Active Effects

BIOE 498/598 PJ

Spring 2022

Active effects

- An **active effect** has an effect size that is large enough to be **practically significant**.
- An **inactive effect** (or **inert effect**) does not have a practically significant effect size.

Active effects

- An **active effect** has an effect size that is large enough to be **practically significant**.
- An **inactive effect** (or **inert effect**) does not have a practically significant effect size.

How do we assess practical significance?

1. Any effect that causes a meaningful change based on our knowledge of the system or process.
   - A 3% increase in yield for a commodity chemical process might be significant financially.
   - A 3% decrease in tumor size after treatment might be insignificant to the patient.
2. Lacking insight about the system or process, we can estimate practical significance by *comparing* effect sizes in the same experiment.

Bioprocess conversion case study

Goal: Improve bioprocess conversion of switchgrass to biofuel

- Four process factors
  - S: Bacterial strain (strain A or B)
  - T: Temperature (30 °C or 37 °)
  - M: Mineral supplement (no or yes)
  - R: Stirring rate (fast or slow)
- Response: Percent conversion of carbon
- 2^4 factorial study $=$ 16 runs
- Unreplicated design
- Randomized run order

Step 1: Load the data

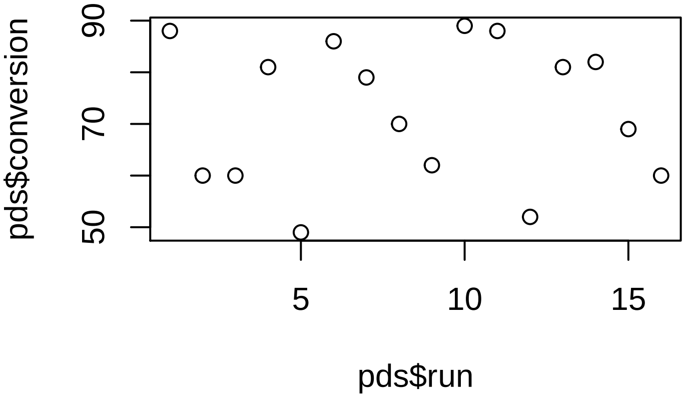
```
pds <- read.csv("ProcessDevelopmentStudy.csv")
head(pds)
```

```
##   run  S  T  M  R conversion
## 1   8 -1 -1 -1 -1         70
## 2   2  1 -1 -1 -1         60
## 3  10 -1  1 -1 -1         89
## 4   4  1  1 -1 -1         81
## 5  15 -1 -1  1 -1         69
## 6   9  1 -1  1 -1         62
```

Step 2: Look at the data

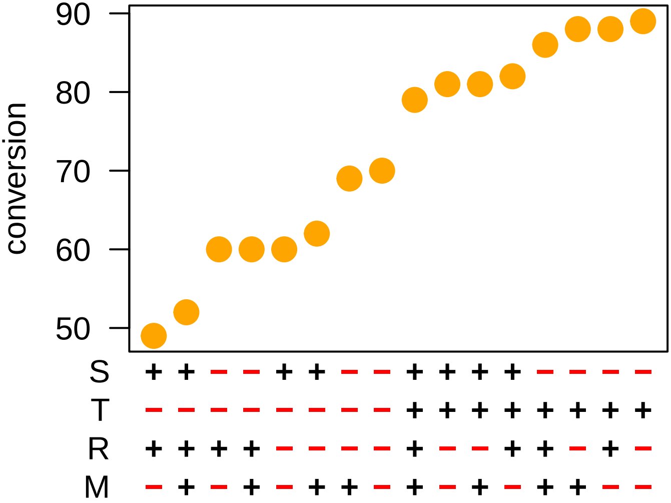
First, let's check if the response is correlated with run order.

```
plot(pds$run, pds$conversion)
```



Step 2: Look at the data

```
library(doetools)
farplot(pds, response="conversion", factors=c("S","T","R","M"))
```



Step 3: Fit a linear model

```
model <- lm(conversion ~ S*T*R*M, data=pds)
show_effects(model, scaling=2)
```

```
## (Intercept)    72.25
##           S    -8.
##           T    24.
##           R    -5.5
##           M     -.25
##         S:T     1.
##         S:R      .
##         T:R     4.5
##         S:M      .75
##         T:M    -1.25
##         R:M     -.25
##       S:T:R      .5
##       S:T:M     -.75
##       S:R:M     -.25
##       T:R:M     -.75
##     S:T:R:M     -.25
```

