ME 488: Design of Experiments

Lecture 4: Fractional Factorials

William 'Ike' Eisenhauer

Fall 2017

Department of Mechanical and Materials Engineering
Portland State University
Portland, Oregon 97223

wde@pdx.edu

What if you know ahead of time that some interactions are unimportant?

- How would you know?
- What is the impact of these being present?
- What can you gain by dropping them?
- What if instead of dropping them, we 're-purposed' them?

Idea behind Fractional Factorials

What if you know ahead of time that some interactions are unimportant?

- How would you know?
- What is the impact of these being present?
- What can you gain by dropping them?
- What if instead of dropping them, we 're-purposed' them?

Fractional Factorial

Fractional means we don't do ALL of the combinations, we choose specific ones not to use directly, to save resources.

Idea behind Fractional Factorials

What if you know ahead of time that some interactions are unimportant?

- How would you know?
- What is the impact of these being present?
- What can you gain by dropping them?
- What if instead of dropping them, we 're-purposed' them?

Fractional Factorial

Fractional means we don't do ALL of the combinations, we choose specific ones not to use directly, to save resources.

Quick check

Can you have a BALANCED Fractional Factorial?

Different than Model Selection

- In Model Selection:
  - Drop unimportant interactions to gain degrees of freedom
  - Maintain model significance [F and ${r_{adj}}^2$] and
  - Coefficient relevancy [β_i].
- In Fractional Factorial Modeling:
  - Re-purpose unimportant interactions to reduce experiments and enhance resource utilization
  - While this does have an impact of gaining degrees of freedom, there is a cost...
  - The cost of aliasing

How to

1. If you have k factors, 2^k runs are required in each replicate of a Balanced Full 2 Level Factorial Experiment
2. Assume you want to cut this down to $n = 2^{k-q}$ runs in each replicate, determine q
   - q needs to be a positive integer less than k
   - $q = 1$ means 'Half Fraction'
   - $q = 2$ means 'Quarter Fraction', etc.
3. Write out full factorial design for a $k - q$ factor design
4. Alias the extra q factors with higher order interactions that are assumed to be insignificant
   - If $q > 1$, use interactions with same number of factors
5. This is called a 2^{k-q} Fractional Design

How to: Example 2^{4-1} Design

This should be familiar to you by now..it was homework last week.

1. You have 4 factors, $2^4 = 16$ runs are required in each replicate of a Balanced Full 2 Level Factorial Experiment
2. Assume you want to cut this down to $n = 8$ runs in each replicate, determine q
   - $n = 2^{k-q} = 8$
   - $2^{4-q} = 8$
   - $q = 1$ or a 'Half Fraction'

How to: Example 2^{4-1} Design

3. Write out full factorial design for a $4 - 1 = 3$ factor design

Standard Order	*A*	*B*	*C*	*AB*	*AC*	*BC*	*ABC*
1	$-$	$-$	$-$	$+$	$+$	$+$	$-$
2	$-$	$-$	$+$	$+$	$-$	$-$	$+$
3	$-$	$+$	$-$	$-$	$+$	$-$	$+$
4	$-$	$+$	$+$	$-$	$-$	$+$	$-$
5	$+$	$-$	$-$	$-$	$-$	$+$	$+$
6	$+$	$-$	$+$	$-$	$+$	$-$	$-$
7	$+$	$+$	$-$	$+$	$-$	$-$	$-$
8	$+$	$+$	$+$	$+$	$+$	$+$	$+$

How to: Example 2^{4-1} Design

4. Alias the extra $q = 1$ factor with higher order interactions that are assumed to be insignificant
5. $D = ABC$ in this case

Standard Order	*A*	*B*	*C*	*AB* *CD*	*AC* *BD*	*BC* *AD*	*ABC* *D*
1	−	−	−	+	+	+	−
2	−	−	+	+	−	−	+
3	−	+	−	−	+	−	+
4	−	+	+	−	−	+	−
5	+	−	−	−	−	+	+
6	+	−	+	−	+	−	−
7	+	+	−	+	−	−	−
8	+	+	+	+	+	+	+

What is Aliasing?

