I. The Full Factorial Design

a. Setup and Definitions

1. If factor A has “a” levels, factor B has “b” levels, etc., then an experiment testing several factors is named something like an “a x b” or an “a x b x c” design. There are M=a·b or M=a·b·c, etc. different treatment combinations. A full factorial design has at least one subject per treatment combination, so the minimum number of subjects, N is equal to the product M.

2. Note the special case where there are k factors, each with 2 levels: M=2^k.

3. E.g., a=3, b=2, N=M=6 so there are N-1=5 df available (“corrected total”). Factor A uses 2 df, factor B uses 1 df, and so there are 2 df left for the error term, and we can test both main effects. The A*B interaction has (3-1)*(2-1)=2 df, so we cannot test for the interaction unless we have replicates, e.g., N=2*M=12. Replicated full factorial designs allow testing of all interactions.

b. Unreplicated Full Factorial Designs are used, especially in industrial experiments, whenever all of the following are true:

1. Running subjects is expensive
2. Many factors need to be tested, particularly with a simple goal of increasing (or decreasing) the mean outcome.
3. It is known that the k-way interaction is zero or negligible

With k 2-level factors we have N=M=2^k, and we need k df for main effects, k choose 2 df for two-way interactions, etc. This leads to the fact that there are not enough df to perform inference on the k-way interaction (i.e., residual df=0).

c. Replicated Full Factorial Designs use, say, r, subjects for each treatment combination for a total of N=r*M subjects and r*M-1 df. Higher r gives lower coefficient SEs.

1. E.g., a 2^k · 2 design has 2(2^k)-1 “corrected total” df, of which k are used for the main effects, leaving 2(2^k)-1-k df for interactions and/or error terms.

<table>
<thead>
<tr>
<th>k</th>
<th>cells</th>
<th>N</th>
<th>Corrected Total df</th>
<th>Main effects df</th>
<th>Remaining df after all 2-way interactions</th>
<th>Remaining df after all interactions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>8</td>
<td>7</td>
<td>2</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>16</td>
<td>15</td>
<td>3</td>
<td>12</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>32</td>
<td>31</td>
<td>4</td>
<td>27</td>
<td>21</td>
</tr>
<tr>
<td>5</td>
<td>32</td>
<td>64</td>
<td>63</td>
<td>5</td>
<td>58</td>
<td>48</td>
</tr>
<tr>
<td>6</td>
<td>64</td>
<td>128</td>
<td>127</td>
<td>6</td>
<td>121</td>
<td>106</td>
</tr>
</tbody>
</table>

1
2. E.g., a $3^k \cdot 2$ design has $2(3^k)-1$ “corrected total” df, and k are used for the main effects, leaving $2(3^k)-1$ k df for interactions and/or error terms.

<table>
<thead>
<tr>
<th>k</th>
<th>cells</th>
<th>N</th>
<th>Corrected Total df</th>
<th>Main effects df</th>
<th>Remaining df after all 2-way interactions</th>
<th>Remaining df after all interactions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>6</td>
<td>5</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>18</td>
<td>17</td>
<td>4</td>
<td>13</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>27</td>
<td>54</td>
<td>53</td>
<td>6</td>
<td>47</td>
<td>35</td>
</tr>
<tr>
<td>4</td>
<td>81</td>
<td>162</td>
<td>161</td>
<td>8</td>
<td>153</td>
<td>129</td>
</tr>
<tr>
<td>5</td>
<td>243</td>
<td>486</td>
<td>485</td>
<td>10</td>
<td>475</td>
<td>435</td>
</tr>
<tr>
<td>6</td>
<td>729</td>
<td>1458</td>
<td>1457</td>
<td>12</td>
<td>1445</td>
<td>1385</td>
</tr>
</tbody>
</table>

d. k 1-factor ANOVAs vs. 1 k-factor ANOVA
   1. Unreplicated factorial experiments vs. multiple 1-way ANOVAs with the same power (assuming no interactions)

   With 2 levels per factor, one k-factor ANOVA requires $2^k$ subjects.

   As an example, a 5 factor unreplicated ANOVA has 32 subjects. For 1-factor experiments with similar power I can calculate that we need 9 subjects each, or $5 \times 9 = 45$ subjects for all five 1-factor experiments. This is a $1-32/45 = 29\%$ improvement in efficiency for the k-way ANOVA.