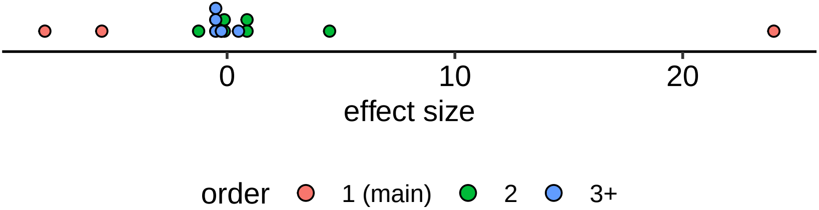
Ordering the effects by magnitude

```
show_effects(model, order="abs", scaling=2, intercept=FALSE)
```

```
##       T    24.
##       S    -8.
##       R    -5.5
##     T:R     4.5
##     T:M    -1.25
##     S:T     1.
##     S:M      .75
##   S:T:M     -.75
##   T:R:M     -.75
##   S:T:R      .5
##     R:M     -.25
## S:T:R:M     -.25
##       M     -.25
##   S:R:M     -.25
##     S:R      .
```

Visualizing effect sizes

```
effect_dotplot(model, scaling=2, binwidth=0.5)
```



Permutation tests

```r
red <- c(8,3,9,6)
black <- c(7,4,5,2)
test <- mean(red) - mean(black)
test
```

```
## [1] 2
```

Permutation tests

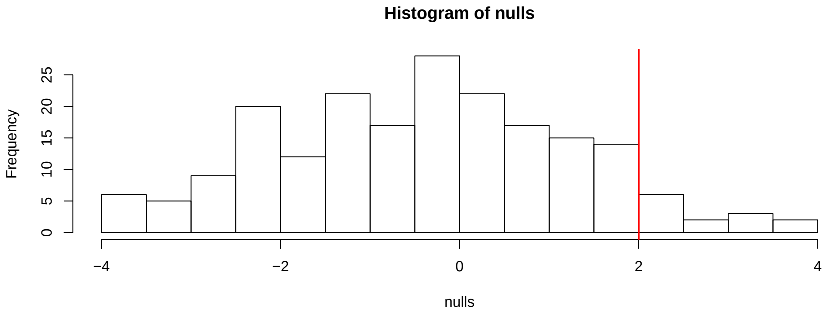
```r
red <- c(8,3,9,6)
black <- c(7,4,5,2)
test <- mean(red) - mean(black)
test
```

```
## [1] 2
```

```r
cards <- c(red, black)
N <- 200
nulls <- numeric(N)
for (i in 1:N) {
  idxs <- sample(1:8, 4)
  nulls[i] <- mean(cards[idxs]) - mean(cards[-idxs])
}
```

Permutation test results

```r
hist(nulls, n=sqrt(N))
abline(v=test, col="red", lwd=2)
```



Calculating a p-value from the null distribution.

```r
1 - sum(nulls <= test) / length(nulls)
```

```
## [1] 0.065
```

Why can we compare effect sizes in the same model?

- A *permutation test* creates a null distribution by randomly re-assigning data to groups.
- Each contrast in a factorial experiment is a permutation of the responses, so the *inactive* effect sizes create a null distribution.

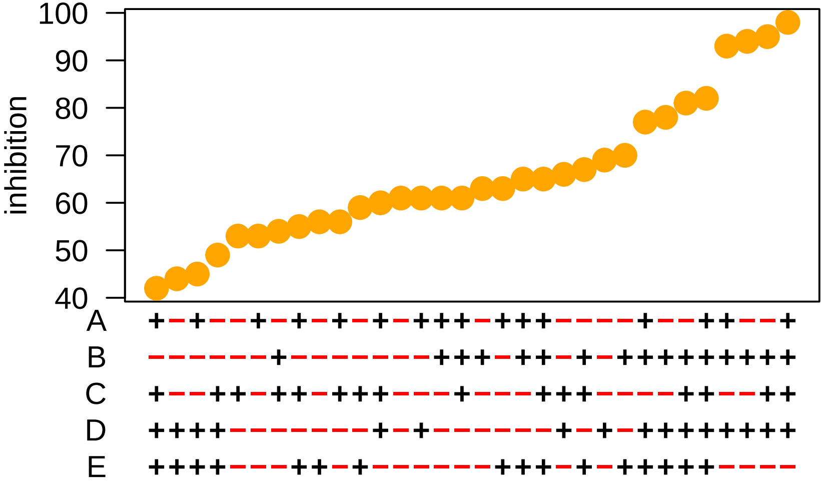
A larger (2^5) unreplicated study

```r
tumor <- read.csv("TumorInhibition.csv")
head(tumor)
```

```
##    A  B  C  D  E inhibition run half pb
## 1 -1 -1 -1 -1 -1         61   1    0  1
## 2  1 -1 -1 -1 -1         53   2    1  0
## 3 -1  1 -1 -1 -1         63   3    1  1
## 4  1  1 -1 -1 -1         61   4    0  0
## 5 -1 -1  1 -1 -1         53   5    1  0
## 6  1 -1  1 -1 -1         56   6    0  1
```

