.2.3*.
- Lawson, J. S. (2003). One step screening and process optimization experiments. *The American Statistician*, 15-20.
- Lawson, J. S. (2008). Bayesian interval estimates of variance components used in quality improvement studies. *Quality Engineering*, 20, 334-345.
- Lawson, J. S., & Gatlin, J. (2006). Finding bad values in factorials-revisited. *Quality Engineering*, 18, 491-501.
- Lawson, J. S., Schaalje, G. B., & Collings, B. J. (2009). Blocking mixed level factorials with SAS. *Journal of Statistical Software*, 32, 1-19.
- Lawson, J., & Erjavec, J. (2001). *Modern Statistics for Engineering and Quality Improvement*. Pacific Grove, CA: Duxbury.
- Lawson, J., Grimshaw, S., & Burt, J. (1998). A quantitative method for identifying active contrasts in unreplicated factorial experiments based on the half normal plot. *Computational Statistics and Data Analysis*, 26, 425-436.
- Leitnaker, M. G., & Cooper, A. (2005). Using statistical thinking and designed experiments to understand process operation. *Quality Engineering*, 17, 279-289.
- Lenth, R. V. (1989). Quick and easy analysis of un-replicated factorials. *Technometrics*, 31, 469-473.
- Lenth, R. V. (2009). Response-surface methods in r, using rsm. *Journal of Statistical Software*, 32(7), 1-17.
- Lephart, S. M., Sell, T. C., Abt, J. P., Nagai, T., Beals, K., Lovalekar, M. T., ... Wirt, M. D. (2013). *Injury Prevention and Performance Enhancement in 101st Airborne Soldiers*. Pittsburgh, PA: U.S. Army Medical Research and Materiel Command.

- Li, W., & Nachtsheim, C. (2000). Model robust factorial designs. *Technometrics*, 42, 345–352.
- Li, X., Sudarsanam, N., & Frey, D. (2006). Regularities in data from factorial experiments. *Complexity*, 11, 32–45.
- Light, R. J., Singer, J. D., & Willett, J. B. (1990). *By Design: Planning Research on Higher Education*. Harvard University Press.
- Lim, D. H., & Wolfe, D. A. (1997). Nonparametric tests for comparing umbrella pattern treatment effects with a control in a randomized block design. *Biometrics*, 53, 410-418.
- Lin, D. K. (1999). Spotlight interaction effects in main effect plans: superstaturated design approach. *Quality Engineering*, 11, 133–139.
- Lin, D. K., & Draper, N. R. (1992). Projection properties of Plackett-Burman designs. *Technometrics*, 34, 423–428.
- Lin, L. I., & Stephenson, W. R. (1998). Validating an assay of viral contamination. In R. Peck, L. D. Haugh, & e. A. Goodman, *Statistical Case Studies: A Collaboration Between Academe and Industry*. Philadelphia.: SIAM.
- Loepky, J. L., Sitter, R. R., & Tang, B. (2007). Nonregular designs with desireable projection properties. *Technometrics*, 49, 454–467.
- Lumley, T. (2009). leaps: Regression subset selection. . *package version 2.9*.
- Ma, H. S., Roberts, A. P., Prévost, J. H., Jullien, R., & Scherer, G. W. (2000). Mechanical structure-property relationship of aerogels. *Journal of Non-Crystalline Solids*, 277(2-3), 127-141.  
doi:[https://doi.org/10.1016/S0022-3093\(00\)00288-X](https://doi.org/10.1016/S0022-3093(00)00288-X)
- Mason, R. L., Gunst, R. F., & Hess, J. L. (1989). *Statistical Design and Analysis of Experiments: with Applications to Engineering and Science*. New York: John Wiley & Sons.
- McAllister, B. (2003). Measures of effectiveness for testing and training. *71st MORS Symposium*. Quantico, VA.

- McCullagh, P., & Nelder, J. A. (1989). *Generalized Linear Models*. (Second ed.). London: Chapman & Hall.
- Meador, M. A., Fabrizio, E. F., Ilhan, F., Dass, A., Zhang, G., Vassilaras, P., . . . Leven, N. (2005). *Crosslinking Amine-Modified Silica Aerogels with Epoxies: Mechanically Strong Lightweight Porous Materials*. Cleveland OH: NASA Glenn Research Center.
- Meier, P. (1972). The biggest health experiment ever: The 1954 the Salk polio vaccine. In J. M. Tanur, F. Mosteller, W. H. Kruskal, R. F. Link, & R. Pieters, *Statistics: A Guide to the Unknown* (pp. 2-13). San Francisco: Holden Day.
- Montgomery, D. C. (1997). *Design and Analysis of Experiments*. Hoboken, NJ: John Wiley and Sons, Inc.,.
- Montgomery, D. C., & Runger, G. C. (1996). Foldovers of 2<sup>k-p</sup> resolution IV experimental designs. *Journal of Quality Technology*, 28, 46–450.
- Moskowitz, H. (1988). *Applied Sensory Analysis of Foods*. Boca Raton, FL: CRC Press.
- Myers, R. H., & Montgomery, D. C. (2002). *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*. New York: John Wiley & Sons.
- Myers, R. H., Montgomery, D. C., Vining, C., Borror, C. M., & Kowalski, S. (2004). Response Surface Methodology: A Retrospective and Literature Survey. *Journal of Quality Technology*, 53–77.
- Nair, V. N. (1992). Taguchi's Parameter Design: A Panel Discussion. *Technometrics*, 34, 127–161.
- Nelson, L. S. (1983). Variance estimation using staggered nested designs. *Journal of Quality Technology*, 15, 232-242.
- Newman, D. (1939). The distribution of the range in samples from a normal population. *Biometrika*, 31, 20-20.

