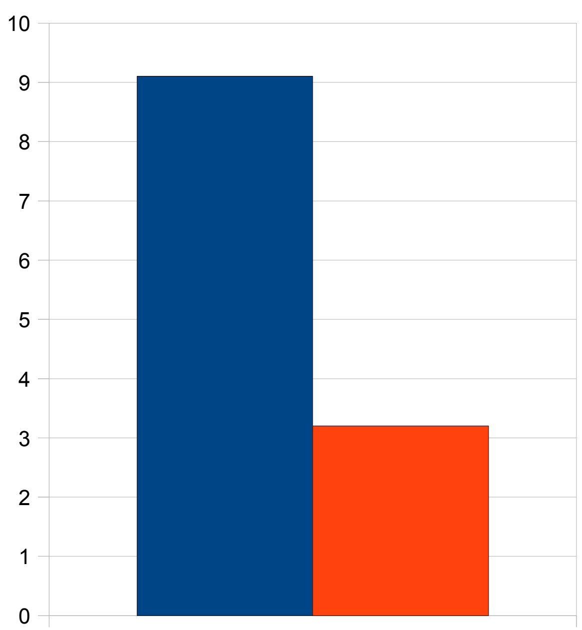
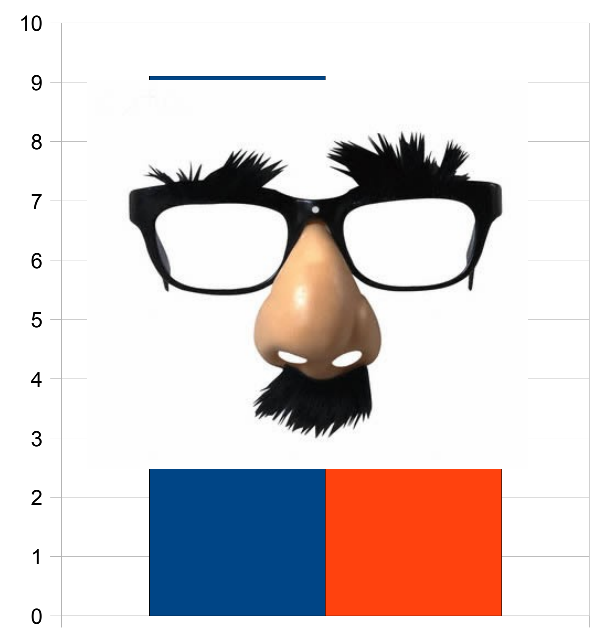
Main Effects vs. Simple Effects

