ANOVA

Testing more than 2 conditions

ANOVA

Today's goal:

Teach you about ANOVA, the test used to measure the difference between more than two conditions

Outline:

- Why anova?
- Contrasts and post-hoc tests
- ANOVA in R

Why ANOVA?

the problem of family-wise error, and how to deal with it

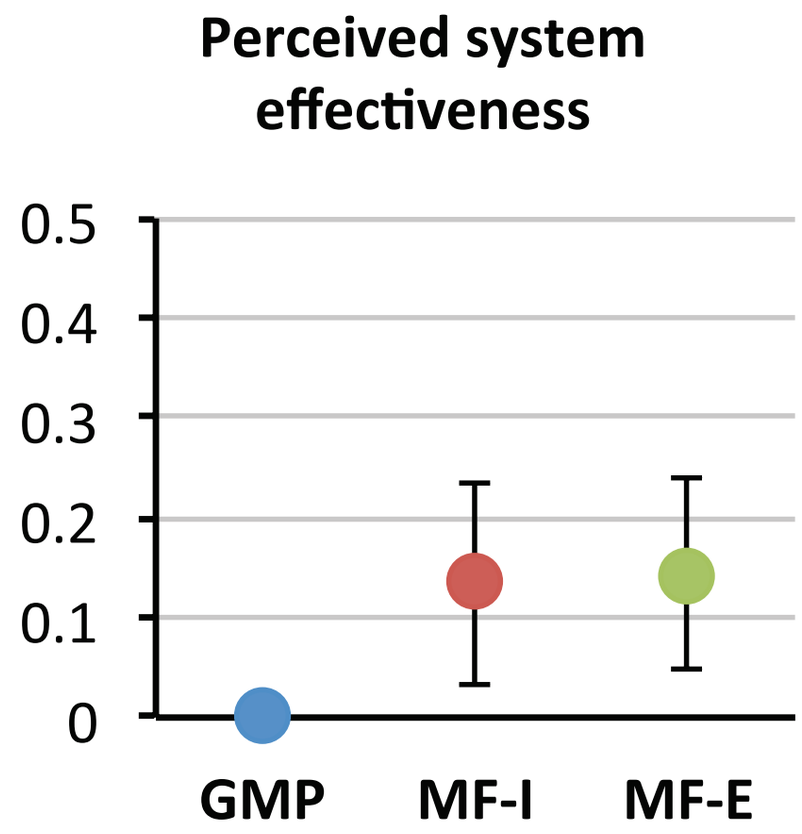
Why ANOVA?

Differences between >2 systems / groups:

Are there differences in perceived system effectiveness between these 3 algorithms?

First do an omnibus test, then post-hoc tests or planned contrasts

Family-wise error!



Family-wise error

One statistical test: is the observed effect is “real” or due to chance variation?

We cannot be 100% certain, so we take alpha = .05

1 out of every 20 significant results could be a mistake!

Test all possible pairs of 5 conditions: 10 tests!

AvB, AvC, AvD, AvE, BvC, BvD, BvE, CvD, CvE, DvE

Family-wise error

Each test finds a true effect 95% at the time. What is the chance of finding all true effects?

$$0.95^{10} = 0.599$$

What is the chance of making at least one mistake?

$$1 - 0.599 = 0.401$$

That's way higher than 5%!

Family-wise error

How to deal with this?

First, do an omnibus test to test if there is any effect

Then, test “planned contrasts”…

Carefully selected follow-up tests

…or “post-hoc tests”

All possible tests, but with a corrected level of alpha

Omnibus test

We test if there is any effect using the F-ratio

The ratio between the variance explained by the model and the residual variance

This tests whether the model is a significant improvement compared to using “no model”

“No model” = the grand mean

Sums of squares

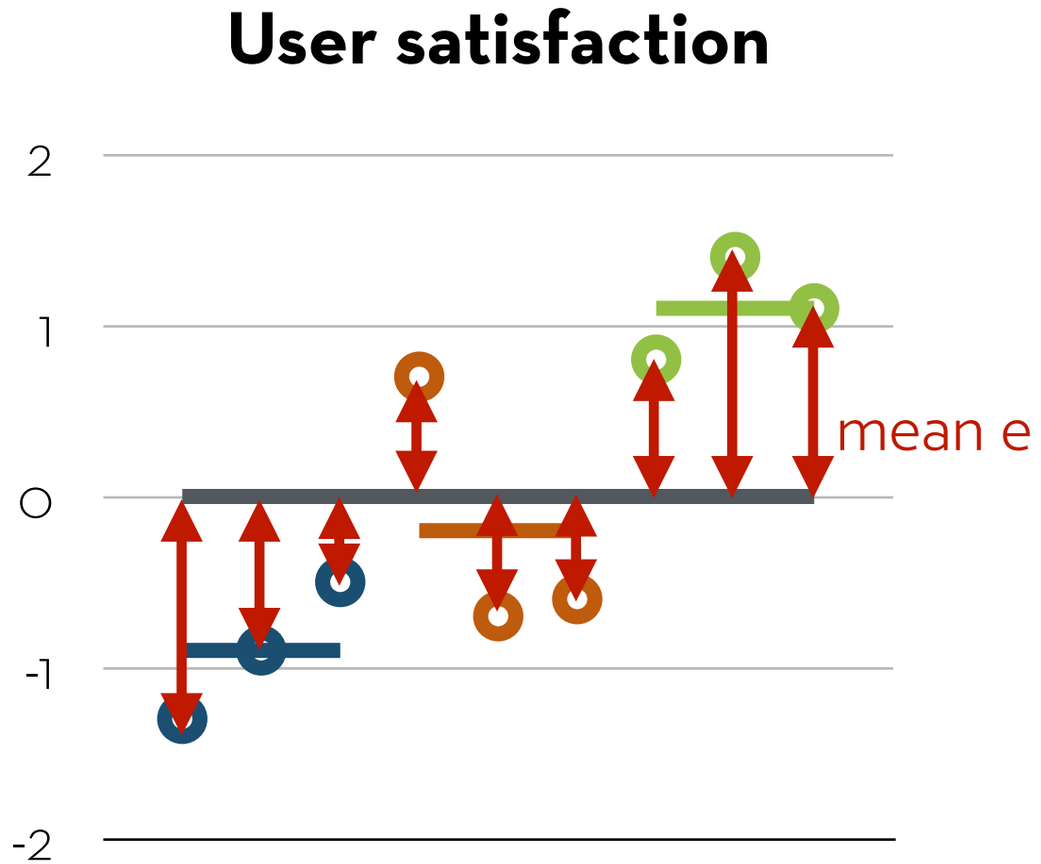
Total sum of squares (SSt)

squared e from the mean

$\sum(obs_i - grand\ mean)^2$

Note: s^2 is $\sum(obs_i - grand\ mean)^2/N-1$

Hence, $SSt = s^2(N-1)$



Sums of squares

Residual sum of sq. (SSr)