Step 2: Look at the data

```
farplot(tumor, response="inhibition", factors=c("A","B","C","D","E"))
```



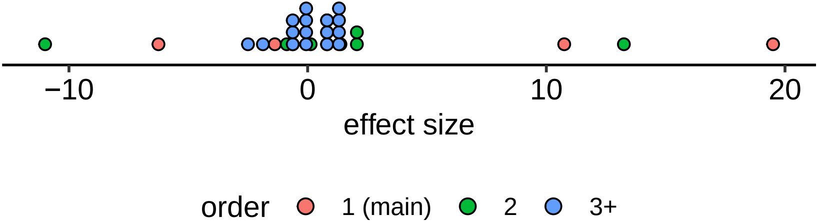
Step 3: Fit a linear model

```
model <- lm(inhibition ~ A*B*C*D*E, data=tumor)
show_effects(model, scaling=2, intercept=FALSE, order="abs", n=18)
```

```
##         B    19.5
##       B:D    13.25
##       D:E   -11.
##         D    10.75
##         E    -6.25
##     A:C:E    -2.5
##       C:D     2.125
##       B:E     2.
##     A:B:E    -1.875
##     A:B:C     1.5
##   A:B:C:E     1.5
##     A:B:D     1.375
##       A:B     1.375
##         A    -1.375
##     B:C:D     1.125
##   A:C:D:E     1.
##       A:D     -.875
##       B:C      .875
```

Visualizing effect sizes

```
effect_dotplot(model, scaling=2, binwidth=0.5)
```

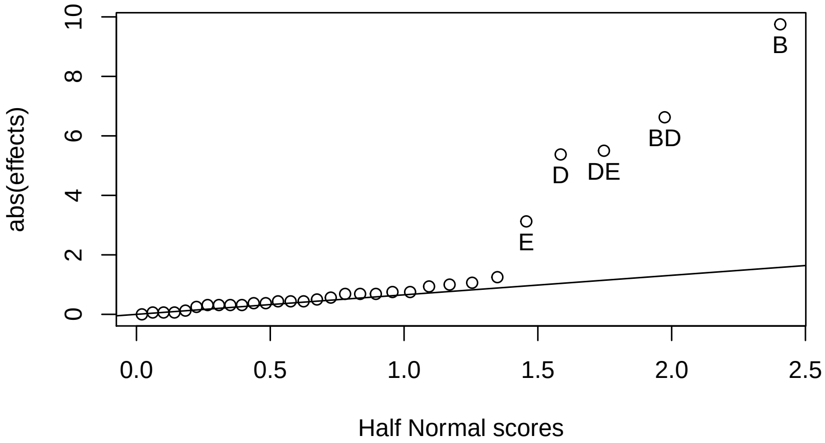


Half-normal plots

- With enough effects, the inactive factors will approximate a normal distribution.
- A *half-normal plot* displays the effect sizes and their associated probabilities.
- Inactive effects fall along a straight line, while active effects deviate.
- The half-normal plot uses the effect *magnitudes* since the sign depends **only** on how the $-$ and $+$ levels were assigned.

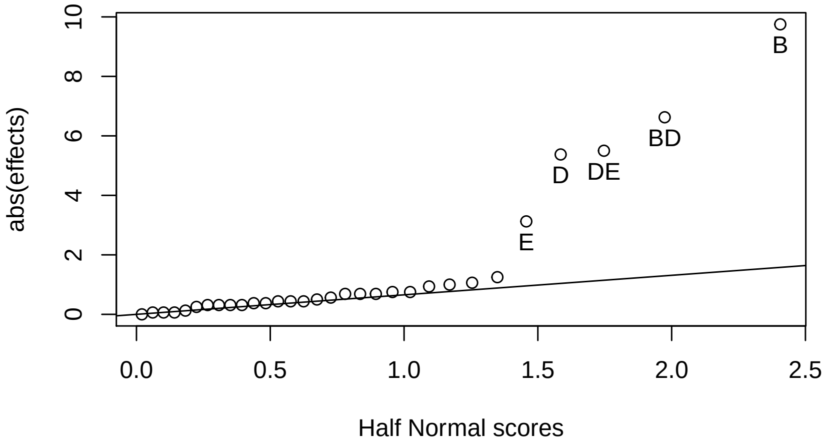
Selecting active effects with a half-normal plot

```
daewr::halfnorm(get_effects(model))
```



Selecting active effects with a half-normal plot

```
daewr::halfnorm(get_effects(model))
```



Active effects: factors B, D, & E and interactions BD & DE.