Nominal-the-Best Optimization

BIOE 498/598 PJ

Spring 2022

Location vs. Dispersion

- Sometimes we want to study the variation in the response, not the response itself.
- **Location** describes the central tendency of a response
  - Mean, median, mode
  - All of our models so far use response = location

Location vs. Dispersion

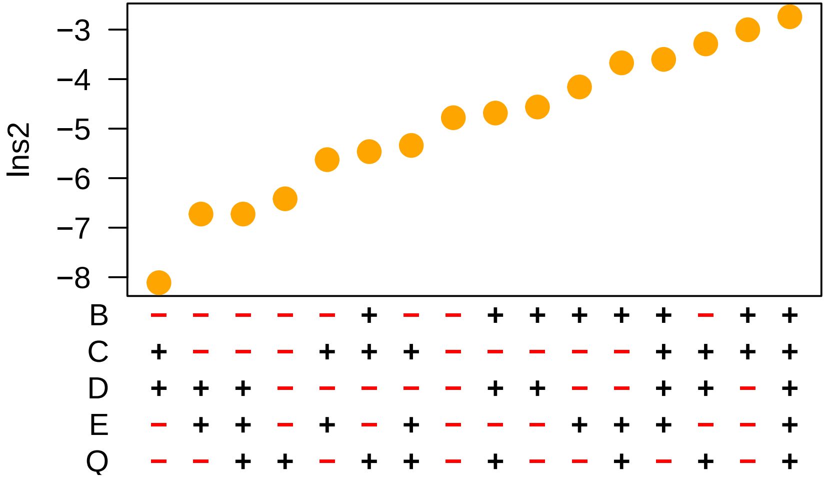
- Sometimes we want to study the variation in the response, not the response itself.
- **Location** describes the central tendency of a response
  - Mean, median, mode
  - All of our models so far use response = location
- **Dispersion** describes the spread of a response
  - Range, inter-quartile range (IQR), variance, standard deviation
- Location can be studied with unreplicated or replicated designs
- Studying dispersion always requires replicates

Studying dispersion

- The variance σ^2 is the natural statistic for studying dispersion with linear models fit by least-squares
- However, the sample variance s^2 is not a good response for studying σ^2
  - s^2 is left-censored ($s^2 \geq 0$)
  - s^2 follows a χ^2 distribution, not a normal distribution
- Both problems are fixed by modeling $\ln s^2$ instead of s^2
- Moreover, maximizing $-\ln s^2$ minimizes the variance, so we can keep the same maximization-based framework used for location models

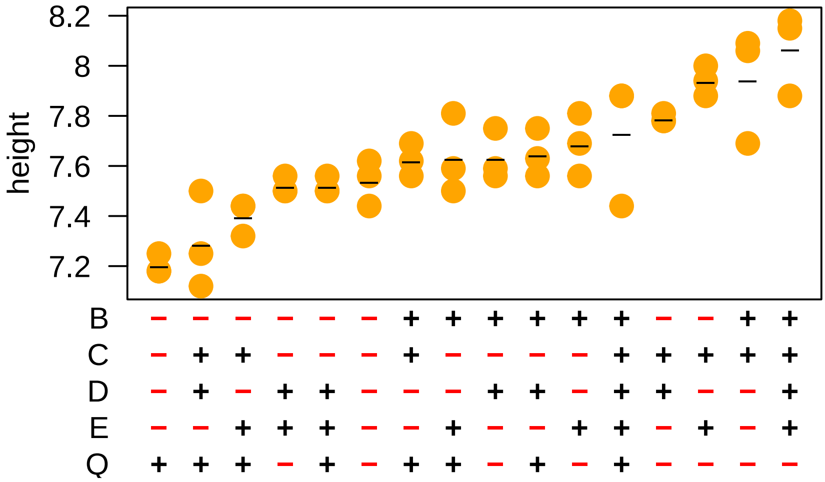
Visualizing the **dispersion**

```
farplot(disp, factors=c("B","C","D","E","Q"), response="lns2")
```



Visualizing the **data**

```
farplot(data, factors=c("B","C","D","E","Q"), response="height")
```



Building the model

```
disp_model <- lm(-lns2 ~ B+C+D+E+Q +
                     B:Q + C:Q + D:Q + E:Q + B:C + B:D + B:E,
                 data=disp)
```

Confounding in the 2^{5-1} design with I=BCDE:

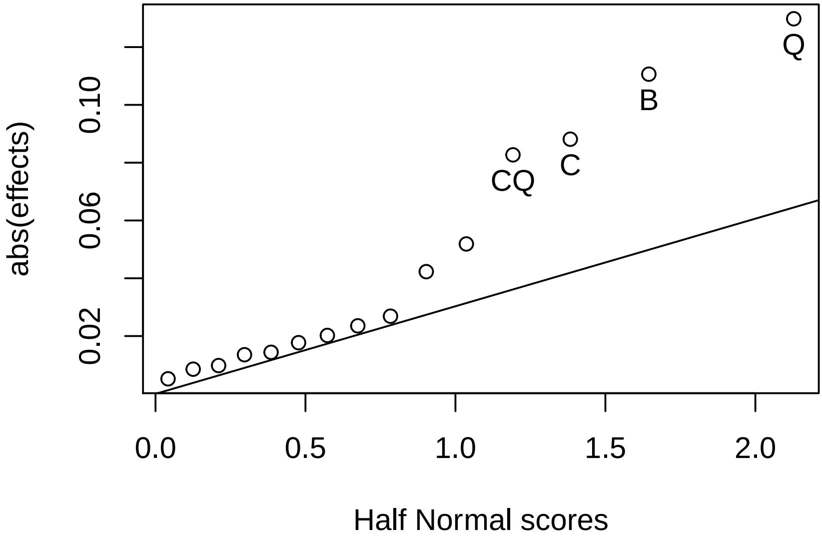
- main effects clear
- BQ
- CQ
- DQ
- EQ
- BC=DE
- BD=CE
- BE=CD

```
show_effects(disp_model, ordered="abs")
```

```
## (Intercept)    4.93131
##           B   -.94543
##         D:Q   -.55538
##         B:E   -.33523
##         C:Q   -.2989
##         B:Q    .29437
##           C   -.28434
##         B:D   -.21234
##           Q   -.13976
##           D    .12375
##           E   -.10777
##         E:Q   -.06457
##         B:C    .00079
```

Factors affecting **location** (spring height)

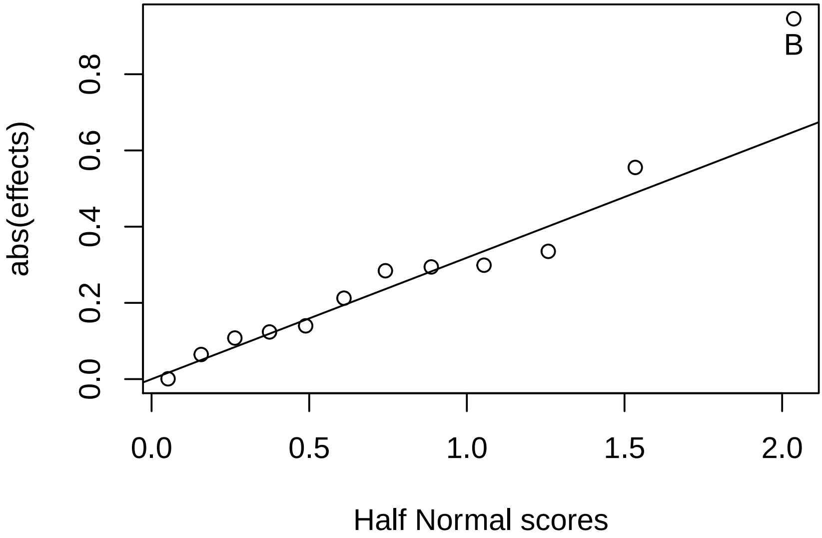
```
daewr::halfnorm(na.omit(get_effects(model)))
```




```
## zscore= 0.0417893 0.1256613 0.2104284 0.2967378 0.3853205 0.4770404
```

Factors affecting **dispersion** ($\ln s^2$)

```
daewr::halfnorm(get_effects(disp_model))
```




```
## zscore= 0.05224518 0.1573107 0.264147 0.3740954 0.4887764 0.6102946
```

Final Models

$$\text{height} = 7.64 + 0.11B + 0.09C - 0.13Q - 0.08CQ$$
$$-\ln s^2 = 4.93 - 0.95B$$

Final Models

$$\text{height} = 7.64 + 0.11B + 0.09C - 0.13Q - 0.08CQ$$
$$-\ln s^2 = 4.93 - 0.95B$$

- We want to minimize process variance (maximize $-\ln s^2$), but what about height?
- Often we have a *nominal value* for a response and need to balance reducing variance with keeping the response near the nominal value.

Final Models

$$\text{height} = 7.64 + 0.11B + 0.09C - 0.13Q - 0.08CQ$$
$$-\ln s^2 = 4.93 - 0.95B$$

- We want to minimize process variance (maximize $-\ln s^2$), but what about height?
- Often we have a *nominal value* for a response and need to balance reducing variance with keeping the response near the nominal value.