Squared e from the model

$\sum(obs_i - mean_k)^2$

Within each group k, s_k^2 is

$\sum(obs_i - mean_k)^2/N_k-1$

Hence, $SSr = \sum s_k^2(N_k-1)$

Sums of squares

Model sum of squares (SSm)

SSt – SSr

Or in terms of sums of sq.:

$\sum n_k(\text{mean}_k - \text{grand mean})^2$

Summed over k groups

Mean squares

SSm is based on k differences

So it increases with an increased number of groups (k)!

It has k–1 degrees of freedom

(k means, minus 1 grand mean)

Mean squares: MSm = SSm/dfm, where dfm = k–1

Mean squares

SSr is the sum of k variances

Each has N_k–1 degrees of freedom

Therefore, SSr has N–k degrees of freedom

(n values, minus k group means)

Mean squares: MSr = SSr/df_r, where df_r = N–k

F-ratio

F = MSm/MSr

It has two df parameters: dfm and dfr

It tests the question:

"How much did the model improve (over the grand mean), compared to the remaining error?"

Null hypothesis:

Ma = Mb = Mc = ...

If significant, there is a difference (but doesn't say where!)

Example

Grand mean: 3.467,
grand s^2: 3.124

SSt = 3.124*14 = 43.74,
SSr = (1.7+1.7+2.5)*4 = 23.60,
SSm = 43.74–23.60 = 20.14

MSm = 20.14/2 = 10.07,
MSr = 23.60/12 = 1.97

F-ratio = 10.07/1.97 = 5.11

with 2 and 12 df

	Placebo	Low	High
	3	5	7
	2	2	4
	1	4	5
	1	2	3
	4	3	6
gr means	2.2	3.2	5.0
s^2	1.7	1.7	2.5

It is all the same!

Multiple regression: $Y_i = a + b_1X_{1i} + b_2X_{2i} + e_i$

T-test: let's say you test system A vs B vs C

Choose a baseline (e.g. A)

Create X dummy variables for B and C:

$X_1 = 1$ for B, $X_1 = 0$ for A and C

$X_2 = 1$ for C, $X_2 = 0$ for A and B

It is all the same!

Multiple regression: $Y_i = a + b_1X_{1i} + b_2X_{2i} + e_i$

$X_1 = 1$ for B, $X_1 = 0$ for A and C

$X_2 = 1$ for C, $X_2 = 0$ for A and B

Interpretation:

For system A: $Y_i = a + b_1*0 + b_2*0 = a$

For system B: $Y_i = a + b_1*1 + b_2*0 = a + b_1$

For system C: $Y_i = a + b_1*0 + b_2*1 = a + b_2$

b_1 is the difference between A and B, b_2 between A and C

It is all the same!

Multiple regression: $Y_i = a + b_1X_{1i} + b_2X_{2i} + e_i$

F-test in regression: test model against the mean

The mean is a model: $Y_i = a + e_i$

Therefore, the F-test has as H_0: $b_1 = 0$ and $b_2 = 0$

In other words, H_0 is that Ma = Mb = Mc

If the F-test is significant, there is a difference

(but doesn't say where!)

Assumptions

Normality within the groups

ANOVA is fairly robust against violations, as long as groups are equal in size

Otherwise, use robust methods

Homoscedasticity

Also robust with equal groups; Welch's method available

Independence

Very important

ANOVA in R

Dataset “Viagra.dat” -> set name to viagra

Effect viagra on libido

Variables:

person: participant ID

dose: Viagra treatment (1=Placebo, 2=Low Dose, 3=High Dose)

libido: level of libido after treatment (between 1 and 7)

Plotting

Let's start by making dose a factor with nice labels:

```
install "plyr" for the revalue function
viagra$dose <- revalue(as.factor(viagra$dose),
c("1"="Placebo","2"="Low Dose","3"="High Dose"))
```

Line plot with bootstrapped CIs:

```
ggplot(viagra,aes(dose,libido)) +
stat_summary(fun.y=mean, geom="line") +
stat_summary(fun.data=mean_cl_boot, geom="errorbar",
width = 0.2)
```

Assumptions

Normality tests per group:

by(viagra$libido, viagra$dose, stat.desc, desc=F, norm=T)

Looks normal

Levene's test:

install "car"

leveneTest(viagra$libido, viagra$dose, center=median)

Variance not significantly different between groups

Run the ANOVA

Run the ANOVA:

viagraAOV <- aov(libido ~ dose, data = viagra)

summary(viagraAOV)

```
            Df Sum Sq Mean Sq F value Pr(>F)
dose         2  20.13  10.067   5.119 0.0247 *
Residuals   12  23.60   1.967
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

plot(viagraAOV) <— to test the assumptions

It is all the same!

Run a regression:

```
viagraLM <- lm(libido ~ dose, data = viagra)
summary(viagraLM)
```

Now try:

```
summary.lm(viagraAOV)
summary.aov(viagraLM)
```

Welch's F version

If variances are not equal across groups, use Welch's F

```
oneway.test(libido ~ dose, data = viagra)
```

Note that this test is not significant!

Robust versions

With WRS2, they are easier than Field's versions

Trimmed version:

```
t1way(libido~dose, data = viagra, tr = 0.1)
```

Based on the bootstrapped median:

```
med1way(libido~dose, data = viagra, iter = 2000)
```

Based on the bootstrapped trimmed mean:

```
t1waybt(libido~dose, data = viagra, tr = 0.1, nboot = 2000)
```

Contrasts

How to test specific differences

Contrasts

If F-test is significant we know that the means differ

- Which means exactly? Ma and Mb? Ma and Mc? Mb and Mc? All of them?

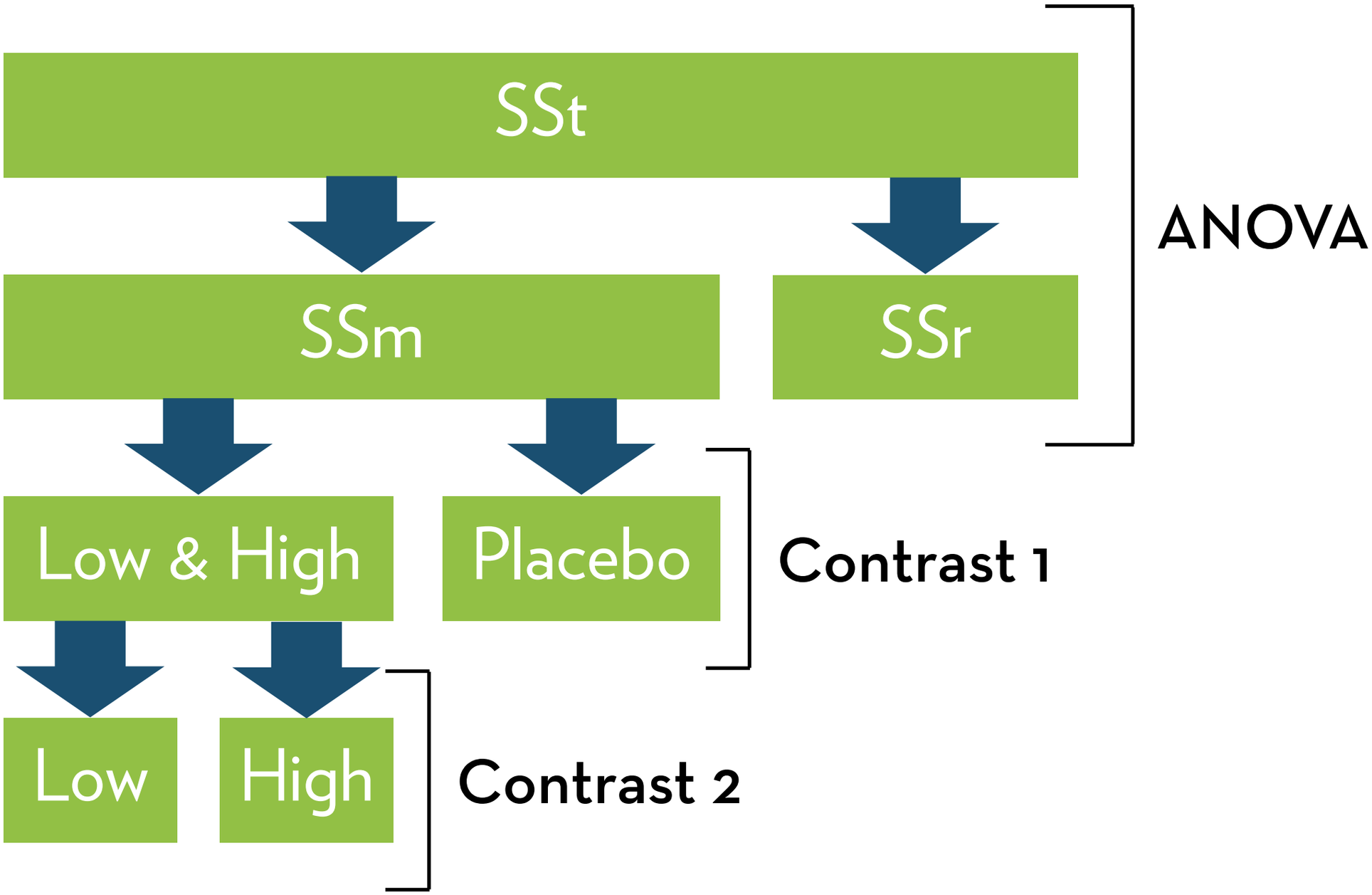
If you have specific hypotheses, do tests on planned contrasts

- Otherwise, do post-hoc tests

F-test divides total variation (SSt) into model variation (SSm) and residual variation (SSr)

- Planned contrasts further divide SSm into components

Contrasts


SSt
SSm
SSr
ANOVA
Low & High
Placebo
Contrast 1
Low
High
Contrast 2

Interpretation

Both contrasts are significant:

> The High dose significantly increases libido over other groups, can't say anything about the Low dose group

Contrast 1 is significant and contrast 2 is not:

> Viagra increases libido, but the dose likely doesn't matter