- Nyberg, K. (1999). Using factorial design and response surface methodology to optimize growth parameters of PECVD silicon nitride. *Physica Scripta*, *T79*, 266-271.
- Ott, E. (1967). Analysis of means- a graphical procedure. *Industrial Quality Control*, *24*, 101-109.
- Pignatiello, J. J., & Ramberg, J. S. (1992). Top Ten Triumphs and Tragedies of Genichi Taguchi. *Quality Engineering*, *4*, 211–225.
- Plackett, R. L., & Burman, J. P. (1946). The design of optimum multifactorial experiments. *Biometrika*, *33*, 305–325.
- Porter, W. P., & Busch, R. L. (1978). Fractional factorial analysis of growth and weaning success in *Peromyscus maniculatus*. *Science*, *202*, 907–910.
- Prairie, R. R., & Anderson, R. L. (1962). Optimal designs to estimate variance components and to reduce product variability for nested classifications. *North Carolina State College, Raleigh, Inst. of Stat Mimeo Series*(313), 80-81.
- R Development Core Team. (2003). R: A language and environment for statistical computing. *R Foundation for statistical computing*.
- Ramakrishna, D. M., Viraraghavan, T., & Jin, Y. (2006). Iron oxide coated sand for arsenic removal: Investigation of coating parameters using a factorial design approach. *Practice Periodical of Hazardous, Toxic, and Radioactive Waste Management*, *10*, 198–206.
- Rao, P. S., & Rao, R. S. (1997). *Variance Component Estimation*. London: Chapman & Hall.
- Roquemore, K. G. (1976). Hybrid designs for quadratic response surfaces. *Technometrics*, *18*, 419-423.
- SAS Institute. (2004a). *Gantt Procedure*. Cary, NC: SAS/QC.
- SAS Institute. (2004b). *Ishikawa Procedure* . Cary, NC: SAS/QC.

- Schoen, E. D. (2010). Optimum designs versus orthogonal arrays for main effects and two-factor interactions. *Journal of Quality Technology*, 42, 197-208.
- Searle, S. R., Casella, G., & McCulloch, C. E. (1992). *Variance Components*. New York: John Wiley & Sons.
- Selwyn, M. R., & Hall, N. R. (1984). On Bayesian methods for bioequivalence. *Biometrics*, 40, 1103-1108.
- Snee, R. D. (1983). Graphical analysis of process variation studies. *Journal of Quality Technology*, 15, 76-88.
- Sower, V. E., Savoie, M. J., & Renick, S. (1999). *An Introduction to Quality Management and Engineering*. Upper Saddle River, NJ: Prentice Hall.
- Strickland, J. (2014). *Predictive Modeling and Analytics*. Lulu, Inc.  
Retrieved from  
[https://www.lulu.com/spotlight/strickland\\_jeffrey](https://www.lulu.com/spotlight/strickland_jeffrey)
- Strickland, J. (2014). *Verification and Validation for Modeling and Simulation*. Colorado Springs: Lulu, Inc. Retrieved from  
<https://www.lulu.com/shop/jeffrey-strickland/verification-and-validation-for-modeling-and-simulation/paperback/product-1mwp8g4g.html>
- Strickland, J. (2016). *Data Analytics using Open-Source Tools*. Lulu, Inc.  
Retrieved from  
[https://www.lulu.com/spotlight/strickland\\_jeffrey](https://www.lulu.com/spotlight/strickland_jeffrey)
- Strickland, J. (2017). *Logistic Regression Inside-Out*. Lulu, Inc. Retrieved from [https://www.lulu.com/spotlight/strickland\\_jeffrey](https://www.lulu.com/spotlight/strickland_jeffrey)
- Strickland, J. (2019). *Data Science Applications using R*. Colorado Springs: Lulu, Inc. Retrieved from <https://www.lulu.com/shop/jeffrey-strickland/data-science-applications-using-r/paperback/product-166qj8wg.html>
- Strickland, J. (2020). *Data Science Applications using Python and R*. Colorado Springs: Lulu, Inc. Retrieved from  
<https://www.lulu.com/shop/jeffrey-strickland/data-science-applications-using-python-and-r/paperback/product-166qj8wg.html>

<applications-using-python-and-r/hardcover/product-vmew6v.html>

Strickland, J. (2020). *Time Series Analysis and Forecasting using Python & R*. Colorado Springs: Lulu, Inc. Retrieved from <https://www.lulu.com/shop/jeffrey-strickland/time-series-analysis-and-forecasting-using-python-r/hardcover/product-4j22z6.html>