Scott Fraundorf
MLM Reading Group
April 7th, 2011



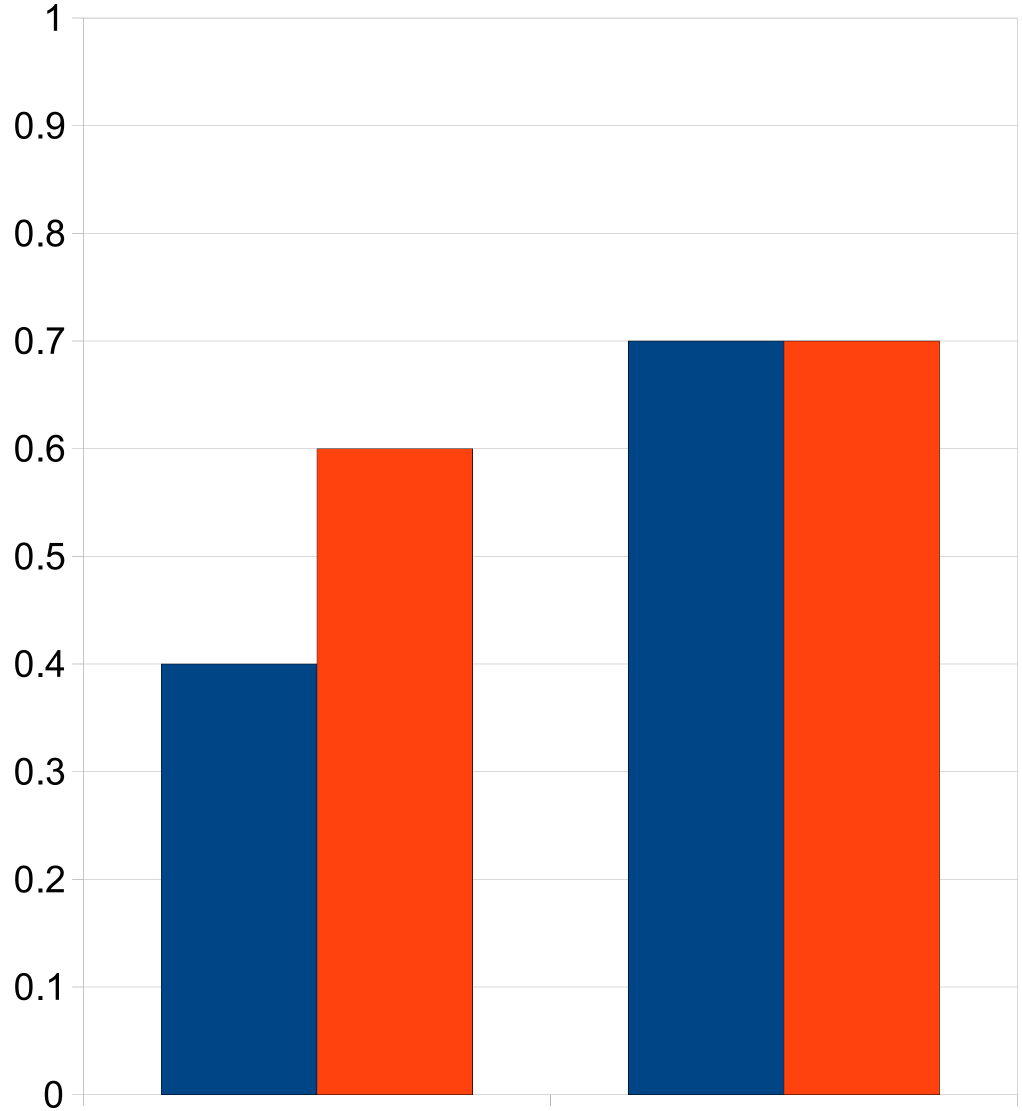
If you want to talk about main effects, need to distinguish true “main effects”...

...from their insidious cousin that may be masquerading as “main effect” in your model



Outline

- The Problem
- Recap of Coding
- Parameter Testing
- Simple Effects & Main Effects
- More Detailed Explanation
- How to Do Contrast Coding
- Continuous Predictors