   Conclusion: Fisher’s development of factorial experiments increased efficiency. The main downside of too many factors in one experiment is the increased chance for mistakes, including lost cells.

   2. Replicated factorials: more power and estimation of the k-way interaction.

e. Software for setup of factorial experiments
   1. In R, use package FrF2:

   E.g., to setup an unreplicated full factorial with three 2-level factors use:

   ```r
   library(FrF2)
   des = FrF2(nruns=2^3, nfactors=3, default.levels=c(0,1))
   design.info(des)
   dtf = cbind(as.data.frame(des), Y = NA) # for outcome
   dtf
   ```

   E.g., to setup a duplicated full factorial with three 2-level factors use:

   ```r
   des = FrF2(nruns=2^3, nfactors=3, replications=2, default.levels=c(0,1))
   design.info(des)
   dtf = cbind(as.data.frame(des), Y = NA) # for outcome
   ```

   Note that each time you run the above code, the order of testing changes. Randomized order of testing reduces confounding due to temporal drift.
2. In SAS, use PROC PLAN (possibly followed by PROC OPTEX)  
   E.g., to setup an unreplicated full factorial with 3 factors use  
   
   ```sas
   PROC PLAN;
   FACTORS a=2 b=2 c=2;
   OUTPUT OUT=myLib.uff3
   a cvals=('Hi' 'Lo') b cvals=('Hi' 'Lo')
   c cvals=('Hi' 'Lo');
   RUN;
   
   TITLE "Duplicated 2^3 full factorial";
   PROC PLAN;
   FACTORS a=2 b=2 c=2 dup=2;
   OUTPUT OUT=myLib.dff3
   a cvals=('Hi' 'Lo') b cvals=('Hi' 'Lo')
   c cvals=('Hi' 'Lo');
   RUN;
   
   Although PROC PLAN randomizes the level ordering for each factor, it changes only one factor at a time. Here is one way to completely randomize the treatment ordering:
   
   ```sas
   DATA myLib.dff3;
   SET myLib.dff3;
   order = RAND("UNIFORM");
   RUN;
   PROC SORT DATA=myLib.dff3;
   BY order;
   RUN;
   DATA myLib.dff3; SET myLib.dff3(DROP=order); RUN;
   PROC PRINT DATA=myLib.dff3; RUN;
   ```
   
   f. Analysis: multi-way ANOVA  
g. Example: Bread quality from a panel of consumers is the DV, and the IVs are high/low fraction whole wheat, rising temperature, amount of yeast, and baking temperature. Replication is used to allow testing all degrees of interaction.

   h. “Google setting”: Many replicates: OK to randomly assign treatment combinations. But: `table(sapply(1:1000, function(dummy) length(table(sample(1:16,64,rep=T)))))`  

   III. Blocking

   a. Blocking means creation of “grouping” factors usually based on non-quantitative similarities among subjects and usually when a treatment x block interaction is not expected. Also, we are usually not interested in estimating differences among blocks. Blocks can be farmers’ fields, machines in a factory, hospitals or doctors in a city, or even individual people (where treatment is repeated on, say, each eye or leg or on various sections of skin surface).
b. Blocking is closely related to random effects models. Generally when the number of blocks is small, it is considered not justified to model the distribution of the block effects and they are typically estimated individually as fixed effects. When a larger number of blocks are used, we don’t want to waste df estimating individual block effects, and we typically use random effects models to only estimate a single variance parameter.

c. If all levels of all other factors cannot be tested within each block, special care must be taken in assigning treatment combinations to blocks to avoid confounding of block and treatment effects. E.g., if all the higher elevation fields (blocks) with better drainage receive fertilizer A and all of the lower fields receive fertilizer B, then fertilizer and drainage will be confounded.

d. Blocking tends to reduce residual variability and increase power.

e. **Randomized Complete Block Designs**
   Complete blocks include all treatment combinations in every block. In the simplest form, all treatments are present once in each block, and the number of blocks is the replication count for the study. We do need to randomize the order of treatments in each block to avoid confounding with order, e.g., machine drift over time when the blocks are different machines. The analysis is a two (or higher) way ANOVA without interaction between blocks and treatment factor(s). Assuming the treatment factors don’t interact with the block factor, we get additional power and narrower confidence intervals with no additional complications, i.e., design generation, power, and contrasts are simple extensions of what we saw before.

f. **Latin Square Designs**
