Introduction (1 of 3)

- Goal is to obtain maximum information with minimum number of experiments
- Proper analysis will help separate out the factors
- Statistical techniques will help determine if differences are caused by variations from errors or not

No experiment is ever a complete failure. It can always serve as a negative example. — Arthur Bloch

The fundamental principle of science, the definition almost, is this: the sole test of the validity of any idea is experiment. — Richard P. Feynman

Introduction (2 of 3)

- Key assumption is non-zero cost
  - Takes time and effort to gather data
  - Takes time and effort to analyze and draw conclusions
  - Minimize number of experiments run
- Good experimental design allows you to:
  - Isolate effects of each input variable
  - Determine effects due to interactions of input variables
  - Determine magnitude of experimental error
  - Obtain maximum info with minimum effort

Introduction (3 of 3)

- Consider
  - Vary one input while holding others constant
    - Simple, but ignores possible interaction between two input variables
    - Test all possible combinations of input variables
      - Can determine interaction effects, but can be very large
      - Ex: 5 factors with 4 levels \( \rightarrow 4^5 = 1024 \) experiments. Repeating to get variation in measurement error \( 1024 \times 3 = 3072 \)
  - There are, of course, in-between choices...
    - (Ch 19, but leads to confounding...)

Outline

- Introduction
- Terminology
- General Mistakes
- Simple Designs
- Full Factorial Designs
  - \( 2^k \) Factorial Designs
  - \( 2^{kr} \) Factorial Designs

Terminology (1 of 4)

(Will explain terminology using example)

- Study PC performance
  - CPU choice: 6800, 280, 8086
  - Memory size: 512 KB, 2 MB, 8 MB
  - Disk drives: 1-4
  - Workload: secretarial, managerial, scientific
  - Users: high school, college, graduate
- Response variable - the outcome or the measured performance
  - Ex: throughput in tasks/min or response time for a task in seconds
Terminology (2 of 4)

- **Factors** - each variable that affects response
  - Ex: CPU, memory, disks, workload, user
  - Also called predictor variables or predictors
- **Levels** - the different values factors can take
  - Ex: CPU 3, memory 3, disks 4, workload 3, users 3
  - Also called treatment
- **Primary factors** - those of most important interest
  - Ex: maybe CPU and memory the most

Terminology (3 of 4)

- **Secondary factors** - of less importance
  - Ex: maybe user type not as important
- **Replication** - repetition of all or some experiments
  - Ex: if run three times, then three replications
- **Design** - specification of the replication, factors, levels
  - Ex: Specify all factors, at above levels with 5 replications so $3 \times 3 \times 4 \times 3 \times 3 = 324$ time 5 replications yields 1215 total

Terminology (4 of 4)

- **Interaction** - two factors A and B interact if one shows dependence upon another
  - Ex: non-interacting factor since A always increases by 2
    
    | A1 | A2 |
    |----|----|
    | B1 | 3  | 5  |
    | B2 | 6  | 8  |
  - Ex: interacting factors since A change depends upon B
    
    | A1 | A2 |
    |----|----|
    | B1 | 3  | 5  |
    | B2 | 6  | 9  |

Outline

- Introduction
- Terminology
- **General Mistakes**
  - Simple Designs
  - Full Factorial Designs
    - $2^k$ Factorial Designs
  - $2^{kr}$ Factorial Designs

Common Mistakes in Experiments (1 of 2)

- **Variation due to experimental error is ignored.**
  - Measured values have randomness due to measurement error. Do not assign (or assume) all variation is due to factors.
- **Important parameters not controlled.**
  - All parameters (factors) should be listed and accounted for, even if not all are varied.
- **Effects of different factors not isolated.**
  - May vary several factors simultaneously and then not be able to attribute change to any one.
  - Use of simple designs (next topic) may help but have their own problems.