Strickland, J. (2022). *The Python Guide for New Data Scientists*. Colorado Springs: Lulu, Inc. Retrieved from <https://www.lulu.com/shop/jeffrey-strickland/the-python-guide-for-new-data-scientists/hardcover/product-wn25p6.html>

Strickland, J. (2023). *Discrete Event Simulation using ExtendSim 10*. Colorado Springs: Lulu Inc. Retrieved from <https://www.amazon.com/Discrete-Event-Simulation-Using-ExtendSim/dp/1716202825>

Strickland, J. (2023). *Regression totum modum*. Colorado Springs: Lulu, Inc. Retrieved from <https://www.lulu.com/shop/jeffrey-strickland/regression-totum-modum/hardcover/product-4kzged.html>

Strickland, J. S. (2023). *Regression totum modum*. Colorado Springs: Lulu, Inc. Retrieved from <https://www.lulu.com/shop/jeffrey-strickland/regression-totum-modum/hardcover/product-4kzged.html>

Sun, D. X., Wu, C. F., & Chen, Y. Y. (1997). Optimal blocking schemes or  $2^p$  and  $2(n - p)$  designs. *Technometrics*, 39, 298–307.

Taguchi, G. (1987). *System of Experimental Design: Engineering Methods to Optimize Quality and Minimize Cost*. White Plains, NY: UNIPUB.

Taguchi, G. (1991). *Introduction to Quality Engineering*. White Plains, NY: UNIPUB.

Taguchi, G., & Wu, Y. (1980). *Introduction to Off-Line Quality Control*. Nagoya, Japan: Central Japan Quality Control Association.

- Therneau, T. (2012). *survival: Survival Analysis. R package version 2.37-2.*
- Tukey, J. W. (1949a). Comparing individual means in the analysis of variance. *Biometrics*, 99-114.
- Tukey, J. W. (1949b). One degree of freedom for non-additivity. *Biometrics*, 5, 232-242.
- Venables, W. N., & Ripley, B. D. (2002). *Modern Applied Statistics with S*. New York: Springer.
- Wagner, J. G. (1967). Use of computers in pharmacokinetics. *Journal of Clinical Pharmacology and Therapeutics*, 8, 201-220.
- Wang, J. C., & Wu, C. F. (1995). A hidden projection property of Plackett-Burman and related designs. *Statistica Sinica*, 5, 235–250.
- Wang, Q., Shi, G., & Chan-Halbrendt, C. (2004). Market potential for fine furniture manufactured from low-grade hardwood: Evidence from a conjoint Analysis. *Forest Products Journal*, 54, 19-25.
- Warnes, G. R. (2012). *gmodels: Various R programming tools for model*. CRAN.
- Westlake, W. J. (1965). Composite designs based on irregular fractions of factorials. *Biometrics*, 21, 324-336.
- Wheeler, B. (2012). *AlgDesign: Algorithmic Experimental Design. R package version 1.1-7.*
- Wheeler, R. E. (1974). Portable power. *Technometrics*, 16, 193-201.
- Wilburn, N. T. (1963). Application of 2^(8-4) Fractional Factorials in Screening of Variables Affecting the Performance of Dry Process Zinc Battery Electrodes. *Proceedings of the Eighth Conference on the Design of Experiments in Army Research Development and Testing*, (pp. 17-49).

- Wilk, M. B., Gnanadesikan, R., & Huyett, M. J. (1962). Probability plots for the gamma distribution. *Technometrics*, 4, 193-201.
- Woignier, T., & Phalippou, J. (1987). Skeletal density of silica aerogels. *Journal of Non-Crystalline Solids*, 93(1), 17-21.  
doi:[https://doi.org/10.1016/S0022-3093\(87\)80024-8](https://doi.org/10.1016/S0022-3093(87)80024-8)
- Woignier, T., & Phalippou, J. (1988). Mechanical strength of silica aerogels. *Journal of Non-Crystalline Solids*, 100(1-3), 404-408.  
doi:[https://doi.org/10.1016/0022-3093\(88\)90054-3](https://doi.org/10.1016/0022-3093(88)90054-3)
- Wu, C. F., & Hamada, M. (2000). *Experiments-Planning, Analysis, and Parameter Design Optimization*. New York: John Wiley & Sons.
- Zahran, A., Anderson-Cook, C. M., & Myers, R. H. (2003). Fraction of design space to assess prediction capability of response surface designs. *Journal of Quality Technology*, 35, 377-386.