   A Latin Square design can be used when we have J treatments and want to block on two other factors, and these factors can be arranged to each have J levels. Label a J x J square with the levels of the two blocking factors, and then assign the J treatments such that no row or column includes any treatment more the once (like a Sudoku). This prevents confounding of treatment and the factors while decreasing sample size/increasing power. The analysis is a three-way ANOVA without interaction. One example is 4 fertilizers applied to a 4x4 grid constructed on a (presumably non-uniform) farm field, where the two factors are unquantified field variations in soil characteristics, light, water drainage, etc. Another example is 3 methods of treating cloth, tested on three machines, where order of testing is the second block variable. For any particular Latin Square, it is always possible to replicate by constructing additional squares, in which case some interactions may be testable. These designs are “complete” because each block has all treatment levels. Without replication, only main effects can be tested, so non-negligible interactions destroy the validity of the results.
Example: An oil company tested four different blends of gasoline for fuel efficiency, blocking for the effects of four different drivers and four different models of cars. Fuel efficiency was measured in miles per gallon (mpg) after driving cars over a standard course. (The data are from Ott: Statistical Methods and Data Analysis, 4th ed., Duxbury, 1993, page 866.)

Fuel Efficiencies (mpg) For 4 Blends of Gasoline
(Blends Indicated by Letters A-D)

<table>
<thead>
<tr>
<th>Driver</th>
<th>Car I</th>
<th>Car II</th>
<th>Car III</th>
<th>Car IV</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>D 15.5</td>
<td>B 33.9</td>
<td>C 13.2</td>
<td>A 29.1</td>
</tr>
<tr>
<td>2</td>
<td>B 16.3</td>
<td>C 26.6</td>
<td>A 19.4</td>
<td>D 22.8</td>
</tr>
<tr>
<td>3</td>
<td>C 10.8</td>
<td>A 31.1</td>
<td>D 17.1</td>
<td>B 30.3</td>
</tr>
<tr>
<td>4</td>
<td>A 14.7</td>
<td>D 34.0</td>
<td>B 19.7</td>
<td>C 21.6</td>
</tr>
</tbody>
</table>

DATA gas;
  INFILE "gas.dat" FIRSTOBS=2;
  INPUT driver car blend $1. mpg;
RUN;
TITLE "Ott fuel efficiency experiment";
PROC GLM DATA=gas;
  CLASS driver car blend;
  MODEL mpg = driver car blend;
  LSMEANS blend / ADJUST=TUKEY;
RUN;

Dependent Variable: mpg

<table>
<thead>
<tr>
<th>Source</th>
<th>DF</th>
<th>SS</th>
<th>Mean Square</th>
<th>F Value</th>
<th>Pr &gt; F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>9</td>
<td>851.791</td>
<td>94.643</td>
<td>23.85</td>
<td>0.0005</td>
</tr>
<tr>
<td>Error</td>
<td>6</td>
<td>23.809</td>
<td>3.968</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cor. Tot.</td>
<td>15</td>
<td>875.600</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source</th>
<th>DF</th>
<th>Type I SS</th>
<th>Mean Square</th>
<th>F Value</th>
<th>Pr &gt; F</th>
</tr>
</thead>
<tbody>
<tr>
<td>driver</td>
<td>3</td>
<td>5.897</td>
<td>1.966</td>
<td>0.50</td>
<td>0.6987</td>
</tr>
<tr>
<td>car</td>
<td>3</td>
<td>736.912</td>
<td>245.637</td>
<td>61.90</td>
<td>&lt;.0001</td>
</tr>
<tr>
<td>blend</td>
<td>3</td>
<td>108.982</td>
<td>36.327</td>
<td>9.15</td>
<td>0.0117</td>
</tr>
</tbody>
</table>

Least Squares Means for effect blend
Pr > |t| for H0: LSMean(i)=LSMean(j)

<table>
<thead>
<tr>
<th>Dependent Variable: mpg</th>
</tr>
</thead>
<tbody>
<tr>
<td>i/j</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
</tbody>
</table>

As for all Latin Squares, negligible interactions are required.
g. **Balanced Incomplete Block Design**

In an incomplete block, less than all treatment combinations are present in any given block. E.g., we wish to test 7 cloth dyes. The dyed cloth must be dried in ovens that hold 4 cloths. Outcome is dye intensity. Here is one way to setup the experiment:

<table>
<thead>
<tr>
<th>Cloth position in an oven</th>
<th></th>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oven 1</td>
<td>D1</td>
<td>D4</td>
<td>D7</td>
<td>D6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oven 2</td>
<td>D3</td>
<td>D6</td>
<td>D5</td>
<td>D7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oven 3</td>
<td>D7</td>
<td>D1</td>
<td>D2</td>
<td>D5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oven 4</td>
<td>D2</td>
<td>D3</td>
<td>D1</td>
<td>D5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oven 5</td>
<td>D2</td>
<td>D7</td>
<td>D3</td>
<td>D4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oven 6</td>
<td>D5</td>
<td>D3</td>
<td>D4</td>
<td>D1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oven 7</td>
<td>D2</td>
<td>D4</td>
<td>D5</td>
<td>D6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

There are 28 cloths tested. Each dye is used 4 times. Each row (oven) has $(4 \choose 2)=6$ dye pairs present for a total of $6*7=42$ pairs tested in the same oven, and each of the $(7 \choose 2)=21$ dye pairs is used in the same oven with any particular other dye 2 times. Although there is a great degree of “balance” in the treatment assignments, the dye effects and the oven effects are not orthogonal (which they are in a completely randomized block design).

**SAS** can generate the design like this:

```
DATA candidates;
  DO dye = 1 TO 7;
    DO oven = 1 TO 7;
      OUTPUT;
    END;
  END;
RUN;

PROC OPTEX DATA=candidates; /* not in University edition */
  CLASS dye oven;
  MODEL dye oven;
  GENERATE n=28;
  EXAMINE DESIGN;
  OUTPUT OUT=ibd;
RUN;
```

And analyze like this:

```
PROC GLM DATA=dye;
  CLASS oven dye;
  MODEL intensity = oven dye;
  RANDOM oven / TEST;
  LSMEANS dye / STDERR;
  ESTIMATE "Dye 1 vs. Dye 3" DYE 1 0 -1 0 0 0 0;
RUN;
```
III. The Fractional Factorial Design

a. **Goal**: reduce the number of subjects needed by sacrificing higher order interactions to **aliasing** (confounding)

b. Most commonly used with J=2 levels of each factor. Rarely used with J>3 levels.

c. Problem with $2^k$ full factorial designs: even with only k=8 2-level factors, you need 1024 “treatment combinations”, which is very difficult to carry out logistically in most settings, even if enough subjects are available. OK for, e.g., Google experiments.

d. Terminology: A $2^{(k-m)}$ design means that there are k factors, each with 2 levels, but only $2^{(k-m)}$ subjects (experimental units) are used. A given fractional factorial design has a **resolution** R. If the resolution is R (usually written with roman numerals), then if you choose any two numbers, $i_1$ and $i_2$, that add to R, then you know that $i_1$ interactions and $i_2$ interactions are confounded, but lower order ones are not. Note that $i=1$ means main effects here.

E.g., with R=3, main effects are confounded with 2-way interactions, so we require prior knowledge of no 2-way interactions to correctly interpret main effects.

E.g., with R=4, main effects are not confounded with 2-way interactions, but some 2-way interactions are confounded with others 2-way interactions, and main effects are confounded with 3-way interactions.

E.g., with R=5, main effects are not confounded with two-way or three way interactions, and two-way interactions are not confounded with any other two-way interactions.

e. In R use FrF2()

Example:

```r
require(FrF2)
des = FrF2(nruns=8, nfactors=4, default.levels=c(0,1))
design.info(des)
# $aliased$fi2
# [1] "AB=CD" "AC=BD" "AD=BC"
# $catlg.entry
# Design:  4-1.1
# 8 runs, 4 factors,
# Resolution  IV

Resolution=IV means main effects and 2-way I/A are not confounded, e.g.:

<table>
<thead>
<tr>
<th>“high”</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>AB</th>
<th>CD</th>
<th>AC</th>
<th>BD</th>
<th>AD</th>
<th>BC</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>C, D</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>A, D</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>A, B</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>B, D</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>A, B, C, D</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>B, C</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>A, C</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
</tbody>
</table>
```

dtf = data.frame(lapply(des, function(x)factor(x)))
dtf$Y = rnorm(8, (sapply(dtf, as.numeric)-1) %*% c(1.5,-2,2.5,-3), 0.2)
dtf
summary(aov(Y~A+B+C+D, dtf))

<table>
<thead>
<tr>
<th></th>
<th>Df</th>
<th>Sum Sq</th>
<th>Mean Sq</th>
<th>F value</th>
<th>Pr(&gt;F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>5.677</td>
<td>5.677</td>
<td>272.7</td>
<td>0.000483 ***</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>9.531</td>
<td>9.531</td>
<td>457.8</td>
<td>0.000223 ***</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>12.725</td>
<td>12.725</td>
<td>611.2</td>
<td>0.000145 ***</td>
</tr>
<tr>
<td>D</td>
<td>1</td>
<td>14.613</td>
<td>14.613</td>
<td>701.9</td>
<td>0.000118 ***</td>
</tr>
<tr>
<td>Residuals</td>
<td>3</td>
<td>0.062</td>
<td>0.021</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

summary(lm(Y-(A+B+C+D)^2, dtf))

Residuals:
ALL 8 residuals are 0: no residual degrees of freedom!

Coefficients: (3 not defined because of singularities)

|               | Estimate | Std. Error | t value | Pr(>|t|) |
|---------------|----------|------------|---------|----------|
| (Intercept)   | -0.1257  | NA         | NA      | NA       |
| A1            | 1.8094   | NA         | NA      | NA       |
| B1            | -2.0745  | NA         | NA      | NA       |
| C1            | 2.4321   | NA         | NA      | NA       |
| D1            | -2.5967  | NA         | NA      | NA       |
| A1:B1         | -0.2170  | NA         | NA      | NA       |
| A1:C1         | 0.1805   | NA         | NA      | NA       |
| A1:D1         | 0.2127   | NA         | NA      | NA       |
| B1:C1         | NA       | NA         | NA      | NA       |
| B1:D1         | NA       | NA         | NA      | NA       |
| C1:D1         | 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 7 and 0 DF,  p-value: NA

f. In SAS use PROC PLAN/FACTEX/OPTEX.
Example:
PROC FACTEX;
   FACTORS x1-x7;
   OUTPUT OUT=FullFac; /* FullFac has 128 rows */
RUN;

TITLE "Generating 2^(7-3) Fractional Factorial Design";
PROC FACTEX;
   FACTORS x1-x7;
   MODEL RESOLUTION=3;
   SIZE DESIGN=16;
   EXAMINE ALIASING;
   OUTPUT OUT=ff7m3; /* ff7m2 has 16 rows */
RUN;
Aliasing Structure
\[ x_1 = x_5 \times x_6 \]
\[ x_2 = x_5 \times x_7 \]
\[ x_3 = x_1 \times x_7 = x_2 \times x_6 = x_3 \times x_4 \]
\[ x_4 = x_1 \times x_7 = x_2 \times x_6 = x_3 \times x_4 \]
\[ x_5 = x_1 \times x_6 = x_2 \times x_7 \]
\[ x_6 = x_1 \times x_5 \]
\[ x_7 = x_2 \times x_5 \]
\[ x_1 \times x_2 = x_6 \times x_7 \]
\[ x_4 \times x_5 \]

Possible failure message: WARNING: No such design found.

IV. A Brief Review of Some Other Designs

a. Within-subjects ANOVA (Repeated Measures ANOVA)

b. Adaptive Response Surface Designs: For 2 or more quantitative IVs with a goal of finding the combination achieving the maximum DV mean. Most common is to fit a quadratic with 2-way interactions to a grid of points, then use calculus to find the predicted point of maximization. Repeat with a grid around the maximum points.

c. \( 3^{(k-p)} \) designs: Allow \( k \) factors with 3 levels using only \( 3^{(k-p)} \) experimental units. Allows detection of non-linearity.

d. Mixture designs: Allow a constraint, usually sum to 1.0 because x’s are fractions of the k components.

e. A- and D-Optimal Designs: Choose x values to maximize information provided by the experiment (minimize standard errors). D-optimal designs maximize the determinant of \( X'X \). E.g., in simple regression, putting half of the x’s at the lowest possible value and half at the highest possible value is D-optimal. Note that in this case you cannot check for non-linearity. A-optimal designs minimize the sum of the variances (trace of the inverse information matrix). In simple regression, the D-optimal design is also A-optimal.