Who is Andrew Rigdley?

What is Aliasing?

Two factors or interaction terms are set at identical levels throughout the entire experiment. In other words, two columns are 100% correlated. Example D is aliased with ABC and AB is aliased with CD, in the previous example.

- It does NOT imply the effects are the same
- Just that the evaluation of the effects can not be separated
- If the column ends up being significant, you can't tell, analytically, if it is due to one, the other, or both
- Used to create Fractional Factorial Designs

Other Terms

- **Design Generator**: The alias set(s) you first pick on purpose. D=ABC in the example
- **Identity Column (I)**: A column of all +1s. Any column multiplied by itself is I
- **Defining Word**: A set of columns whose product = I. In our example ABCD is a defining word
  - $ABC = D$
  - $DABC = DD$
  - $ABCD = D^2 = I$
- **Defining Relationship**: Set of All Defining Words
- **Aliasing Patterns**: Multiply a specific effect by defining relationship, to show which effects are aliased

Aliasing Pattern Example

CV Joints on Front Wheel Drive

In the design of a CV joint the ozone attack on the elastomer of the boot can result in failure. Critical Factors involved with the best protected boot are determined to be:

- A: Neoprene Level
- B: Antioxidant Type
- C: Antioxidant Level
- D: Antiozonant Type
- E: Antiozonant Level
- F: Wax Type
- G: Wax Level

The three-way interactions between A,B,C,D are chosen as generators for E(=ABC),F(=BCD), G(=ABD)

Answer

- How many runs would be required in each replicate for a BFFD (Balanced Full Factorial Design)?

Aliasing Pattern Example

Answer

- How many runs would be required in each replicate for a BFFD (Balanced Full Factorial Design)?
  - $2^k = 2^7 = 128$

Answer

- How many runs would be required in each replicate for a BFFD (Balanced Full Factorial Design)?
  - $2^k = 2^7 = 128$
- What type of Fractional Factorial design is needed to reduce runs/replicate to 16?
  - $2^{k-q} = 16$
  - $2^{7-q} = 16$
  - $log_2(2^{7-q}) = log_2(16)$
  - $7 - q = 4$
  - $q = 3$
  - A 2^{7-3}, an eighth fraction factorial

Answer (cont.)

What is the Defining Relationship? (This is where it gets ugly...)

Answer (cont.)

What is the Defining Relationship? (This is where it gets ugly...)

- First multiply each generator by itself to get the first set of defining words
- $I = E(ABC) = F(BCD) = G(ABD)$
- $I = ABCE = BCDF = ABDG$

Answer (cont.)

What is the Defining Relationship? (This is where it gets ugly...)

- First multiply each generator by itself to get the first set of defining words
- $I = E(ABC) = F(BCD) = G(ABD)$
- $I = ABCE = BCDF = ABDG$
- Then multiply each defining word by the others
- $I = (\text{ABCE})(\text{BCDF}) = (\text{ABCE})(\text{ABDG}) = (\text{BCDF})(\text{ABDG})$

Answer (cont.)

What is the Defining Relationship? (This is where it gets ugly...)

- First multiply each generator by itself to get the first set of defining words
- $I = E(ABC) = F(BCD) = G(ABD)$
- $I = ABCE = BCDF = ABDG$
- Then multiply each defining word by the others
- $I = \text{(ABCE)(BCDF)} = \text{(ABCE)(ABDG)} = \text{(BCDF)(ABDG)}$
- Then multiply all three defining words together
- $I = \text{(ABCE)(BCDF)(ABDG)}$

Answer (cont.)

What is the Defining Relationship? (This is where it gets ugly...)