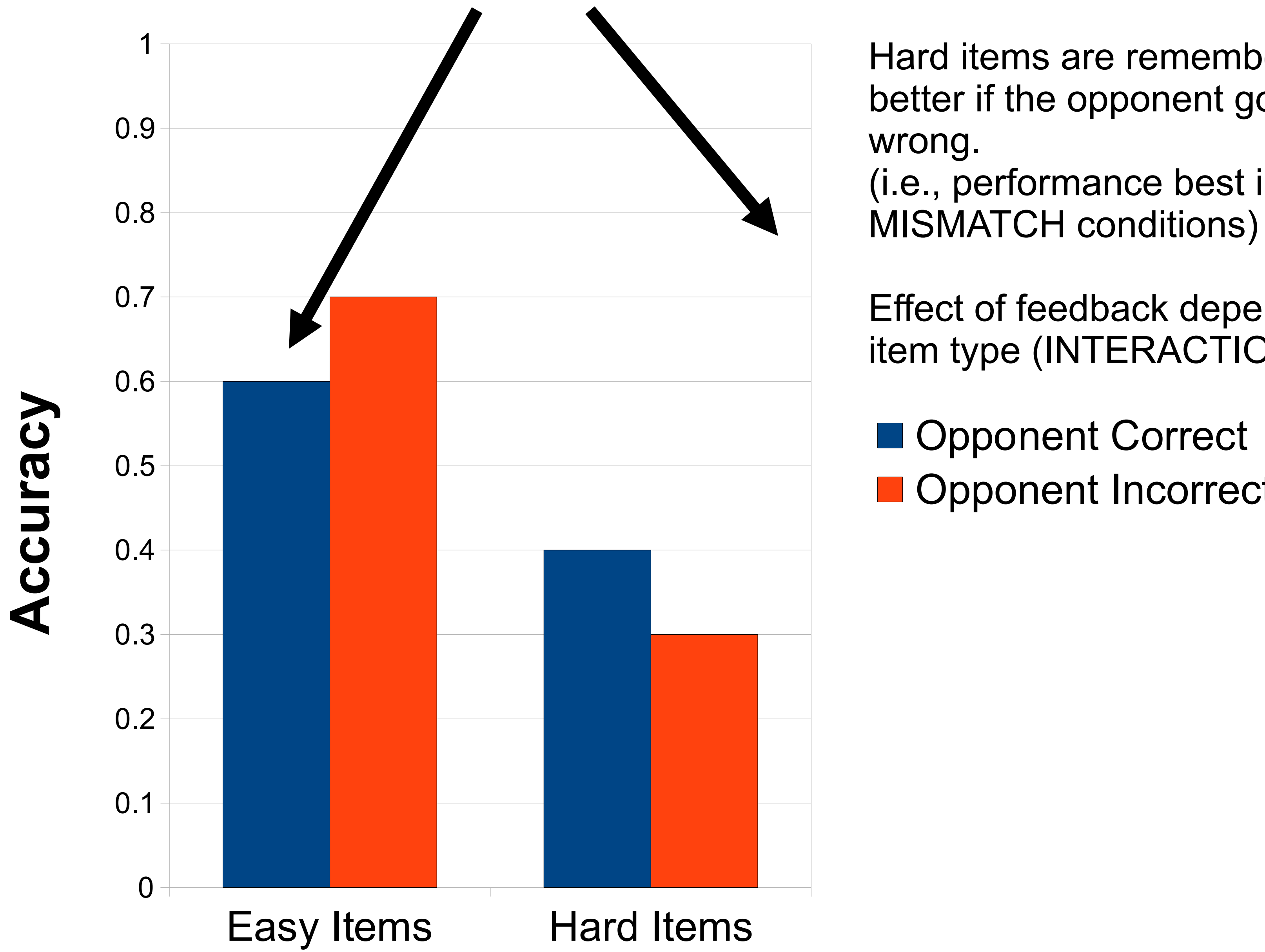


Prototypical psychology study: 2 x 2 design

Example Study

- Study **easy** and **difficult** word pairs
  - VIKING—HELMET (*related* and thus *easy*)
  - VIKING—COLLEGE (*unrelated* and thus *hard*)
- Do cued recall task:
  - VIKING---?????
- During test phase, told if an opponent supposedly got the item **correct** or **incorrect**

INTERACTION!