Common Mistakes in Experiments (2 of 2)

- **Interactions are ignored.**
  - Often effect of one factor depend upon another. Ex: effects of cache may depend upon size of program. Need to move beyond one-factor-at-a-time designs
- **Too many experiments are conducted.**
  - Rather than running all factors, all levels, at all combinations, break into steps
    - First step, few factors and few levels
    - Determine which factors are significant
    - Two levels per factor (details later)
  - More levels added at later design, as appropriate
**Simple Designs**

- Start with typical configuration
- Vary one factor at a time
  - Ex: typical may be PC with 8080, 2 MB RAM, 2 disks, managerial workload by college student
  - Vary CPU, keeping everything else constant, and compare
  - Vary disk drives, keeping everything else constant, and compare
  - Given k factors, with ith having $n_i$ levels
    $$\text{Total} = 1 + \sum (n_i-1)$$ for $i = 1$ to $k$
  - Example: in workstation study
    $$1 + (3-1) + (3-1) + (4-1) + (3-1) + (3-1) + (3-1) = 14$$
  - But may ignore interaction (Example next)

---

**Example of Interaction of Factors**

- Consider response time vs. memory size and degree of multiprogramming

<table>
<thead>
<tr>
<th>Degree</th>
<th>32 MB</th>
<th>64 MB</th>
<th>128 MB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.25</td>
<td>0.21</td>
<td>0.15</td>
</tr>
<tr>
<td>2</td>
<td>0.52</td>
<td>0.45</td>
<td>0.36</td>
</tr>
<tr>
<td>3</td>
<td>0.81</td>
<td>0.66</td>
<td>0.50</td>
</tr>
<tr>
<td>4</td>
<td>1.50</td>
<td>1.45</td>
<td>0.70</td>
</tr>
</tbody>
</table>

- If fixed degree 3, mem 64 and vary one at a time, may miss interaction
  - Example: degree 4, non-linear response time with memory

---

**Full Factorial Designs**

- Every possible combination at all levels of all factors
- Given k factors, with $i$th having $n_i$ levels
  $$\text{Total} = \Pi n_i$$ for $i = 1$ to $k$
- Example: in CPU design study
  (3 CPUs)(3 mem)(4 disks)(3 loads)(3 users) = 324 experiments
- Advantage is can find every interaction component
- Disadvantage is costs (time and money), especially since may need multiple iterations (later)
- Can reduce costs by: reduce levels, reduce factors, run fraction of full factorial
  (Next, reduce levels)

---

**2^k Factorial Designs**

- Very often, many levels at each factor
  - Ex: effect of network latency on user response time
  - There are lots of latency values to test
- Often, performance continuously increases or decreases over levels
  - Ex: response time always gets higher
  - Can determine direction with min and max
- For each factor, choose 2 alternatives at each level
- 2^factorial designs
- Then, can determine which of the factors impacts performance the most and study those further
2\(^2\) Factorial Design (1 of 4)
- Special case with only 2 factors
  - Easily analyzed with regression
- Example: MIPS for Mem (4 or 16 Mbytes) and Cache (1 or 2 Kbytes)

<table>
<thead>
<tr>
<th>Mem 4MB</th>
<th>Mem 16MB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache 1 KB</td>
<td>15</td>
</tr>
<tr>
<td>Cache 2 KB</td>
<td>25</td>
</tr>
</tbody>
</table>
  
Define \(x_a = -1\) if 4 Mbytes mem, +1 if 16 Mbytes
Define \(x_b = -1\) if 1 Kbyte cache, +1 if 2 Kbytes
Performance:
\[y = q_0 + q_ax_a + q_bx_b + q_{ab}x_ax_b\]

2\(^2\) Factorial Design (2 of 4)
- Substituting:
  \[15 = q_0 - q_a - q_b + q_{ab}\]
  \[45 = q_0 + q_a - q_b - q_{ab}\]
  \[25 = q_0 - q_a + q_b - q_{ab}\]
  \[75 = q_0 + q_a + q_b + q_{ab}\]

Can solve to get:
\[y = 40 + 20x_a + 10x_b + 5x_ax_b\]
- Interpret:
  - Mean performance is 40 MIPS, memory effect is 20 MIPS, cache effect is 10 MIPS and interaction effect is 5 MIPS
  (Generalize to easier method next)

2\(^2\) Factorial Design (3 of 4)