- In this case, we set $B = -$ and adjust the nominal value with C and Q.
- C and Q are called *adjustment factors* since they appear in the model for location but not dispersion.

Nominal-the-Best Optimization

1. Use the dispersion model to reduce variation.
2. Use adjustment factors to move the location near the nominal value.
3. If the location is too far off, repeat but reduce the variation less than before.

Nominal-the-Best Optimization

1. Use the dispersion model to reduce variation.
2. Use adjustment factors to move the location near the nominal value.
3. If the location is too far off, repeat but reduce the variation less than before.

Why it works

Given a nominal value t for a response y, our goal (using quadratic loss) is

$$\begin{aligned} \min \mathbb{E}(y-t)^2 &= \mathbb{E}[(y-\mathbb{E}(y)) + (\mathbb{E}(y)-t)]^2 \\ &= \mathrm{Var}(y) + (\mathbb{E}(y)-t)^2 \end{aligned}$$

Robust Parameter Design

- Factors can be split into two groups
  - **Control factors** can be changed easily
  - **Noise factors** are difficult or impossible to change

Robust Parameter Design

- Factors can be split into two groups
  - **Control factors** can be changed easily
  - **Noise factors** are difficult or impossible to change
- Robust Parameter Design finds settings for control factors that mitigate variation from noise factors.
- Mitigation is achieved through *noise* $\times$ *control* interactions.

Example: Manufacturing boxed cake mixes

- Control factors: Ingredients in the box $A, \ldots, D$.
- Noise factors: Controlled by the customer
  - E: Egg (small or large)
  - M: Milk (skim or 2%)
  - T: Oven temperature (340°–360°F)

Example: Manufacturing boxed cake mixes

- Control factors: Ingredients in the box $A, \ldots, D$.
- Noise factors: Controlled by the customer
  - E: Egg (small or large)
  - M: Milk (skim or 2%)
  - T: Oven temperature (340°–360°F)

Controlling for egg size variation (E) using baking soda (B)

Example: Manufacturing boxed cake mixes

- Control factors: Ingredients in the box $A, \ldots, D$.
- Noise factors: Controlled by the customer
  - E: Egg (small or large)
  - M: Milk (skim or 2%)
  - T: Oven temperature (340°–360°F)

Controlling for egg size variation (E) using baking soda (B)

Recall the definition of Int(EB):

$$\text{Int}(EB) = \frac{\text{ME}(E|B+) - \text{ME}(E|B-)}{2}$$

Mitigating noise trades optimality for robustness

Partial model for egg size E and baking soda B:

$$\text{taste} = \ldots + 0.2B - 0.7E + 0.4EB + \ldots$$

Mitigating noise trades optimality for robustness

Partial model for egg size E and baking soda B:

$$\text{taste} = \ldots + 0.2B - 0.7E + 0.4EB + \ldots$$

When $B = +$

- $E+ \Rightarrow \text{taste} = -0.1$
- $E- \Rightarrow \text{taste} = 0.5$

When $B = -$

- $E+ \Rightarrow \text{taste} = -1.3$
- $E- \Rightarrow \text{taste} = 0.9$

Mitigating noise trades optimality for robustness

Partial model for egg size E and baking soda B:

$$\text{taste} = \ldots + 0.2B - 0.7E + 0.4EB + \ldots$$

When $B = +$

- $E+ \Rightarrow \text{taste} = -0.1$
- $E- \Rightarrow \text{taste} = 0.5$

When $B = -$

- $E+ \Rightarrow \text{taste} = -1.3$
- $E- \Rightarrow \text{taste} = 0.9$

What should we do?

- The optimal cake has low baking soda and instructions to use a small egg (taste=0.9).
- If the customer uses a large egg, the taste drops a lot (-1.3).
- Using high baking soda gives a suboptimal taste (0.5 with small egg).
- Customers incorrectly using a large egg will not change taste as much (-0.1).

Designs to optimize robustness

- Noise mitigation relies on interactions between control and noise factors.

Designs to optimize robustness

- Noise mitigation relies on interactions between control and noise factors.
- Typically, a Resolution V design is needed.
- The noise$\times$control interactions are most important, so a clever choice of generator can work with Resolution IV.

Designs to optimize robustness

- Noise mitigation relies on interactions between control and noise factors.
- Typically, a Resolution V design is needed.
- The noise$\times$control interactions are most important, so a clever choice of generator can work with Resolution IV.

In the leaf spring experiment, the generator $E = BCD$ produced a 2^{5-1}_{IV} design with:

- main effects clear
- BQ
- CQ
- DQ
- EQ
- BC=DE
- BD=CE
- BE=CD

All the interactions with the noise factor Q (quench oil temperature) are clear.