- Then combine and clean up the mess
- $I = ABCE = BCDF = ABDG = AB^2C^2DEF = A^2B^2CDEG = AB^2CD^2FG = A^2B^3C^2D^2EFG$
- $I = ABCE = BCDF = ABDG = ADEF = CDEG = ACFG = BEFG$

Answer (cont.)

What interactions are aliased with main Antioxidant Level effect?

Answer (cont.)

What interactions are aliased with main Antioxidant Level effect?

- Antioxidant Level is C
- To find aliases, multiply it by the Defining Relationship
- $I = ABCE = BCDF = ABDG = ADEF = CDEG = ACFG = BEFG$
- $(C)I = (C)ABCE = (C)BCDF = (C)ABDG = (C)ADEF = (C)CDEG = (C)ACFG = (C)BEFG$
- $C = ABC^2E = BC^2DF = ABCDG = ACDEF = C^2DEG = AC^2FG = BCEFG$
- $C = ABE = BDF = ABCDG = ACDEF = DEG = AFG = BCEFG$

So it is aliased with 4 three-way interactions and 3 five-way interactions

What is Resolution?

Level of aliasing occurring in a design. Formally, for 2-level designs, the length of the shortest defining word, expressed as a Roman Numeral. Example, in the last example the shortest defining word was 4 (in fact they were all 4). So the resolution in that case was IV or R_{IV}.

- There is no Resolution of a full factorial experiment
- R_{II}: Main effects aliased with other main effects.
- R_{III}: Main effects not aliased with other main effects, but aliased with 2-way interactions.
- R_{IV}: Main effects not aliased with other main effects, or 2-way interactions. But 2-way interactions are aliased.
- R_V: Mains and 2-way interactions not aliased with each other.

Example

Defining word is

$I = ACD = BCE = ABCF = ABDE = BDF = AEF = CDEF$

- What is this design?
- What resolution does it have?
- If the cost per run is \$450 and you need 10 replicates, what is the cost savings of this design over a full factorial version?

Example

Defining word is

$I = ACD = BCE = ABCF = ABDE = BDF = AEF = CDEF$

- What is this design?
  - A, B, C, D, E, F = 6 factors
  - q = $log_2(definingwords + 1) = log_2(7 + 1) = 3$
  - 2^{6-3} design or an eighth fraction factorial
- What resolution does it have?
  - Shortest word in $I = ACD = BCE = ABCF = ABDE = BDF = AEF = CDEF$ = 3, therefore R_{III}

Example

- If the cost per run is \$450 and you need 10 replicates, what is the cost savings of this design over a full factorial version?
    - Number of runs per replicate required in BFF = 2^6 = 64
    - Number of runs per replicate required in $2^{6-3} = 2^3 = 8$
    - Total Saved = (64-8)(10)(\$450) = \$252,000

Not having to do this all by hand

The nice thing about R, is there are some add-on **packages** that have been developed to do this work for you. So now that you have the idea, you won't get stuck having to generate all these by hand.

Loading a Package

The walk-through will require you load some packages into R. This is a two step process.

1. **Install** the package into R
2. Use the package in your code by referring to its **library**

The two key packages we use for this DoE class are:

1. **DoE.base**
2. **FrF2**

Not having to do this all by hand

Install is easy, you only have to do this once. It downloads the package and installs it into your system. (Some GUI's for R have this as a menu option)

```
> install.packages("DoE.base")
> install.packages("FrF2")
```

Use by adding it to the library, you need to do this once every R session

```
> library("DoE.base")
> library("FrF2")
```

Loading Packages in R

Test if you have it installed right, this will make a Full Factorial

```
> FrF2(8,3)
```

Current Situation [Modified From: Schmidt(1994) p 8-23]

A factory that produces plastic injection molded parts is always concerned with 'shrinkage'. Typically, a die for a part is created oversize to allow for the plastic to shrink after it has been produced. In this instance, a new die has been created for a part.