<table>
<thead>
<tr>
<th>Exp</th>
<th>a</th>
<th>b</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>(y_1)</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>-1</td>
<td>(y_2)</td>
</tr>
<tr>
<td>3</td>
<td>-1</td>
<td>1</td>
<td>(y_3)</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>(y_4)</td>
</tr>
</tbody>
</table>

- Solving, we get:
  \[q_0 = \frac{1}{4}(y_1 + y_2 + y_3 + y_4)\]
  \[q_a = \frac{1}{4}(-y_1 + y_2 - y_3 + y_4)\]
  \[q_b = \frac{1}{4}(-y_1 - y_2 + y_3 + y_4)\]
  \[q_{ab} = \frac{1}{4}(y_1 - y_2 - y_3 + y_4)\]

Notice for \(q_a\) can obtain by multiplying “a” column by “y” column and adding
- Same is true for \(q_b\) and \(q_{ab}\)

2\(^2\) Factorial Design (4 of 4)

<table>
<thead>
<tr>
<th>i</th>
<th>a</th>
<th>b</th>
<th>ab</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>15</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>45</td>
</tr>
<tr>
<td>3</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>25</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>75</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
</tr>
</tbody>
</table>

- Multiply column entries by \(y\), and sum
- Dived each by 4 to give weight in regression model
- Final:
  \[y = 40 + 20x_a + 10x_b + 5x_ax_b\]

Allocation of Variation (1 of 3)
- Importance of a factor measured by proportion of total variation in response explained by the factor
  - Thus, if two factors explain 90% and 5% of the response, then the second may be ignored
  * Ex: capacity factor (768 Kbps or 10 Mbps) versus TCP version factor (Reno or Sack)
- Sample variance of \(y\)
  \[s_y^2 = \frac{\sum (y_i - \bar{y})^2}{(2^2 - 1)}\]
- With numerator being total variation, or Sum of Squares Total (SST)
  \[SST = \sum (y_i - \bar{y})^2\]

Allocation of Variation (2 of 3)
- For a 2\(^2\) design, variation is in 3 parts:
  \[SST = 2q_{a}^2 + 2q_{b}^2 + 2q_{ab}^2\]
  (Derivation 17.1, p.287)
- Portion of total variation:
  - of \(a\) is \(2q_{a}^2\)
  - of \(b\) is \(2q_{b}^2\)
  - of \(ab\) is \(2q_{ab}^2\)
  - Thus, \(SST = SSA + SSB + SSAB\)
- And fraction of variation explained by \(a\):
  \[SSA/SST\]
  * Note, may not explain the same fraction of variance since that depends upon errors
Allocation of Variation (3 of 3)

- In the memory-cache study
  \[ y = \frac{1}{4} (15 + 55 + 25 + 75) = 40 \]
- Total variation
  \[ 2y(y-y)^2 = (25^2 + 15^2 + 15^2 + 35^2) = 2100 = 4\times20^2 + 4\times10^2 + 4\times5^2 \]
- Thus, total variation is 2100
  - 1600 (of 2100, 76%) is attributed to memory
  - 400 (of 2100, 19%) is attributed to cache
  - Only 100 (of 2100, 5%) is attributed to interaction
- This data suggests exploring memory further and not spending more time on cache (or interaction)
  (That was for 2 factors. Extend to k next)

General 2^k Factorial Designs (1 of 4)

- Can extend same methodology to k factors, each with 2 levels \( \rightarrow \) Need 2^k experiments
  - k main effects
  - (k choose 2) two factor effects
  - (k choose 3) three factor effects...
- Can use sign table method
  (Show with example, next)

General 2^k Factorial Designs (2 of 4)

- Example: design LISP machine
  - Cache, memory and processors
  - Memory (a) 4 Mbytes 16 Mbytes
  - Cache (b) 1 Kbytes 2 Kbytes
  - Processors (c) 1 2
- The 2^3 design and MIPS perf results are:

<table>
<thead>
<tr>
<th>Cache (b)</th>
<th>One proc (c) Two proc</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 KB</td>
<td>14 46 22 58</td>
</tr>
<tr>
<td>2 KB</td>
<td>10 50 34 86</td>
</tr>
</tbody>
</table>