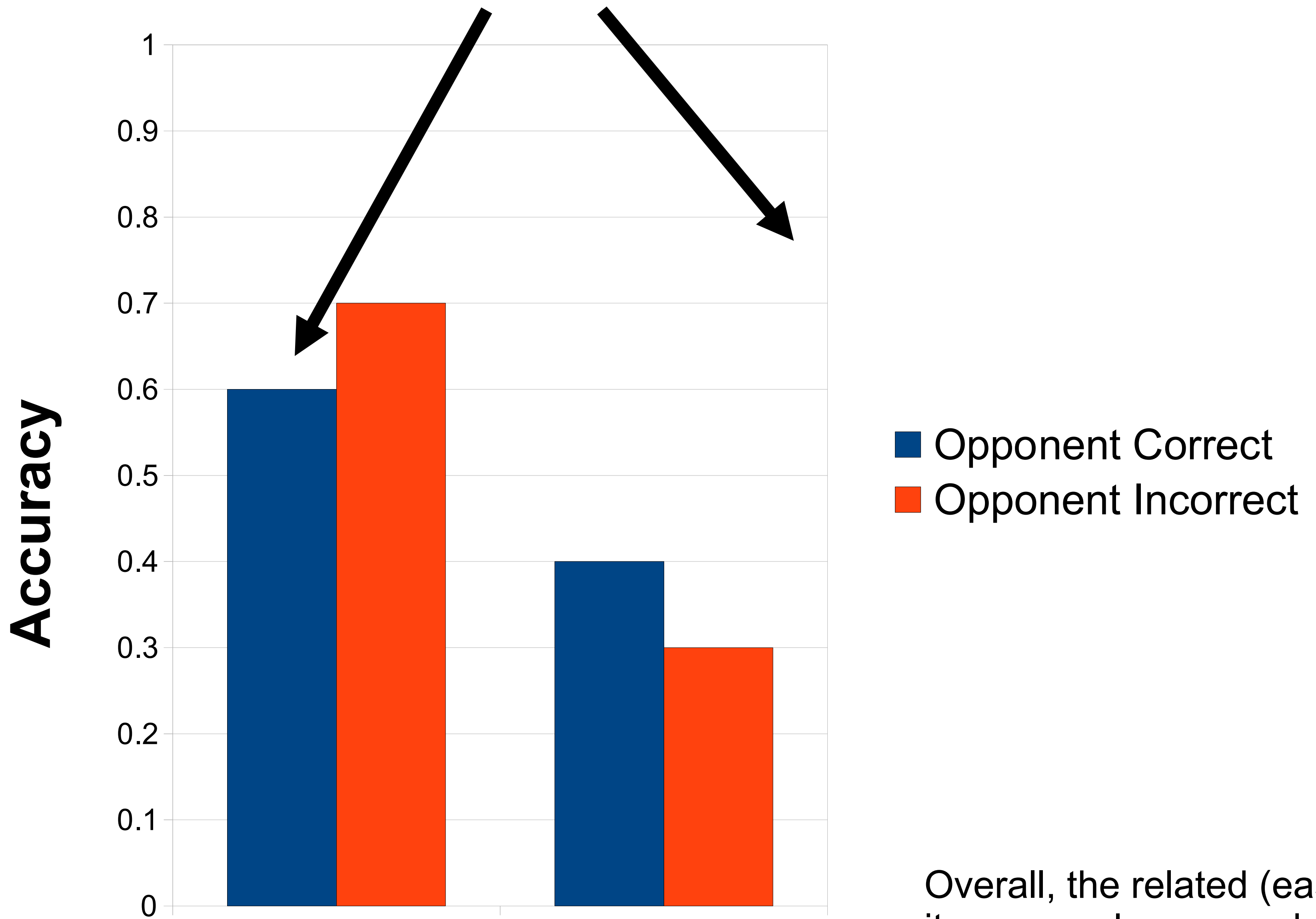


Easy items are remembered better if the opponent supposedly got them right.

Hard items are remembered better if the opponent got them wrong.
(i.e., performance best in the MISMATCH conditions)