Current Situation [Modified From: Schmidt(1994) p 8-23]

A factory that produces plastic injection molded parts is always concerned with 'shrinkage'. Typically, a die for a part is created oversize to allow for the plastic to shrink after it has been produced. In this instance, a new die has been created for a part.

Engineering Question

What are the proper process settings to produce a product with the proper dimensions?

Objective

Determine the effect of (A)Injection Velocity, (B)Cooling Time, (C)Barrel Zone Temperature, (D)Mold Temperature, (E) Hold Pressure, and (F) Back Pressure on shrinkage of the part in both the length (14.500 nominal) and width (9.380 nominal) at an $\alpha = 0.01$

Issues to be aware of

- Only two levels are allowed for each setting. There was a discussion regarding BZT (there are actually 6 control points), but after discussions it was agreed that in the interest of SOP simplification only two setup levels should be used.

Issues to be aware of

- Only two levels are allowed for each setting. There was a discussion regarding BZT (there are actually 6 control points), but after discussions it was agreed that in the interest of SOP simplification only two setup levels should be used.
- Only main effects are under review at this time

Issues to be aware of

- Only two levels are allowed for each setting. There was a discussion regarding BZT (there are actually 6 control points), but after discussions it was agreed that in the interest of SOP simplification only two setup levels should be used.
- Only main effects are under review at this time
- Due to automation level of this process the need for true randomization is not required.

Issues to be aware of

- Only two levels are allowed for each setting. There was a discussion regarding BZT (there are actually 6 control points), but after discussions it was agreed that in the interest of SOP simplification only two setup levels should be used.
- Only main effects are under review at this time
- Due to automation level of this process the need for true randomization is not required.
- Finance and Plant Control has indicated there are funds and time available to only do 40 tests.

Issues to be aware of

- Only two levels are allowed for each setting. There was a discussion regarding BZT (there are actually 6 control points), but after discussions it was agreed that in the interest of SOP simplification only two setup levels should be used.
- Only main effects are under review at this time
- Due to automation level of this process the need for true randomization is not required.
- Finance and Plant Control has indicated there are funds and time available to only do 40 tests.
- Plant Control reiterates that mold temperature adjustment requires a special technician who is only available for one change

Thought Questions

Full Factorial

How many runs are required per replicate if the decision is made to run the experiment as a Balance Full Factorial?

Thought Questions

Full Factorial

How many runs are required per replicate if the decision is made to run the experiment as a Balance Full Factorial?

Answer

There are 6 factors ($k = 6$) with 2 levels each. $2^k = 2^6 = 64$

THOUGHT QUESTIONS

Full Factorial

How many runs are required per replicate if the decision is made to run the experiment as a Balance Full Factorial?

Answer

There are 6 factors ($k = 6$) with 2 levels each. $2^k = 2^6 = 64$

Full Factorial?

Is a BFF experiment even required? Why or Why Not?

Full Factorial

How many runs are required per replicate if the decision is made to run the experiment as a Balance Full Factorial?

Answer

There are 6 factors ($k = 6$) with 2 levels each. $2^k = 2^6 = 64$

Full Factorial?

Is a BFF experiment even required? Why or Why Not?

Answer

No, there is no need for anything beyond main effects. So our goal is to minimize the runs per replicate in order to maximize the usage of our available resources.

Fractional Factorial

What would be the optimal Fractional Factorial Experiment Design?

Fractional Factorial

What would be the optimal Fractional Factorial Experiment Design?

Answer

We have 40 available total runs. We have a need to only concern ourselves with 6 effects (the main ones). As long as we don't so something stupid, like run a R_{II} experiment, we should be fine. So let's review our options.

Fractional Factorial

What would be the optimal Fractional Factorial Experiment Design?

Answer

- Half Fractional: $2^{6-1} = 32$ runs/replicate. Not going to work. Why?

Fractional Factorial

What would be the optimal Fractional Factorial Experiment Design?