# Index

---

## 2

$2^k$  factorial..... 111, 259, 260, 381

## A

aberration criteria..... 277, 278

adjusted R-squared..... 33, 56, 120, 126, 134, 147, 172, 269, 283, 319, 320, 337, 339, 340, 360, 362, 453, 456

Advanced Combat Helmet (ACH)..... 168

aerogel density..... 32, 34, 38, 44, 46, 47, 52, 55, 56, 199

`agricolae` package..... 59, 166, 189

Air Force Datalab VAULT..... 354

`AlgDesign` package ..... 99, 302, 370, 407

alias structure ..... 262

`Altscreen` function..... 315

analysis of variance ..... 19, 28, 36, 38, 101, 204, 209, 210, 223, 256, 475, 482

`Anova` function..... 162

ANOVA table .... 36, 45, 47, 79, 81, 90, 91, 95, 96, 162, 167, 172, 175, 182, 191, 204, 205, 206, 207, 212, 219, 221, 226, 228, 234, 235, 241, 242, 451, 454

antireflection coating..... 143

A-optimal ..... 355

`aov` function ..... 37, 41, 82, 90, 162, 185, 205, 207, 226, 242, 243, 246, 376

arcsin square root transformation..... 104

Arsenic Removal Experiment ..... 297, 300

`as.factor` function..... 145

`as.numeric` ..... 271, 273, 286, 287, 288, 290, 303, 402, 404, 405

atmospheric drag..... 2

`audit` function..... 156

`auditor` package ..... 149, 156

autocorrelation plot..... 158

## B

Balanced Incomplete Block Design ..... 368, 370, 371

bar chart..... 49

Bayes posterior probabilities ..... 138

bias error ..... 7

biased factor ..... 7

BIB .....	<i>See</i> Balanced Incomplete Block
block.....	177, 178, 179, 180, 185, 187
blocking ...	9, 16, 19, 163, 168, 175, 177, 179, 180, 188, 189, 190, 191, 192, 193, 195, 294, 299, 335, 358, 374, 380, 394, 401, 403, 417, 418, 426, 469, 470, 481
<b>boxcox</b> function.....	44
Box-Cox plot .....	44
Box-Cox transformation .....	43, 44, 45, 48
<b>BsMD</b> package.....	130, 308
<b>BsProb</b> function.....	138

## C

<b>car</b> package .....	90, 162, 373
cause and effect .....	3, 4, 5, 12, 22, 65, 85
cell means ...	30, 31, 34, 35, 40, 41, 43, 47, 51, 52, 53, 57, 58, 59, 66, 67, 70, 76, 77, 81, 83, 84, 85, 86, 87, 88, 91, 148, 159, 162, 225, 226
Central Limit Theorem .....	46, 134
chi-square distribution.....	34, 35, 204, 207
chi-square statistic .....	49
clear effects criteria.....	277
Clopper-Pearson method .....	385
Cochran and Cox .....	18
coefficient matrix .....	35, 36
complete block.....	180
completely randomized design ....x, 23, 25, 29, 31, 32, 35, 36, 38, 45, 160, 167, 168, 179, 180, 374, 396	
conceptual battlespace .....	1
confidence interval.....	58
<b>confint</b> function.....	213
confounding factor .....	7, 9, 262
confounding pattern .....	262
constancy .....	39
<b>constrOptim</b> function.....	466
contour plot .....	71, 458, 464
<b>contr.poly</b> function.....	55, 56, 95, 120, 172, 176
contrast matrix.....	54, 56, 82
corrected sum of squares.....	52
covariance matrix.....	34, 254, 302, 406, 424
CRD..... <i>See</i> completely randomized design, <i>See</i> completely randomized design	
CRFD .....	<i>See</i> completely randomized factorial design

## D

<b>daewr</b> function.....	52
<b>daewr</b> package.....	87, 130, 145
<b>DanielPlot</b> function .....	130, 134
<b>DanielPlots</b> .....	151
<b>data.frame</b> function .....	27
Defense Acquisition Framework.....	1
defining relation.....	263
degrees of freedom .....	33, 34, 35, 36, 37, 52, 56, 79, 87, 93, 94, 120, 126, 130, 133, 134, 146, 153, 154, 166, 167, 168, 172, 178, 181, 186, 187, 192, 207, 209, 213, 221, 222, 223, 228, 229, 231, 237, 249, 257, 268, 269, 282, 283, 319, 320, 324, 337, 339, 340, 360, 361, 382, 389, 390, 393, 394, 397, 398, 399, 402, 404, 405, 410, 436, 452, 458
dependent variable.....	See Response
design objectives.....	3
design of experiments .....	1, 3, 20, 180, 353
<b>design.info</b> function.....	265
<b>design.lsd</b> function .....	189
<b>design.rcbd</b> function .....	166
detect.....	4, 5, 13, 14, 86, 87, 88, 99, 109, 129, 178, 321, 363, 364, 368
discrete variable.....	116
<b>dispersion</b> parameter.....	153, 154
<b>DoE.base</b> package .....	402
doer Shiny App.....	354
D-optimal .....	302, 304, 355, 370
Ds criteria.....	406
Dunnett contrasts .....	63, 64
Dunnett's method .....	62, 63, 67
Dunnett's test .....	63
duplicates.....	7

## E

effects model .....	30, 31, 45, 76, 77
error control.....	xi, 9, 26, 51, 163
Error Sums of Squares.....	31, 44, 78, 97
error variance .....	37
esistance to penetration.....	169
estimability order.....	395
<b>estimable</b> function .....	34, 54, 77, 82, 91, 243, 244, 257