Contrast 2 is significant and contrast 1 is not:

> The High dose significantly increases libido over other groups, the Low dose does not

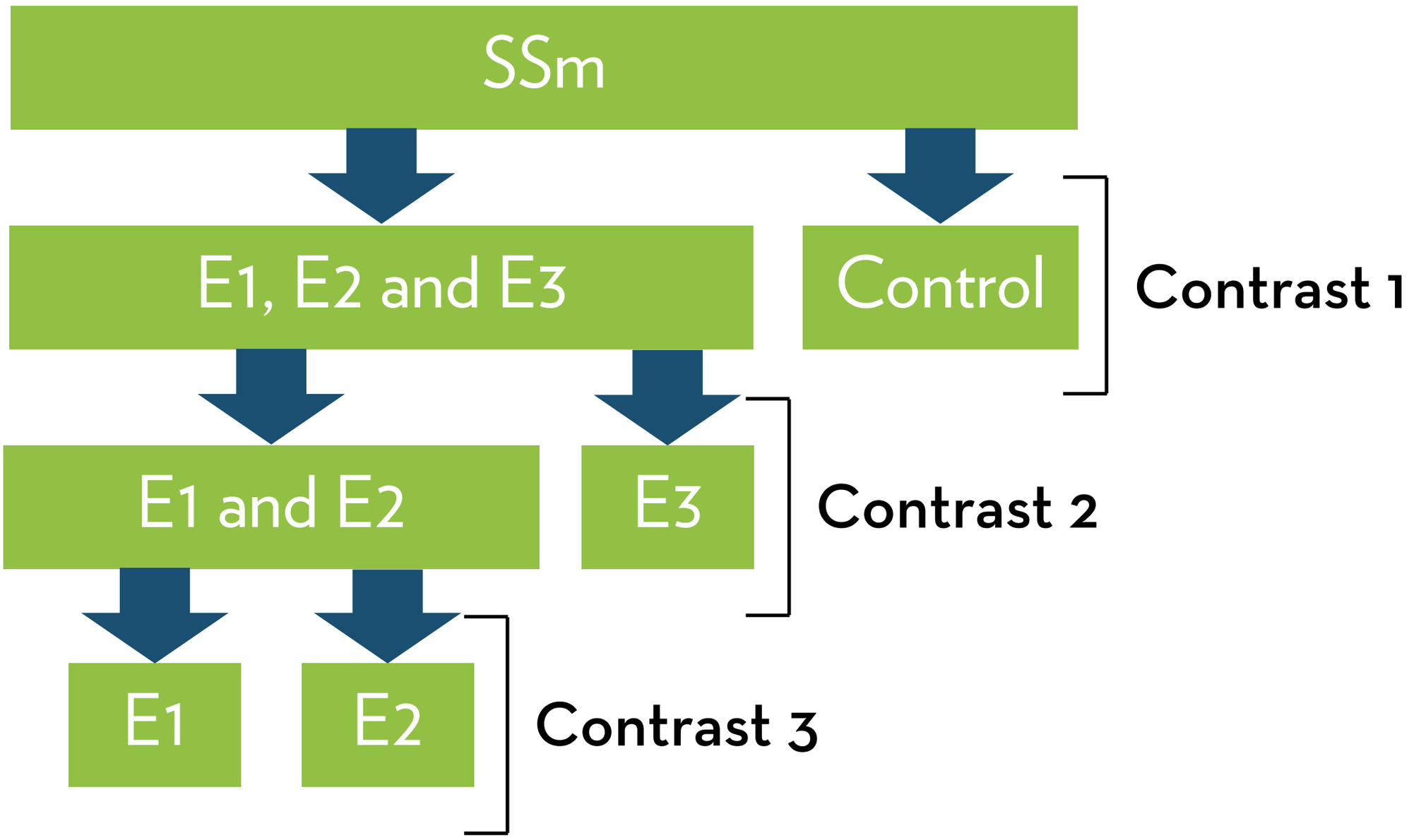
Contrasts

How to design good contrasts:

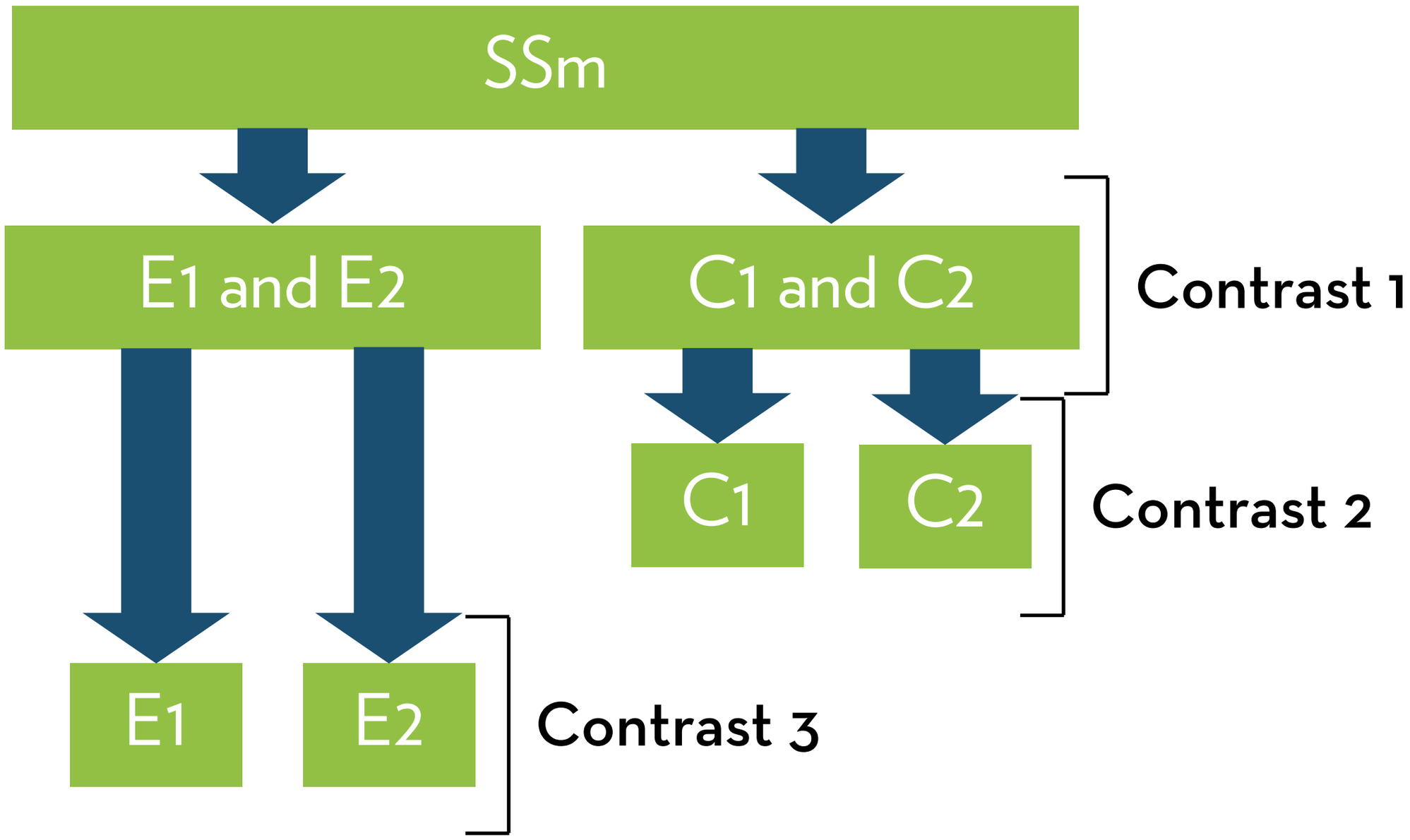
- Split any chunk of variance (multiple conditions) into two chunks at most
- If a condition has been singled out, you can't reuse it
- Only split, don't merge
- If you have a control group, your first contrast usually compares everything else against the control group (or groups)

You will always end up with k–1 contrasts

More examples



More examples



Contrast dummies

To test a contrast, you create dummies, which assign weights to each condition

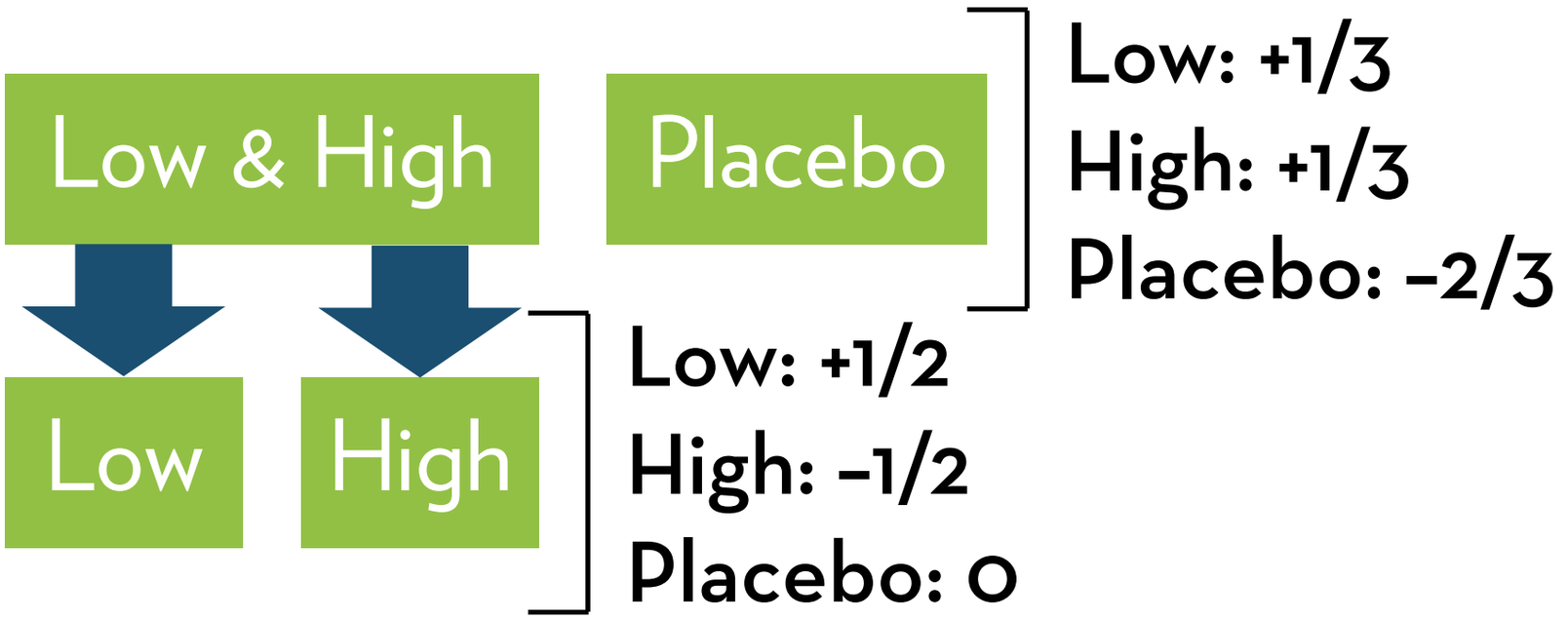
Say that you are testing chunk P versus chunk Q:

- conditions that belong to neither chunk get a weight of 0
- conditions in chunk P get a weight of $k_q/(k_p+k_q)$

  k_p, k_q = number of conditions in chunk P, Q