Answer

- Half Fractional: $2^{6-1} = 32$ runs/replicate. Not going to work. Why?
- Quarter Fractional: $2^{6-2} = 16$ runs/replicate. Feasible

Fractional Factorial

What would be the optimal Fractional Factorial Experiment Design?

Answer

- Half Fractional: $2^{6-1} = 32$ runs/replicate. Not going to work. Why?
- Quarter Fractional: $2^{6-2} = 16$ runs/replicate. Feasible
- Eighth Fractional: $2^{6-3} = 8$ runs/replicate. Feasible

Thought Questions

Fractional Factorial

What would be the optimal Fractional Factorial Experiment Design?

Answer

- Half Fractional: $2^{6-1} = 32$ runs/replicate. Not going to work. Why?
- Quarter Fractional: $2^{6-2} = 16$ runs/replicate. Feasible
- Eighth Fractional: $2^{6-3} = 8$ runs/replicate. Feasible
- Sixteenth Fractional: $2^{6-4} = 4$ runs/replicate. Not going to work. Why?

Thought Questions

Fractional Factorial

What would be the optimal Fractional Factorial Experiment Design?

Answer

- Half Fractional: $2^{6-1} = 32$ runs/replicate. Not going to work. Why?
- Quarter Fractional: $2^{6-2} = 16$ runs/replicate. Feasible
- Eighth Fractional: $2^{6-3} = 8$ runs/replicate. Feasible
- Sixteenth Fractional: $2^{6-4} = 4$ runs/replicate. Not going to work. Why?

We could either go with 5 replicates of a 2^{6-3} or 2 replicates of a 2^{6-2} (and not use all the available runs). What is your choice?

Good Choice!

Let's look at what it means to use a 2^{6-3} in R.

Good Choice!

Let's look at what it means to use a 2^{6-3} in R.

```
> library("FrF2")
> FrF2(8,6)
   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
```

Good Choice!

Let's look at what it means to use a 2^{6-3} in R.

```
> library("FrF2")
> FrF2(8,6)
   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
```

You can get the same thing by `FrF2(nfactors=6,resolution=3)`.

Let's look at what that gets us

You can also get specific characteristics about the design

```
> design.info(FrF2(8,6))$type
[1] "FrF2"
> design.info(FrF2(8,6))$aliased
$legend
[1] "A=A" "B=B" "C=C" "D=D" "E=E" "F=F"

$main
[1] "A=BD=CE" "B=AD=CF" "C=AE=BF" "D=AB=EF" "E=AC=DF" "F=BC=DE
    "


$fi2
[1] "AF=BE=CD"
```

Let's look at what that gets us

You can also get specific characteristics about the design

```
> design.info(FrF2(8,6))$type
[1] "FrF2"
> design.info(FrF2(8,6))$aliased
$legend
[1] "A=A" "B=B" "C=C" "D=D" "E=E" "F=F"

$main
[1] "A=BD=CE" "B=AD=CF" "C=AE=BF" "D=AB=EF" "E=AC=DF" "F=BC=DE
    "

$fi2
[1] "AF=BE=CD"
```

There are no mains being aliased with other mains. Great! This design should work.

Levels

The actual settings for the level factors are:

Factor	Symbol	$-$	$+$
Injection Velocity	A	1.0	3.0
Cooling Time	B	30sec	40sec
Barrel Zones	C	'low'	'high'
Mold Temperature	D	100	150
Hold Pressure	E	200	1100
Back Pressure	F	50	150

So let's put that into `FrF2` to make it more meaningful

Continuing on from before

```
> dsg <-
FrF2(8,6,factor.names=list(InjVel=c("1.0","3.0"),CoolTime=c("
    30sec","40sec"),
BarrelZone=c("low","high"),MoldTemp=c("100","150"),HoldPres=c(
    "200","1100"),
BackPres=c("50","150")))
> dsg
  InjVel CoolTime BarrelZone MoldTemp HoldPres BackPres
1    3.0    30sec       high      100     1100       50
2    3.0    40sec        low      150      200       50
3    1.0    40sec       high      100      200      150
4    1.0    30sec        low      150     1100      150
5    3.0    40sec       high      150     1100      150
6    1.0    40sec        low      100     1100       50
7    3.0    30sec        low      100      200      150
8    1.0    30sec       high      150      200       50
class=design, type= FrF2
```

Continuing on from before

But what if we want the +1/-1 chart?