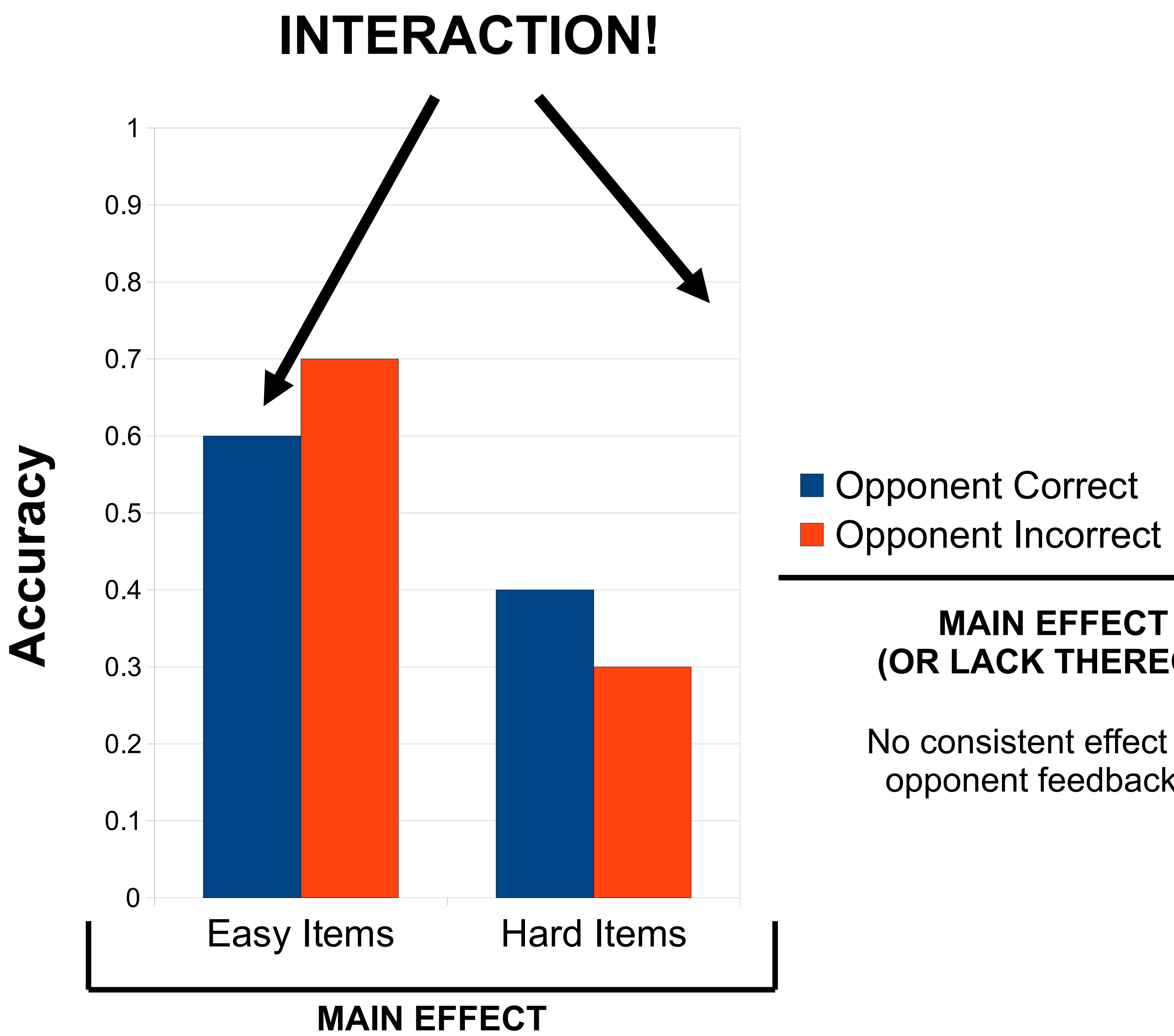
Effect of feedback depends on item type (INTERACTION).

INTERACTION!



MAIN EFFECT

Overall, the related (easy) items are also remembered better than the unrelated (hard) items



MAIN EFFECT
(OR LACK THEREOF)

No consistent effect of opponent feedback.

The Problem

ANOVA WORLD

- Get test of interaction
- And of 2 main effects

MLM WORLD

- Not in Kansas anymore!
- What to do?

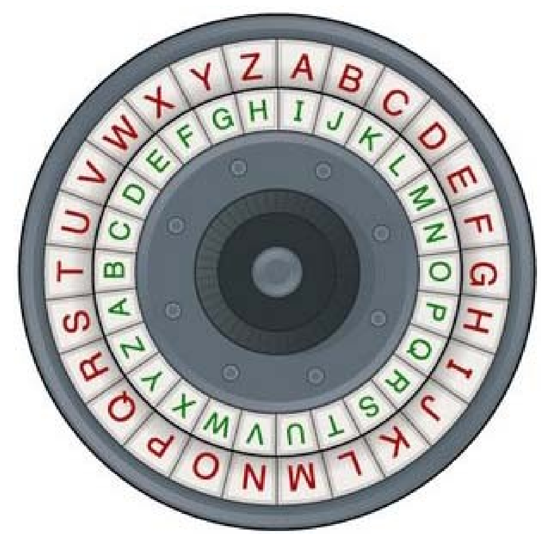
The Problem

- Modeling our outcome variable in a regression equation

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_{12} X_1 X_2 + \ldots$$

- Need to *code* categorical variables into numerical ones

- Consequences for *how you interpret* hypothesis tests



R's secret decoder wheel

Outline

- ~~The Problem~~
- Recap of Coding
- Parameter Testing
- Simple Effects & Main Effects
- More Detailed Explanation
- How to Do Contrast Coding
- Continuous Predictors