<b>expand.grid</b> function .....	75
experimental design....	ix, x, xii, 1, 3, 5, 6, 7, 8, 10, 11, 12, 13, 14, 18, 19, 22, 28, 65, 115, 131, 132, 144, 237, 333, 334, 335, 348, 349, 352, 444, 471
experimental error ..	7, 9, 12, 13, 14, 26, 34, 37, 39, 40, 47, 51, 65, 87, 129, 130, 134, 163, 164, 166, 170, 175, 180, 187, 195, 254, 264, 380, 422, 426, 436, 449
experimental error variance .....	26, 37, 380
experimental unit .x, xi, .	6, 7, 9, 10, 13, 14, 15, 16, 22, 25, 26, 27, 28, 30, 34, 38, 39, 40, 41, 48, 65, 66, 71, 80, 97, 99, 101, 124, 132, 139, 160, 163, 164, 165, 166, 167, 168, 171, 174, 175, 177, 179, 180, 181, 183, 184, 188, 193, 194, 199, 201, 202, 227, 238, 242, 257, 265, 268, 328, 367, 368, 369, 370, 371, 374, 375, 378, 379, 380, 381, 383, 384, 387, 396, 408, 409, 411, 413, 414, 416, 417, 419
experimentation....	2, 4, 5, 6, 10, 11, 15, 18, 28, 69, 72, 129, 132, 162, 197, 199, 216, 259, 321, 324, 328, 368, 375, 421, 426, 430, 432, 441, 463
<b>F</b>	
<b>fac.design</b> function .....	402
factor effects .....	9, 12, 13, 69, 71, 81, 85, 104, 111, 132, 162
<b>factor</b> function.....	27, 32
factor levels.... xi, .	9, 11, 13, 14, 26, 27, 29, 31, 32, 38, 39, 40, 41, 42, 54, 55, 58, 66, 69, 80, 89, 95, 97, 124, 125, 126, 129, 132, 133, 134, 140, 163, 164, 165, 172, 174, 175, 176, 188, 189, 199, 227, 260, 261, 262, 263, 267, 274, 291, 292, 293, 305, 322, 327, 335, 363, 368, 370, 375, 378, 384, 403, 405, 409, 427, 428, 433, 438, 440, 441, 465
factorial design.....	69, 75
factors ix, 2, 10, 20, 64, 70, 80, 90, 101, 111, 120, 131, 150, 153, 160, 171, 181 , 190, 201, 223, 231, 244, 260, 272, 281, 291, 301, 310, 316, 321, 326, 327, 330, 331, 332, 333, 347, 349, 362, 364, 380, 383, 384, 397, 399, 400, 401, 424, 426, 428, 430, 431, 440, 450, 454, 458, 460, 463, 472	
<b>F-distribution</b> .....	36, 50, 52, 79, 207, 215
First Article Test protocol .....	169
first order effects .....	115, 348
Fisher .....	131
<b>Fisher Scoring</b> .....	153, 154
Fisher, Ronald A. ix, 5, 9, 18, 19, 25, 28, 51, 65, 69, 70, 153, 154, 164, 204, 370, 469, 472	
<b>fit.contrast</b> function .....	35, 54
flow diagram .....	13
<b>fold.design</b> function .....	294

<b>Fpower</b> function.....	52
<b>Fpower2</b> function.....	87
fractional factorial design .....	3, 23, 101, 102, 262, 263, 264, 265, 268, 274, 278, 280, 281, 291, 292, 297, 303, 304, 305, 309, 310, 315, 321, 322, 323, 324, 326, 330, 333, 334, 335, 363, 364, 381, 388, 389, 392, 394, 395, 396, 419, 426, 434, 469, 470, 471
fractional factorial experiment .....	101
fractional replicate design .....	102
<b>FrF2</b> function.....	265, 279, 397
<b>FrF2</b> package .....	127, 275, 294, 333
<i>F</i> -statistic.....	33, 55, 56, 120, 126, 134, 147, 172, 247, 269, 283, 319, 320, 337, 339, 340, 360, 362, 453, 456
<i>F</i> -test ...	6, 48, 57, 67, 78, 90, 93, 94, 95, 130, 162, 172, 175, 178, 184, 186, 195, 241, 245, 406
<b>fullnormal</b> function.....	134
<i>F</i> -value.....	37
<b>G</b>	
Gage R&R sampling.....	217
Gage repeatability.....	217, 218
Gage reproducibility .....	217
Gantt Chart .....	15
gap statistic .....	140
<b>Gaptest</b> function .....	140, 141
GCB .....	<i>See</i> generalized complete block design
<b>gen.factorial</b> function.....	99, 303, 407
generalized complete block design.....	183
generalized cyclic designs .....	417
generalized likelihood ratio test .....	37
Generalized Linear Model.....	153, 356
<b>glht</b> function.....	62, 63
<b>GLM</b> .....	<i>See</i> Generalized Linear Model
<b>glm</b> function .....	104
<b>gmodels</b> function .....	54
<b>gmodels</b> package .....	35, 82, 244
grand mean.....	30, 36, 52, 260
graphical user interface .....	1, 472
greenhouse warning .....	2
GUI .....	<i>See</i> graphical user interface