Continuing on from before

But what if we want the +1/-1 chart?

```
> desnum(dsg)
  InjVel CoolTime BarrelZone MoldTemp HoldPres BackPres
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
```

Alias recheck

```
> design.info(dsg)$aliased
$legend
[1] "A=InjVel"     "B=CoolTime"    "C=BarrelZone" "D=MoldTemp"
[5] "E=HoldPres"   "F=BackPres"

$main
[1] "A=BD=CE" "B=AD=CF" "C=AE=BF" "D=AB=EF" "E=AC=DF" "F=BC=DE
    "

$fi2
[1] "AF=BE=CD"
```

Replications

If you want FrF2 to generate replications for you, just tack on how many you want. [Note this list has been truncated]

Replications

If you want FrF2 to generate replications for you, just tack on how many you want. [Note this list has been truncated]

```
> FrF2(8,6,replications=5)
   run.no run.no.std.rp  A  B  C  D  E  F
1       1           3.1 -1  1 -1 -1  1 -1
2       2           2.1  1 -1 -1 -1 -1  1
3       3           5.1 -1 -1  1  1 -1 -1
4       4           7.1 -1  1  1 -1 -1  1
5       5           8.1  1  1  1  1  1  1
6       6           1.1 -1 -1 -1  1  1  1
...
22     22           1.3 -1 -1 -1  1  1  1
...
39     39           4.5  1  1 -1  1 -1 -1
40     40           8.5  1  1  1  1  1  1
class=design, type= FrF2
```

Repeating Replications

If you want to address the situation we have where we don't want to reset it each time, then add the `repeat.only=TRUE` option.[Note this list has been truncated]

Repeating Replications

If you want to address the situation we have where we don't want to reset it each time, then add the `repeat.only=TRUE` option.[Note this list has been truncated]

```
> FrF2(8, 6, replications=5, repeat.only=TRUE)
   run.no run.no.std.rp  A  B  C  D  E  F
1       1           7.1 -1  1  1 -1 -1  1
2       2           7.2 -1  1  1 -1 -1  1
3       3           7.3 -1  1  1 -1 -1  1
4       4           7.4 -1  1  1 -1 -1  1
5       5           7.5 -1  1  1 -1 -1  1
...
18     18           4.3  1  1 -1  1 -1 -1
19     19           4.4  1  1 -1  1 -1 -1
...
39     39           3.4 -1  1 -1 -1  1 -1
40     40           3.5 -1  1 -1 -1  1 -1
class=design, type= FrF2
```

How To

To set yourself up for collecting the data needed:

```
> dsg <- FrF2(8,6,factor.names=list(InjVel=c("1.0","3.0"),
    CoolTime=c("30sec","40sec"),
BarrelZone=c("low","high"),MoldTemp=c("100","150"),HoldPres=c(
    "200","1100"),
BackPres=c("50","150")), replications=5, repeat.only=TRUE)
> myCollector<-desnum(dsg)
> myCollector<-myCollector[order(myCollector[,4]),]
> write.csv(myCollector,file="myCollector.csv")
```