General 2^k Factorial Designs (3 of 4)

- \( q_a=10, q_b=5, q_c=20 \) and \( q_{ab}=5, q_{ac}=2, q_{bc}=3 \) and \( q_{abc}=1 \)
- \( \text{SST} = 2^3 (q_a + q_b + q_c + q_{ab} + q_{ac} + q_{bc} + q_{abc})^2 \)
  \[ = 8 (10^2 + 5^2 + 20^2 + 5^2 + 1^2 + 1^2 + 1^2 + 1^2) \]
  \[ = 800 + 25 + 400 + 25 + 1 + 1 + 1 + 1 \]
  \[ = 1656 \]
- The portion explained by the 7 factors are:
  - \( \text{mem} = 800/1656 (48%) \)
  - \( \text{cache} = 200/1656 (48%) \)
  - \( \text{proc} = 3200/1656 (2%) \)
  - \( \text{mem-proc} = 8/1656 (0.5%) \)
  - \( \text{mem-cache} = 200/1656 (12%) \)
  - \( \text{cache-proc} = 72/1656 (2.5%) \)

Outline

- Introduction
- Terminology
- General Mistakes
- Simple Designs
- Full Factorial Designs
  - 2^k Factorial Designs
  - 2^r Factorial Designs
2\(^r\) Factorial Designs

- No amount of experimentation can ever prove me right; a single experiment can prove me wrong. —Albert Einstein

- With 2\(^r\) factorial designs, not possible to estimate error since only done once
- So, repeat r times for 2\(^{kr}\) observations
- As before, will start with 2\(^{2r}\) model and expand
  - Two factors at two levels and want to isolate experimental errors
  - Repeat 4 configurations r times
- Gives you error term:
  \[ y = q_0 + q_a x_a + q_b x_b + q_{ab} x_a x_b + e \]
- Want to quantify e (Illustrate by example, next)

No amount of experimentation can ever prove me right; a single experiment can prove me wrong.
- Albert Einstein

2\(^r\) Factorial Design Errors (1 of 2)

- Previous cache experiment with r=3
- Mean y
- Ex: e_{11} = y_{11} - y_1 = 15 – 15 = 0
- SSE = 0^2 + 3^2 + (-3)^2 + (-3)^2 + 0^2 + 3^2 + 1^2 + 4^2 + (-2)^2 + (-2)^2 + 42 = 102

2\(^r\) Factorial Design Errors (2 of 2)

- Use sum of squared errors (SSE) to compute variance and confidence intervals
- Example:
  \[ SSE = \sum_{i=1}^{4} \sum_{j=1}^{r} (y_{ij} - \bar{y})^2 \]
- Ex: e_{11} = y_{11} - \bar{y}_1 = 15 – 15 = 0
- SSE = 0^2 + 3^2 + (-3)^2 + (-3)^2 + 0^2 + 3^2 + 1^2 + 4^2 + (-2)^2 + (-2)^2 + 42 = 102

2\(^r\) Factorial Allocation of Variation

- Total variation (SST)
  \[ SST = \sum (y_{ij} - \bar{y})^2 \]
- Can be divided into 4 parts:
  \[ \sum (y_{ij} - \bar{y})^2 = 2^2r q_a^2 + 2^2r q_b^2 + 2^2r q_{ab}^2 + \Sigma e_{ij}^2 \]
  - SST = SSA + SSB + SSAB + SSE
- Thus:
  - SSA, SSB, SSAB are variations explained by factors a, b and ab
  - SSE is unexplained variation due to experimental errors
  - Can also write SST = SSY-SS0 where SS0 is sum squares of mean
    (Derivation 18.1, p.296)

2\(^r\) Factorial Allocation of Variation Example

- For memory cache study:
  - SSY = 15^2 + 18^2 + 12^2 + ... + 75^2 + 81^2 = 27,204
  - SS0 = 2^2r q_0^2 = 12x412 = 20,172
  - SSA = 2^2r q_a^2 = 12x(21.5)^2 = 5547
  - SSB = 2^2r q_b^2 = 12x(9.5)^2 = 1083
  - SSAB = 2^2r q_{ab}^2 = 12x(5)^2 = 300
  - SSE = 27,204-2^2r(15^2+21.5^2+9.5^2+5^2)=102
  - SST = 5547 + 1083 + 300 + 102 = 7032
- Thus, total variation of 7032 divided into 4 parts:
  - Factor a explains 5547/7032 (78.88%), b explains 15.40%, ab explains 4.27%
  - Remaining 1.45% unexplained and attributed to error