## H

half-fraction.....	260
<b>halfnorm</b> function .....	308
<b>half-normal</b> plot .....	135, 136, 139, 140, 142, 149, 173, 251, 255, 269, 284, 285, 298, 299, 307, 308, 363, 385, 386, 393
half-normal plots.....	130, 162, 247, 249, 257, 470
Hartley's method.....	223, 232
heterogeneity.....	42, 45
homogeneous .....	25
homogeneous groupings.....	163

## I

<b>IAPlot</b> function .....	127, 271
<b>ihstep</b> fucntion .....	320
immediacy .....	19
incomplete block designs .....	368
independence.....	40, 66, 204, 242
independent factors.....	8
independent variables.6, 7, 8, 10, 13, 25, 29, 104, 121, 125, 351, 421, 422, 423, 424, 444	
interaction plot .....	74, 83, 84, 95, 120, 126, 127, 128, 136, 150, 151, 247, 257, 271, 272, 286, 287, 288, 289, 342, 386
<b>interaction.plot</b> function .....	83, 95
Ishikawa Cause-and-Effect Diagrams .....	13

## J

jammers .....	329
---------------	-----

## L

Latin-square design .....	iv, 24, 187, 188, 189, 192, 193, 195
Lawson, Grimshaw & Burt Rn Statistic .....	141, 142, 271
least squares .18, 31, 47, 48, 66, 67, 91, 162, 240, 246, 264, 301, 351, 368, 373, 380, 381, 406, 409, 411, 415, 454, 455, 457, 458, 461, 462	
least-squares means.....	91
Lenth's plot of effects .....	138
<b>LenthPlot</b> function.....	138
level.... 4, 6, 20, 25, 26, 29, 30, 32, 34, 38, 39, 43, 47, 56, 57, 58, 62, 63, 65, 67, 69, 70, 71, 74, 76, 78, 82, 83, 84, 85, 89, 91, 92, 93, 95, 96, 103, 106, 108, 109, 110, 111, 113, 129, 130, 134, 137, 139, 148, 150, 154, 157, 160, 161, 162, 163, 166, 167, 170, 173, 181, 183, 185, 186, 187, 189, 192, 193, 195,	

197, 199, 204, 205, 207, 210, 216, 227, 228, 230, 239, 245, 248, 256, 257, 271, 286, 287, 298, 300, 305, 313, 321, 323, 324, 325, 326, 331, 356, 364, 368, 369, 371, 373, 374, 375, 376, 377, 378, 379, 381, 398, 400, 401, 403, 404, 409, 412, 413, 416, 417, 429, 447, 452, 468, 469, 470, 472, 474, 476	
<b>LGB</b> function.....	130, 135, 136, 269
Light Gradient Boosting .....	135
likelihood ratio test.....	48, 49
linear combination.....	34, 462
<b>lm</b> function.....	35, 37, 47, 77, 99, 111, 125, 134, 136, 385
<b>lme4</b> package.....	213, 236, 257
<b>lmer</b> function .....	211, 224, 236, 243, 247, 254
log likelihood.....	44
logistic regression .....	xv
Lot Acceptance Testing protocol .....	169
LSD .....	<i>See Latin-square design</i>
<b>lsmeans</b> function.....	19, 91, 92, 93, 144, 162, 244, 245, 373, 376, 416
<b>lsmeans</b> package.....	162
lurking variable .....	6, 7, 8, 9, 13, 16, 26, 29, 65, 76, 100, 164, 189, 455
<b>M</b>	
margin of error.....	138
marginal means.....	77, 81, 85, 86, 87, 88, 89, 91, 92, 162, 242, 373
<b>MASS</b> package.....	44, 48
mathematical model.....	4, 11, 29, 76
maximum likelihood .....	30, 48, 66, 67, 104, 105, 209, 210, 223, 256
ME.....	<i>See margin of error</i>
mean squares.....	36
measures.....	3
model residuals.....	39, 40
<b>model.tables</b> function.....	77
<b>ModelRobust</b> .....	315
<b>ModelRobust</b> function .....	316
msT.....	<i>See Mean Squares Treatment</i>
multi-factor factorial design .....	98, 101
multiple R-squared.....	33, 56, 120, 126, 134, 147, 172, 269, 283, 319, 320, 337, 339, 340, 353, 360, 362, 453, 456
multistage sampling.....	212
multivariable testing .....	22
multivariate normal distribution.....	34, 204

## **N**

nested sampling experiment.....	23, 227
Newman-Keuls method.....	59, 60, 67
noncentral <i>F</i> -distribution .....	50, 52, 79
noncentrality factor .....	90
noncentrality parameter .....	35, 36, 50, 52, 79, 86, 90, 178
non-replicated $2^k$ design .....	140
normal plot.....	140
normal probability plot .....	40, 41, 42, 67, 134, 247, 254, 269
normality.....	39
null deviance .....	153, 154
null hypothesis .....	36, 37, 50, 51, 53, 58, 90, 104, 204
number of runs .....	129, 132, 259, 262, 277, 278, 301, 302, 305, 310, 311, 317, 322, 324, 335, 363, 413, 445