Creating a Data Collection Sheet

How To

To set yourself up for collecting the data needed:

```
> dsg <- FrF2(8,6,factor.names=list(InjVel=c("1.0","3.0"),
    CoolTime=c("30sec","40sec"),
BarrelZone=c("low","high"),MoldTemp=c("100","150"),HoldPres=c(
    "200","1100"),
BackPres=c("50","150")), replications=5, repeat.only=TRUE)
> myCollector<-desnum(dsg)
> myCollector<-myCollector[order(myCollector[,4]),]
> write.csv(myCollector,file="myCollector.csv")
```

1. Set up your chosen design, name factors, name levels, set for 5 replications repeating, and store it

How To

To set yourself up for collecting the data needed:

```
> dsg <- FrF2(8,6,factor.names=list(InjVel=c("1.0","3.0"),
    CoolTime=c("30sec","40sec"),
BarrelZone=c("low","high"),MoldTemp=c("100","150"),HoldPres=c(
    "200","1100"),
BackPres=c("50","150")), replications=5, repeat.only=TRUE)
> myCollector<-desnum(dsg)
> myCollector<-myCollector[order(myCollector[,4]),]
> write.csv(myCollector,file="myCollector.csv")
```

1. Set up your chosen design, name factors, name levels, set for 5 replications repeating, and store it
2. Extract just the +1/-1 Grid

How To

To set yourself up for collecting the data needed:

```
> dsg <- FrF2(8,6,factor.names=list(InjVel=c("1.0","3.0"), CoolTime=c("30sec","40sec"),
BarrelZone=c("low","high"),MoldTemp=c("100","150"),HoldPres=c("200","1100"),
BackPres=c("50","150")), replications=5, repeat.only=TRUE)
> myCollector<-desnum(dsg)
> myCollector<-myCollector[order(myCollector[,4]),]
> write.csv(myCollector,file="myCollector.csv")
```

1. Set up your chosen design, name factors, name levels, set for 5 replications repeating, and store it
2. Extract just the +1/-1 Grid
3. Sort the Grid by the fourth factor. Why?

Collect the Data and Bring it back in

How To

Use your sheet and add a column to store the response variables. Then read it back in

```
> myCollectedData <- read.delim(file="myCollector.csv")
```

You can down load the file I used from the course website. Note the sheet I used is a different design than what FrF2 generated. There are 16 equivalent designs, you can force them with generators.

Let's Confirm those insignificant interactions

```
> m<-lm(Length~InjVel*CoolTime*BarrelZone*MoldTemp*HoldPres*
    BackPres,data=myCollectedData)
> summary(m)
                                Pr(>|t|)
(Intercept)                      < 2e-16
InjVel                          7.69e-05
CoolTime                        8.80e-08
BarrelZone                      1.24e-07
MoldTemp                         < 2e-16
HoldPres                         < 2e-16
BackPres                        8.92e-07
InjVel:CoolTime                 1.47e-07
InjVel:BarrelZone                     NA
...
```

Let's Confirm those insignificant interactions

```
> m<-lm(Length~InjVel*CoolTime*BarrelZone*MoldTemp*HoldPres*
    BackPres,data=myCollectedData)
> summary(m)
                                Pr(>|t|)
(Intercept)                      < 2e-16
InjVel                          7.69e-05
CoolTime                        8.80e-08
BarrelZone                      1.24e-07
MoldTemp                         < 2e-16
HoldPres                         < 2e-16
BackPres                        8.92e-07
InjVel:CoolTime                 1.47e-07
InjVel:BarrelZone                     NA
...
```

There are a lot of them but you will notice that only one is critical. We are willing to live with that, but keep it in mind.