ITEM TYPE

Related : 0
Unrelated : 1

One level is 1

Other level is 0

OPPONENT FEEDBACK

Correct : 0
Incorrect : 1

DUMMY CODING a/k/a TREATMENT CODING (R's default)

Predictor with >2 levels: get more dummy-coded variables

ITEM TYPE	
Related	: 0
Unrelated	: 1

OPPONENT (A)	
Didn't See	: 0
Correct	: 1
Incorrect	: 0

OPPONENT (B)	
Didn't See	: 0
Correct	: 0
Incorrect	: 1

DUMMY CODING a/k/a TREATMENT CODING
(R's default)

ITEM TYPE

Related	: 0
Unrelated	: 1

One level is 1

Other level is 0

OPPONENT FEEDBACK

Correct	: 0
Incorrect	: 1

DUMMY CODING

CONTRAST CODING

ITEM TYPE

Related	: -0.5
Unrelated	: 0.5

One level is positive

Other level is negative

OPPONENT FEEDBACK

Correct	: -0.5
Incorrect	: 0.5

Outline

- ~~The Problem~~
- ~~Recap of Coding~~
- Parameter Testing
- Simple Effects & Main Effects
- More Detailed Explanation
- How to Do Contrast Coding
- Continuous Predictors

Testing a Parameter

How to tell if the opponent's feedback is related to memory?
(e.g. possible main effect: you just try harder when someone else got the item wrong)

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots$$

Feedback
0 = Correct
1 = Incorrect

Item Type
0 = Related
1 = Unrelated

Testing a Parameter

Compare when feedback = 0...

$$Y = \beta_0 + \beta_1 0 + \beta_2 X_2 + \; ...$$

Feedback
- 0 = Correct
- 1 = Incorrect

Item Type
- 0 = Related
- 1 = Unrelated

Testing a Parameter

… to when feedback = 1

$$Y = \beta_0 + \beta_1 1 + \beta_2 X_2 + \ldots$$

Feedback
0 = Correct
→ 1 = Incorrect

Item Type
0 = Related
1 = Unrelated

Testing a Parameter

β_1: "The effect of changing feedback, while holding item type constant"

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots$$

But, we know there's an interaction … so it will matter *what value* we hold item type constant at!