## **O**

<a href="#">oa.design</a> function.....	323
objectives .....	12, 14, 15, 28
observation .....	1, 2, 39, 65, 89, 91, 93, 94, 104, 140, 180
observational studies .....	1, 4, 8, 22
one-factor sampling .....	201
operational acceptance.....	1
operational experiments.....	1
<a href="#">optBlock</a> function .....	370
<a href="#">optFederov</a> function .....	302, 321, 324
<a href="#">optim</a> function.....	324
orthogonal array designs .....	409
orthogonal comparisons .....	67
orthogonal contrasts.....	57
orthogonal matrix .....	56
orthogonal polynomial.....	55, 93, 94, 97, 98, 172, 176
orthogonal polynomial contrasts .....	57, 94, 97, 98, 172, 176
orthogonality.....	111, 113, 128, 260, 291, 302, 305, 363, 364, 409, 410
outlier.....	141

## **P**

partial confounding .....	411
partially balanced incomplete block .....	377, 418
PECVD.....	<i>See</i> Plasma Enhanced Chemical Vapor Deposition
<a href="#">persp</a> function .....	459

pilot tests .....	13, 87
Plackett-Burman design matrix.....	307
Plasma Enhanced Chemical Vapor Deposition .....	143
<b>polr</b> function .....	49
<i>porosimetry</i> .....	117, 350
power .. 43, 45, 50, 51, 52, 53, 69, 71, 85, 86, 87, 88, 93, 98, 124, 129, 130, 132, 160, 162, 163, 167, 178, 179, 182, 187, 192, 193, 215, 216, 305, 354, 358, 359, 367, 368, 369, 372, 374, 375, 381, 388, 392, 396, 482	
practical difference .....	14, 52, 86
practical effect .....	7, 130
practical significance.....	51, 85
preplanned comparisons .....	54, 55, 57, 81
projective property .....	278, 364
PSE .....	<i>See</i> pseudo standard error
pseudo standard error .....	140
p-value 33, 37, 45, 49, 56, 58, 120, 126, 134, 147, 172, 269, 283, 319, 320, 337, 339, 340, 360, 362, 453, 456	
<i>p</i> -values.....	55
Python programming .....	xiv

## **Q**

<b>qchisq</b> function .....	207, 214
<b>qf</b> function.....	207
quadratic form .....	90, 204

## **R**

rad-hard .....	<i>See</i> Radiation Space Component Protection
Radiation Space Component Protection.....	54
raicloud plot.....	122
random error .....	7
random sampling experiments .....	199
randomization.. ix, 9, 10, 14, 16, 19, 25, 26, 30, 40, 76, 164, 166, 171, 175, 189, 195, 268, 335	
randomization test.....	26
randomized block design .....	163, 164, 178, 181, 187, 193, 367, 477
randomized complete block design .....	164, 166, 178, 179, 180, 193, 371, 372
randomizing .....	14, 65, 189, 195
RBD .....	<i>See</i> randomized block design
<b>rbind</b> function .....	75
RCBD .....	<i>See</i> randomized complete block design

<b>RE</b> <i>See</i> relative efficiency	
<b>regsubsets</b> function .....	311
relative efficiency .....	193
replicate runs.....	7, 8, 9, 14, 70, 80
replicates ix, 25, 26, 29, 30, 31, 34, 39, 40, 41, 51, 52, 53, 66, 69, 71, 77, 78, 79, 85, 86, 87, 88, 90, 91, 94, 98, 113, 124, 129, 130, 143, 160, 162, 167, 173, 179, 183, 192, 194, 202, 204, 207, 214, 218, 221, 223, 225, 228, 238, 264, 265, 269, 363, 367, 369, 373, 374, 379, 380, 385, 411, 413, 436, 470	
reproducibility.....	220
<b>residual deviance</b> .....	153, 154
residual standard error ..... 33, 56, 120, 126, 134, 147, 172, 269, 283, 319, 320, 337, 339, 340, 360, 362, 458	
<b>Residuals</b> function .....	37
resolution 277, 278, 281, 284, 285, 292, 293, 294, 295, 296, 307, 309, 310, 314, 322, 323, 327, 333, 335, 363, 364, 390, 396, 397, 426, 434, 474, 478	
resolution criteria.....	277
resolution R design.....	278
response score .....	49
response variable .xii, 2, 3, 4, 6, 7, 8, 9, 11, 13, 14, 16, 19, 24, 25, 29, 30, 40, 42, 43, 46, 47, 48, 55, 58, 61, 64, 65, 71, 74, 76, 80, 85, 89, 98, 99, 100, 101, 104, 105, 109, 110, 111, 112, 117, 118, 124, 125, 127, 129, 132, 134, 140, 141, 142, 145, 146, 162, 165, 169, 174, 178, 183, 186, 190, 197, 198, 199, 201, 205, 211, 216, 218, 226, 238, 247, 257, 258, 263, 266, 267, 268, 269, 271, 281, 285, 294, 297, 307, 308, 311, 317, 319, 320, 321, 330, 342, 353, 354, 359, 375, 384, 385, 392, 414, 421, 422, 423, 424, 425, 426, 431, 433, 434, 437, 438, 441, 444, 446, 447, 450, 453, 454, 455, 456, 458, 460, 461, 463, 464, 466, 467, 469, 470, 471, 472, 473, 479, 483	
RIBD.....	<i>See</i> Randomized Incomplete Block Design
ROE.....	<i>See</i> rules of engagement
Rothamsted.....	5, 19, 28, 29, 474
RSE.....	<i>See</i> random sampling experiments
RTP .....	<i>See</i> resistance to penetration
rules of engagement .....	329, 331, 332, 339, 341, 342, 343