Main Effects Only

For Length

```
m<-lm(Length~InjVel+CoolTime+BarrelZone+MoldTemp+HoldPres+
    BackPres,data=myCollectedData)
summary(m)
Coefficients:
              Estimate Std. Error   t value Pr(>|t|)
(Intercept) 14.5006100  0.0004073 35603.674  < 2e-16 ***
InjVel       0.0007200  0.0003099     2.323   0.0265 *
CoolTime    -0.0003200  0.0003099    -1.033   0.3093
BarrelZone   0.0030800  0.0003099     9.939 1.89e-11 ***
MoldTemp     0.0034700  0.0003099    11.197 8.86e-13 ***
HoldPres     0.0068200  0.0003099    22.008  < 2e-16 ***
BackPres    -0.0002200  0.0003737    -0.589   0.5601
---
Residual standard error: 0.0009344 on 33 degrees of freedom
Multiple R-squared:  0.9603,^^IAdjusted R-squared:  0.953
F-statistic: 132.9 on 6 and 33 DF,  p-value: < 2.2e-16
```

Lets look at our specific main effects model only

Main Effects Only

For Width

```
Coefficients:
             Estimate Std. Error   t value Pr(>|t|)
(Intercept)  9.3577350  0.0003649 25646.820  < 2e-16 ***
InjVel      -0.0023050  0.0002776    -8.303 1.37e-09 ***
CoolTime     0.0034550  0.0002776    12.445 5.19e-14 ***
BarrelZone   0.0001550  0.0002776     0.558 0.580402
MoldTemp    -0.0035550  0.0002776   -12.805 2.37e-14 ***
HoldPres     0.0041950  0.0002776    15.110  < 2e-16 ***
BackPres    -0.0012200  0.0003348    -3.644 0.000914 ***
---
Residual standard error: 0.0008371 on 33 degrees of freedom
Multiple R-squared:  0.9518,^^IAdjusted R-squared:  0.9431
F-statistic: 108.7 on 6 and 33 DF,  p-value: < 2.2e-16
```

Interpretation

Length

- Mold Temperature needs to be low
- Barrel Zone needs to be low
- Hold Pressure needs to be low
- Injection Velocity, Cool time, and Back Pressure doesn't matter

Width

- Mold Temperature needs to be low
- Hold Pressure needs to be high
- Injection Velocity needs to be low
- Cool time needs to be high
- Barrel Zone and Back Pressure doesn't matter

Epilogue

25 verification runs at best condition

25 runs were done at the 'best settings', not one of the 25 were able to achieve the proper nominal tolerance for width of 9.380 ± 0.005

25 verification runs at best condition

25 runs were done at the 'best settings', not one of the 25 were able to achieve the proper nominal tolerance for width of 9.380 ± 0.005

Engineering Decision?

Using Generators

Out of the box, FrF2 tries to pick what it wants to use. If you have generator(s) you know you want to use, then you call tell FrF2 to do just that. Here is a 2^{4-1} with a D=ABC generator [like last week]

Using Generators

Out of the box, FrF2 tries to pick what it wants to use. If you have generator(s) you know you want to use, then you call tell FrF2 to do just that. Here is a 2^{4-1} with a D=ABC generator [like last week]

```
> dsg<-FrF2(nruns=8, nfactors=4,generator=c("ABC"))
> dsg
   A  B  C  D
1 -1 -1  1  1
2 -1  1 -1  1
3  1 -1  1 -1
4  1  1  1  1
5 -1  1  1 -1
6 -1 -1 -1 -1
7  1 -1 -1  1
8  1  1 -1 -1
class=design, type= FrF2.generators
```

Mid Term Next Week

Details

- Open Book - Any set of books you want
- Calculator and/or Laptop
- No Wireless
- Pencil/Pen - Blue or Black ONLY!! [No Quills!]
- Will have the entire class time for the exam
- Will not be required to know R syntax, but will be expected to interpret R output for ANOVA, LM, and FrF2
- Covers all material including this week!
- Homework Due in TWO Weeks

Format of Non-MC Questions

- Analyze Scenario
- Determine Alternatives
- Choose and Justify Alternative
- Evalute Situational Change Impact
- Decide to Change your Mind? Why or Why Not?
- Focus on being CONSISTENT. If you screw up, notate it, and let it go and carry it through
- DO NOT PANIC! Worse thing you can do is leave it blank! I can't read minds.