- conditions in chunk Q get a weight of $-k_p/(k_p+k_q)$

Check: weights sum to zero, product also sums to zero

Example



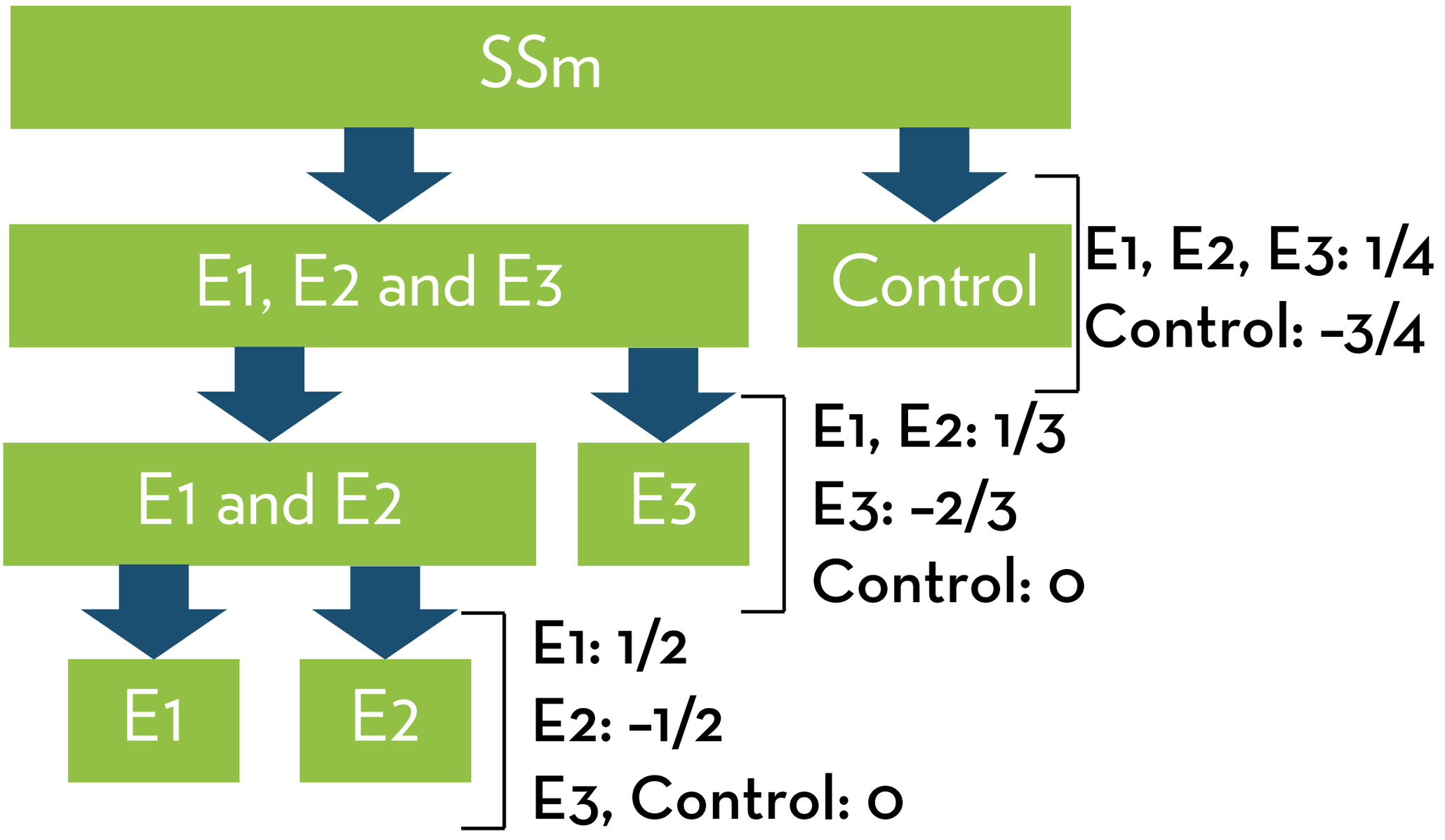
Group	Dummy 1	Dummy 2	Product
Placebo	–2/3	0	0
Low dose	1/3	–1/2	–1/6
High dose	1/3	1/2	1/6
Sum	0	0	0

Interpretation:

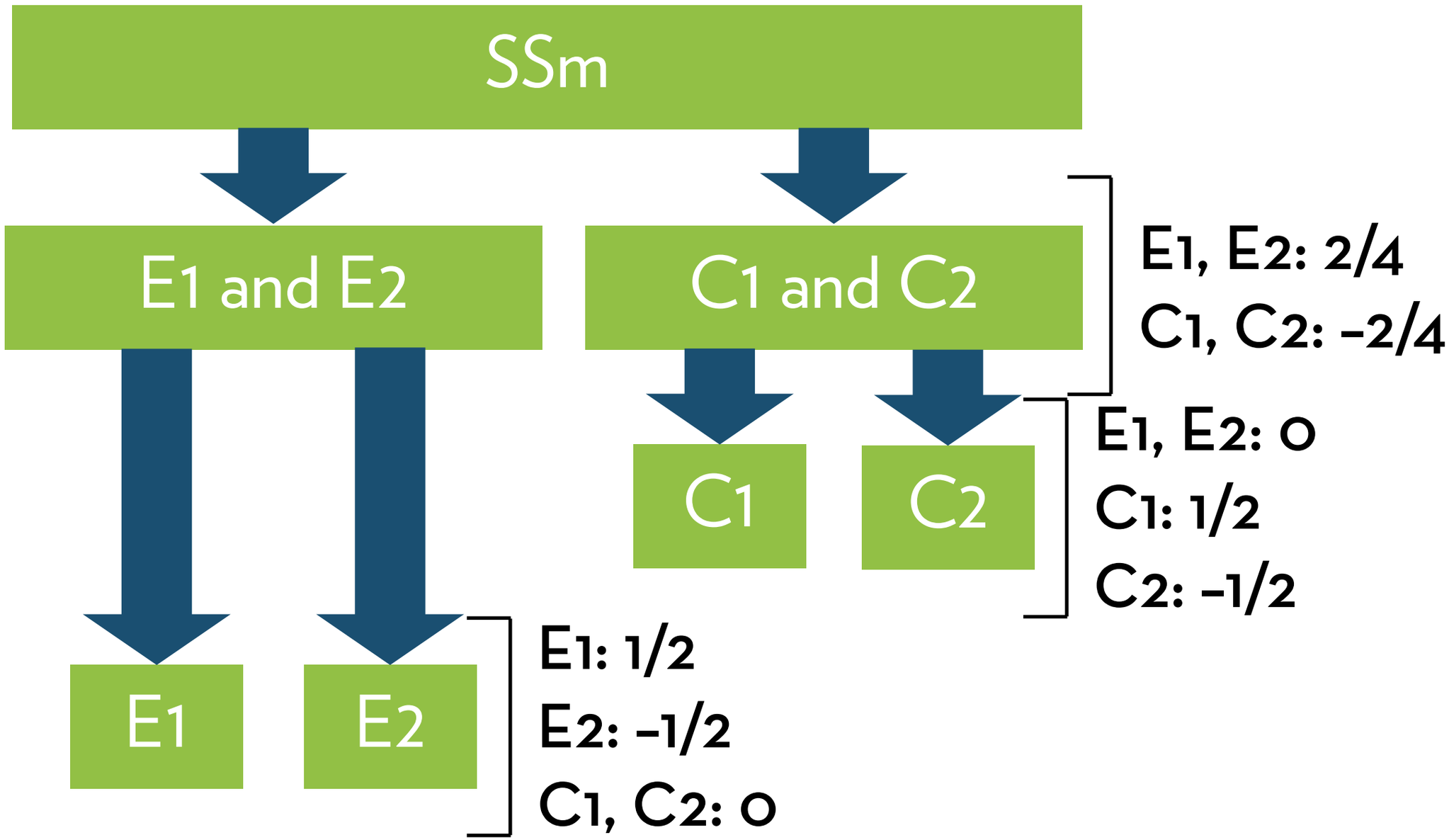
Dummy 1: difference between placebo and average of Low and High

Dummy 2: difference between Low and High

More examples



More examples



Orthogonal

Make sure you do not reuse or merge!

This way, you test contrasts that are **orthogonal**: your tests never include the same chunk twice

This means that your p-values are not affected by family-wise error!

In other words: alpha = .05, even though you test multiple contrasts

Orthogonal

One famous orthogonal contrast is the Helmert-contrast:

contr.helmert(k):

Contrast	Chunk P	Chunk Q
Contrast 1	2	1
Contrast 2	3	1, 2
…	…	…
Contrast k–1	k	1, 2, 3, … k–1

Using this function, you don't have to code the dummies yourself

Non-orthogonal

If you do reuse chunks, your contrasts are non-orthogonal

If so, you should use a smaller alpha (see post-hoc tests)

Some standard non-orthogonal contrasts:

- dummy (this is the default contrast)
- contr.SAS(k)
- contr.treatment(k, base = x)

k–1 contrasts that compare every other condition against the first (dummy), last (SAS), or x'th (treatment) condition

Polynomial

Find a trend in the data: contr.poly(k)

Useful if your X groups are ordered

Types: linear, quadratic, cubic, ... (usually only the first two are used)

Contrasts in R

You already know how to get the default dummy coding:

summary.lm(viagraAOV)

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)     2.2000     0.6272   3.508  0.00432 **
doseLow Dose    1.0000     0.8869   1.127  0.28158
doseHigh Dose   2.8000     0.8869   3.157  0.00827 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Compares each condition against Placebo

Helmert

Let's try Helmert:

```
contrasts(viagra$dose) <- contr.helmert(3)
viagra$dose, or: levels(viagra$dose)
viagraHelmert <- aov(libido ~ dose, data = viagra)
summary.lm(viagraHelmert)
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.4667     0.3621   9.574 5.72e-07 ***
dose1         0.5000     0.4435   1.127   0.2816
dose2         0.7667     0.2560   2.994   0.0112 *
```

dose1: Low vs. Placebo; dose 2: High vs. (Low, Placebo)

Polynomial

Let's try Polynomial:

```
contrasts(viagra$dose) <- contr.poly(3)
viagra$dose
viagraPoly <- aov(libido ~ dose, data = viagra)
summary.lm(viagraPoly)
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.4667     0.3621   9.574 5.72e-07 ***
dose.L        1.9799     0.6272   3.157  0.00827 **
dose.Q        0.3266     0.6272   0.521  0.61201
```

L is linear contrast, Q is quadratic

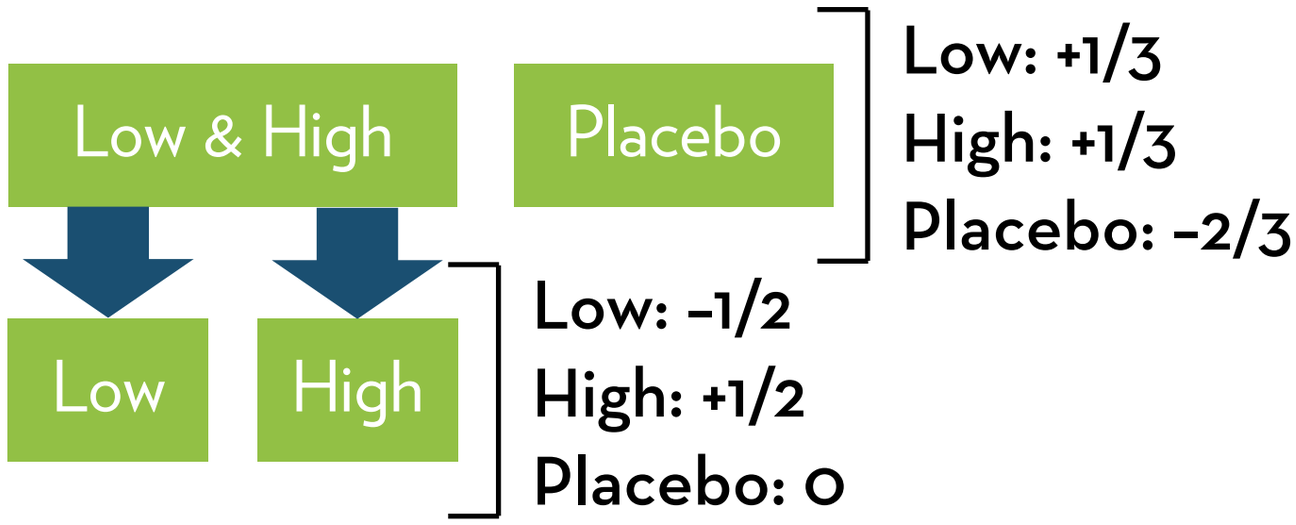
Make our own

Placebo against other two:

```
PvLH <- c(-2/3, 1/3, 1/3)
```

Low vs. High dose:

```
LvH <- c(0, -1/2, 1/2)
```



Load the contrasts:

```
contrasts(viagra$dose) <- cbind(PvLH, LvH)
viagra$dose
```

Make our own

Run the model:

```
viagraPlanned <- aov(libido ~ dose, data = viagra)
summary.lm(viagraPlanned)
```

```
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    3.4667     0.3621   9.574 5.72e-07 ***
dosePvLH       1.9000     0.7681   2.474   0.0293 *
doseLvH        1.8000     0.8869   2.029   0.0652 .
```

Viagra has a significant effect on libido ($p_{\text{one-tailed}} = .015$)

A high dose is significantly more effective than a low dose ($p_{\text{one-tailed}} = .033$)

Reporting

Planned contrasts:

Planned contrasts revealed that taking any dose significant increased libido, compared to the placebo, $t(12) = 2.47$, $p_{\text{one-tailed}} = .015$, and that a high dose is significantly more effective than a low dose $t(12) = 2.03$, $p_{\text{one-tailed}} = .033$.

Robust contrasts?

Run robust t-tests on the contrast dummies!

Create dummies:

```
viagra$d1 <- -2/3*(viagra$dose==“Placebo”)
+1/3*(viagra$dose!=”Placebo”)

viagra$d2 <- -1/2*(viagra$dose==”Low Dose”)
+1/2*(viagra$dose==”High Dose”)
```

Run yuen, yuenbt, or pb2gen:

```
pb2gen(libido~d1, data = viagra, nboot = 2000)
```

Post-hoc tests

...make ALL the comparisons!

Post-hoc tests

What if we have no idea which chunks to compare?

Just compare all conditions against each other!

But what about family-wise error?

Post-hoc methods: reduce alpha to account for family-wise error

Simplest method: Bonferroni

divide alpha by the number of tests (p_{crit} = alpha/k)

Holm

Bonferroni is very conservative!

Alternative: Holm

Order your p-values by size, smallest first

The p_{crit} for the first one is alpha/k

If $p < p_{crit}$, move to the next (otherwise stop)

The p_{crit} for the next one is alpha/(k–1)

If $p < p_{crit}$, move to the next (otherwise stop)

Etc.

Holm

Example:

p-value	p_{crit}	Verdict?
0.0000	.05/6 = .0083	significant
0.0014	.05/5 = .0100	significant
0.0120	.05/4 = .0125	significant
0.0252	.05/3	not significant, stop!
0.1704		(also not significant)
0.3431		(also not significant)

Benjamini-Hochberg

How about controlling false discovery rate rather than type I error rate?

Benjamini-Hochberg:

Order your p-values by size, **largest** first

The p_{crit} for the first one is (k/k)*alpha

If p **>** p_{crit}, move to the next (otherwise stop)

The p_{crit} for the next one is ((k–1)/k)*alpha

If p > p_{crit}, move to the next (otherwise stop)

Etc.

Benjamini-Hochberg

Example:

p-value	p_{crit}	Verdict?
0.3431	0.05	not significant
0.1704	.05*(5/6) = .0417	not significant
0.0252	.05*(4/6) = .0333	significant, stop!
0.0127		(also significant)
0.0014		(also significant)
0.0000		(also significant)

Others

With equal variances and group sizes, Tukey is quite good

When variances are unequal, or groups differ in size:

- Hochberg GT2, unfortunately not implemented in R!

Alternative: Robust methods!

- Trimming, bootstrapping, M-estimators
- Also good for non-normality
- Note: bootstrapping is more conservative, M-estimator is more powerful

Post-hoc tests in R

Note: R will adjust the p-value up (rather than adjusting alpha down)

Bonferroni:

```
pairwise.t.test(viagra$libido, viagra$dose, p.adjust.method = "bonferroni")
```

Benjamini-Hochberg:

```
pairwise.t.test(viagra$libido, viagra$dose, p.adjust.method = "BH")
```

Post-hoc tests in R

Tukey:

```
install the package "multcomp"
post <- glht(viagraAOV, linfct=mcp(dose = "Tukey"))
summary(post)
confint(post)
```

Dunnett (tests against a baseline):

```
post <- glht(viagraAOV, linfct=mcp(dose = "Dunnett"))
```

Robust post-hoc

Trimming:

lincon(libido~dose, data=viagra, tr=0.1)

CIs are corrected, p-values are not

Trimming + bootstrapping:

mcppb20(libido~dose, data=viagra, tr=0.2, nboot=2000)

M-estimators:

involves linconm, but not enough data here!

Effect sizes

R^2 = SSm/SSt (in ANOVA, we call it eta-squared)

In R, run summary.lm —> .460

Take the square root for r: .679, this is a large effect

Better: get omega-squared:

(SSm – dfm*MSr)/(SSt+MSr)

You can get all of these from the aov summary

In the viagra case: omega-squared = .35; omega = .60

Effect sizes

Effect sizes for specific differences: use mes() in the "compute.es" package

Get means, sds, and ns from stat.desc:

```
desc <- by(viagra$libido,viagra$dose,stat.desc)
```

Plug values into mes, e.g.:

```
mes(desc$Placebo["mean"], desc$`Low Dose`["mean"],
desc$Placebo["std.dev"], desc$`Low Dose`["std.dev"],5,5)
```

"5,5" is the N in each group (change for your data!)

Common to report *d* (sd difference) instead of *r*

Effect sizes

Effect sizes for specific contrasts: use $\sqrt{(t^2/(t^2+df))}$

Get values from contrast:

```
summary.lm(viagraPlanned)
```

Plug into formula:

```
sqrt(2.474^2/(2.474^2+12)) = .581
sqrt(2.029^2/(2.029^2+12)) = .505
```

AB Reporting

Omnibus test (always present this first!):

> There was a significant effect of Viagra on levels of libido, $F(2, 12) = 5.12$, $p = .025$, $\omega = .60$.

Then follow up with one of the following...

Linear contrast:

> There was a significant linear trend, $t(12) = 4.157$, $p = .008$, indicating that libido increases proportionally with dose.

Reporting

Planned contrasts:

Planned contrasts revealed that taking any dose significant increased libido, compared to the placebo, $t(12) = 2.47$, $p_{\text{one-tailed}} = .015$, and that a high dose is significantly more effective than a low dose $t(12) = 2.03$, $p_{\text{one-tailed}} = .033$.

Reporting

Post hoc tests:

> Despite large effect sizes, Bonferroni tests revealed non-significant differences between low dose and placebo, p = .845, d = –0.77 and between low dose and high dose, p = .196, d = –1.24. However, the high dose group had a significantly higher libido than the placebo group, p = .025; a difference of almost 2 standard deviations, d = –1.93.

“It is the mark of a truly intelligent person
to be moved by statistics.”

T H A N K S !

George Bernard Shaw