Confidence Intervals for Effects

- Assuming errors are normally distributed, then y_{ij}s are normally distributed with same variance
- Since q_0, q_a, q_b, q_{ab} are all linear combinations of y_{ij}'s (divided by 2^r), then they have same variance (divided by 2^r)
- Variance \( s^2 = SSE /(2^2(r-1)) \)
- Confidence intervals for effects then:
  \[ \pm t_{\alpha/2, (2^2(r-1))} \sqrt{s^2} \]
- If confidence interval does not include zero, then effect is significant
Confidence Intervals for Effects (Example)

- Memory-cache study, std dev of errors:
  \[ s_e = \sqrt{\frac{\text{SSE}}{2(2^r-1)}} = \sqrt{102/8} = 3.57 \]
- And std dev of effects:
  \[ s_i = \frac{s_e}{\sqrt{2^r}} = \frac{3.57}{3.47} = 1.03 \]
- The t-value at 8 degrees of freedom and 95% confidence is 1.86
- Confidence intervals for parameters:
  \[ q_i \pm (1.86)(1.03) = q_i \pm 1.92 \]
- \( q_0 \) (18.66, 35.92)
- \( q_a \) (19.58, 23.41)
- \( q_b \) (7.58, 11.41)
- \( q_{ab} \) (3.08, 6.91)
- Since none include zero, all are statistically significant

Confidence Intervals for Predicted Responses (1 of 2)

- Mean response predicted
  \[ y = q_0 + q_a x_a + q_b x_b + q_{ab} x_a x_b \]
- If predict mean from \( m \) more experiments, will have same mean but confidence interval on predicted response decreases
- Can show that std dev of predicted \( y \) with \( m \) more experiments:
  \[ s_y = s_e \sqrt{\frac{1}{n_{eff}} + \frac{1}{m}} \]
  - Where \( n_{eff} = \frac{2^r}{1 + df} \)
- In 2 level case, each parameter has 1 df, so \( n_{eff} = \frac{2^r}{5} \)

Confidence Intervals for Predicted Responses Example (1 of 2)

- Mem-cache study, for \( x_a = -1, x_b = -1 \)
- Predicted mean response for future experiment:
  - \( y_1 = q_0 - q_a - q_b + q_{ab} = 41 - 21.5 + 1 = 15 \)
- Std dev = \( 3.57 \times \sqrt{\frac{5}{12} + 1} = 4.25 \)
- Using \( t_{0.95;8} = 1.86 \), 90% conf interval:
  - 15 ± 1.86 x 4.25 = (8.09, 22.91)

Confidence Intervals for Predicted Responses Example (2 of 2)

- Predicted Mean Response for Large Number of Experiments:
  - Std dev = \( 3.57 \times \sqrt{\frac{5}{12}} = 2.30 \)
  - The confidence interval:
    - 15 ± 1.86 x 2.30 = (10.72, 19.28)

Confidence Intervals for Predicted Responses (2 of 2)

- A 100(1-\( \alpha \))% confidence interval of response:
  - \( y \pm t_{1-\alpha/2; 2^r-2} \cdot s_y \)
- Two cases are of interest:
  - Std dev of one run (\( m = 1 \))
    - \( s_1 = s_e \sqrt{\frac{5}{12} + 1} \)
  - Std dev for many runs (\( m = \infty \))
    - \( s_1 = s_e \sqrt{\frac{5}{12}} \)

Confidence Intervals for Predicted Responses Example (2 of 2)

- Predicted mean response for 5 future experiments:
  - Std dev = \( 3.57 \times \sqrt{\frac{5}{12} + \frac{1}{5}} = 2.80 \)
  - 15 ± 1.86 x 2.80 = (9.79, 20.29)