## S

sampling surveys.....	1, 22
saturated model.....	269, 282, 393
saturated set of estimable contrasts.....	35, 36
scientific method.....	8, 22
<b>score_mse</b> function .....	156

sensitivity .....	45, 51, 163, 182, 187, 188, 193
sequentiality .....	19
Shiny.....	354
significance level .....	37, 50, 51, 57
silica aerogels.....	25, 114, 115, 347, 351, 352, 471, 483
simulated data .....	14, 331
simulation .....	2, 328, 331, 354, 480
simultaneous margin of error .....	138
<code>skpr</code> package .....	354
SME .....	<i>See</i> simultaneous margin of error
<code>Snk.test</code> function .....	59, 60
SNSE .....	<i>See</i> Staggered nested sampling experiments
software system.....	1
space debris .....	2
space domain awareness.....	1
<code>ssE</code> .....	<i>See</i> Sum of Squares Error
<code>ssT</code> .....	<i>See</i> Sum of Squares Treatment
<code>ssTotal</code> .....	<i>See</i> Sum of Squares Total
stabilizing transformation .....	42, 43
Staggered nested sampling experiments.....	230
standard deviation .....	45, 47, 81, 87, 129, 130, 132, 252, 268
standard error of an effect .....	135
standardized-gap .....	141
statistical significance .....	51
statistical experimental design .....	21
<code>str</code> function.....	145
studentized range distribution .....	57
studentized range statistic.....	59
Sum of Squares Error .....	36, 37, 38, 39, 104, 133, 209, 210
Sum of Squares Total .....	36
Sum of Squares Treatment .....	36, 37, 209, 210
<code>summary.aov</code> function .....	172
<code>summary.lm</code> function.....	172
sums of squares .....	78
sums of squares about the mean.....	<i>See</i> sum of squares total
system characterization.....	ix, 3
system comparison .....	3
system effectiveness.....	3
system error identification .....	3
system performance .....	3

system reliability .....	3
--------------------------	---

## T

t value	55, 56, 63, 64, 82, 120, 126, 134, 146, 153, 154, 171, 212, 220, 225, 237, 243, 244, 268, 283, 319, 320, 337, 339, 340, 360, 361, 453, 455, 458
Taguchi's methodology .....	20
<b>tapply</b> function .....	92, 108
target .....	6, 149, 156, 202, 329
<i>t</i> -distribution .....	54, 57
test design .....	ix, 3, 354
Test Design Suite .....	354
thermosphere .....	2, 3
thermospheric cooling .....	2
three-factor interaction .....	382, 405
treatment effects .....	26, 37, 45, 51, 71, 192, 193, 477
treatment factor. xi, 15, 25, 26, 29, 30, 34, 35, 38, 39, 40, 42, 43, 47, 48, 49, 53, 54, 62, 63, 66, 71, 89, 124, 160, 163, 164, 166, 172, 174, 175, 176, 181, 183, 184, 185, 188, 189, 192, 193, 195, 199, 201, 204, 205, 245, 367, 368, 369, 370, 371, 372, 375, 377, 378, 380, 381, 384, 417, 418	
treatment sums of squares .....	36, 381
troposphere .....	1, 3
<b>TukeyHSD</b> function .....	57, 58, 59, 148, 157, 173, 192
Tukey's HSD .....	57, 59, 60, 67, 416
Tukey's test of additivity .....	97
two-level factorial .....	109, 110, 113, 129, 130, 134, 139, 150, 468, 469
two-way interaction .....	115, 348
<b>type I error</b> .....	57, 62, 67
type II sums of squares .....	90, 91
type III sums of squares .....	90, 91, 104, 105, 162, 167, 223, 380, 409, 411

## U

UAV .....	<i>See</i> Unmanned Aerial Vehicle
unbiased estimator .....	34
unconfounded factor .....	12
unconstrained optimization .....	11
Unmanned Aerial Vehicle .....	xi, 64, 65, 69, 70, 72, 73, 76, 86, 93, 201, 202
unweighted mean squares .....	225
upper atmosphere .....	2

***V***

variability ..	10, 12, 20, 26, 29, 39, 41, 43, 45, 140, 163, 164, 175, 178, 183, 184, 188, 197, 198, 201, 202, 207, 210, 211, 212, 217, 223, 224, 228, 230, 233, 237, 256, 266, 271, 273, 367, 368, 380, 409, 473, 479	
variable score.....		<b>49</b>
<a href="#">Vdgraph</a> package.....		<b>428</b>

***W***

weighted least squares .....		47
workabilty .....		426
<a href="#">write.csv</a> function .....		27

***Y***

Yates' order.....		110
-------------------	--	-----