Feedback
0 = Correct
1 = Incorrect

Item Type
0 = Related
1 = Unrelated

Testing a Parameter

$$Y = \beta_0 + \beta_1 X_1 +$$

Feedback
0 = Correct
1 = Incorrect

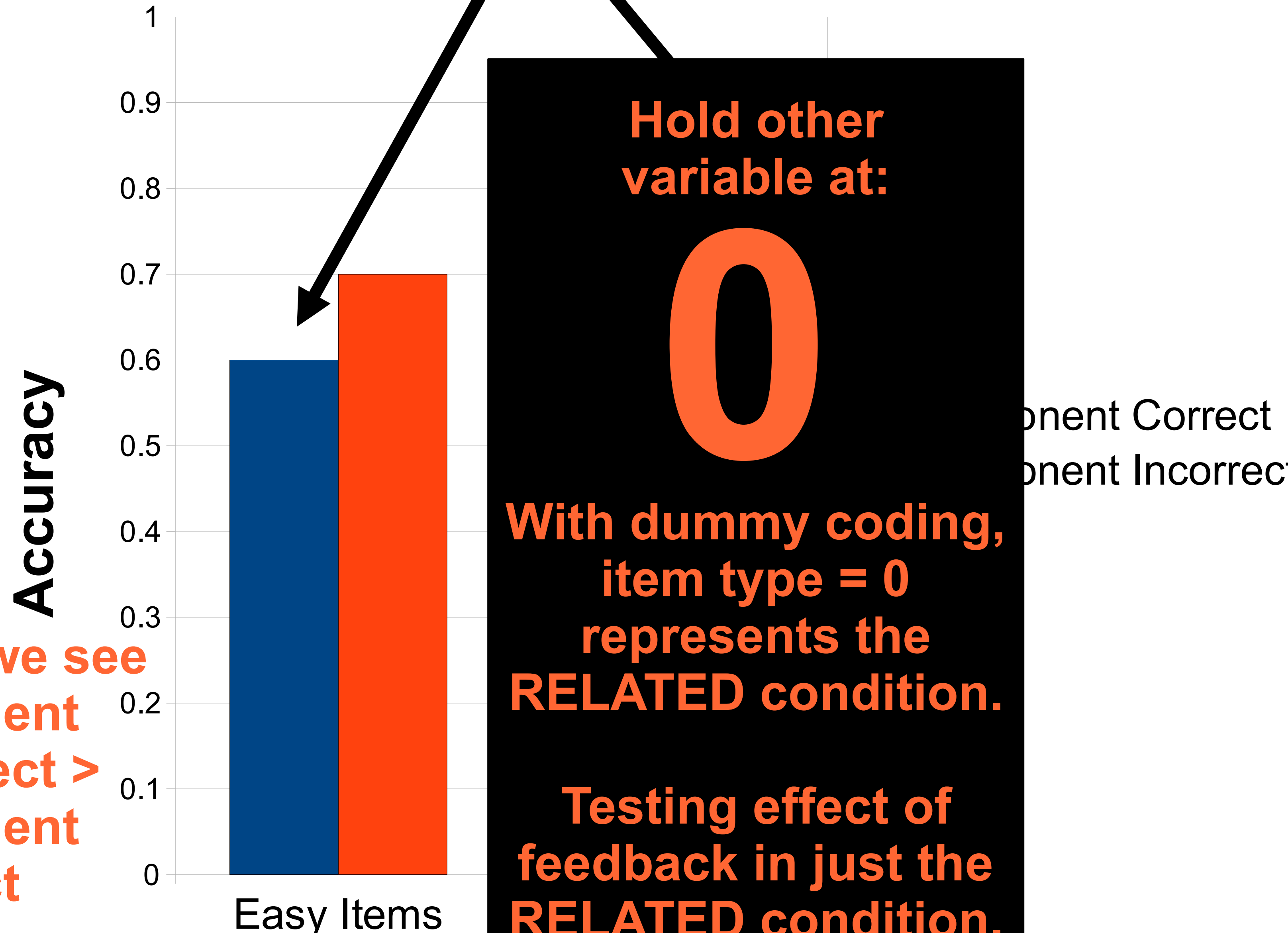
Hold other variable at:

0

With dummy coding, item type = 0 represents the RELATED condition.

Testing effect of feedback in just the RELATED condition.

INTERACTION!



Hold other variable at:

0

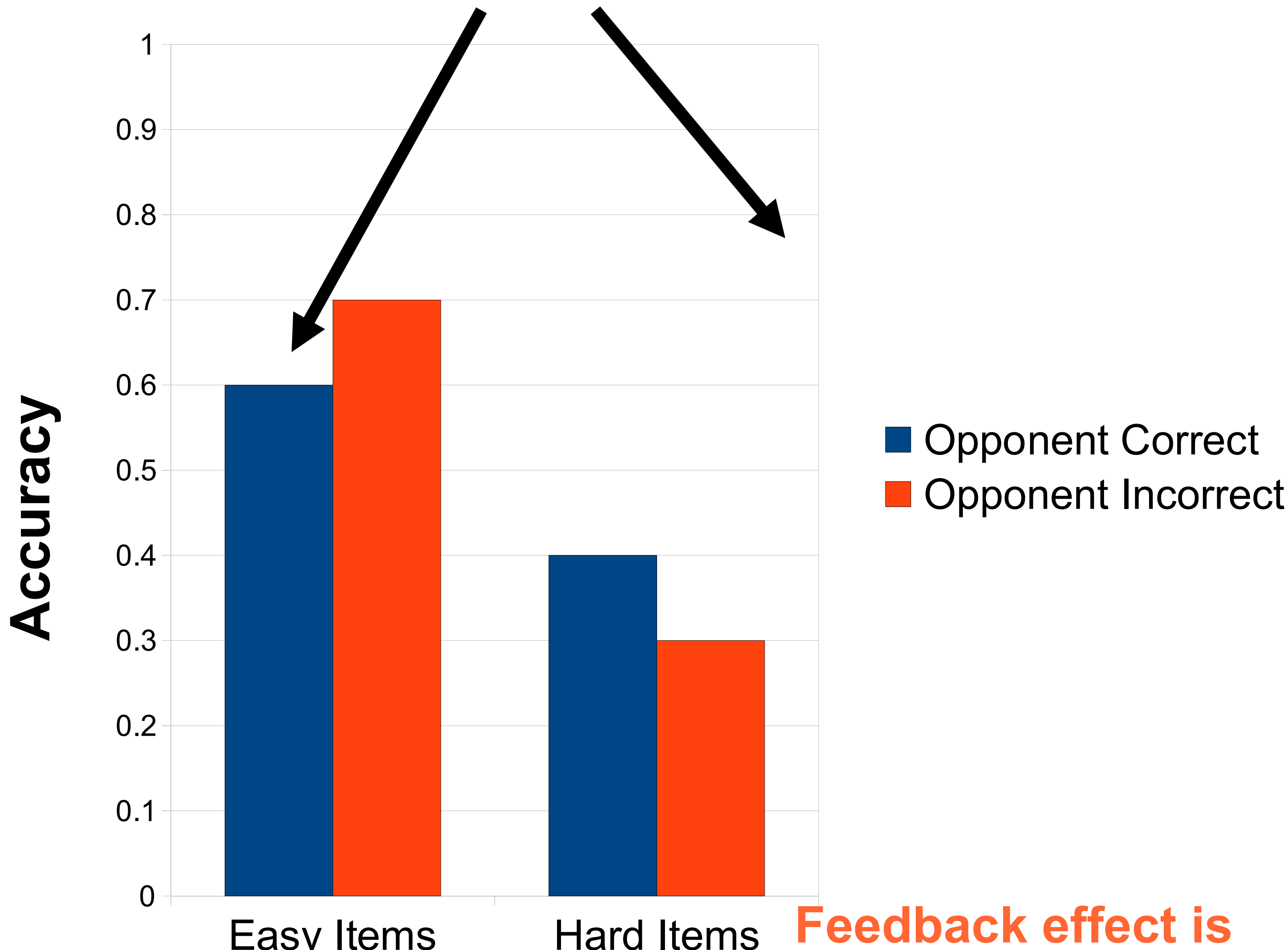
With dummy coding, item type = 0 represents the RELATED condition.

Testing effect of feedback in just the RELATED condition.

Here, we see Opponent Incorrect > Opponent Correct

But, this test only reflects HALF of the graph!

INTERACTION!



But, this test only reflects HALF of the graph!

Feedback effect is different in the other half. Misleading!

Outline

- ~~The Problem~~
- ~~Recap of Coding~~
- ~~Paramoter Testing~~
- Simple Effects & Main Effects
- More Detailed Explanation
- How to Do Contrast Coding
- Continuous Predictors

FEEDBACK

Correct : 0
Incorrect : 1

ITEM TYPE

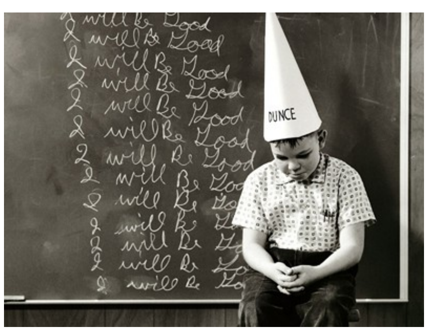
Related : 0
Unrelated : 1

DUMMY CODING

What is effect of the Feedback variable?

R holds **Item Type** at **0.**

Only reflects Related condition.



Simple Effect

Problem: Effect of Feedback depends on the Item Type. (i.e., there's an INTERACTION)

Main Effect

What is effect of the Feedback variable?

R holds **Item Type** at **0**.

Averaged between 2 conditions.

Test now uses information from *both* Item Types in testing Feedback. (No main effect here.)

CONTRAST CODING

FEEDBACK

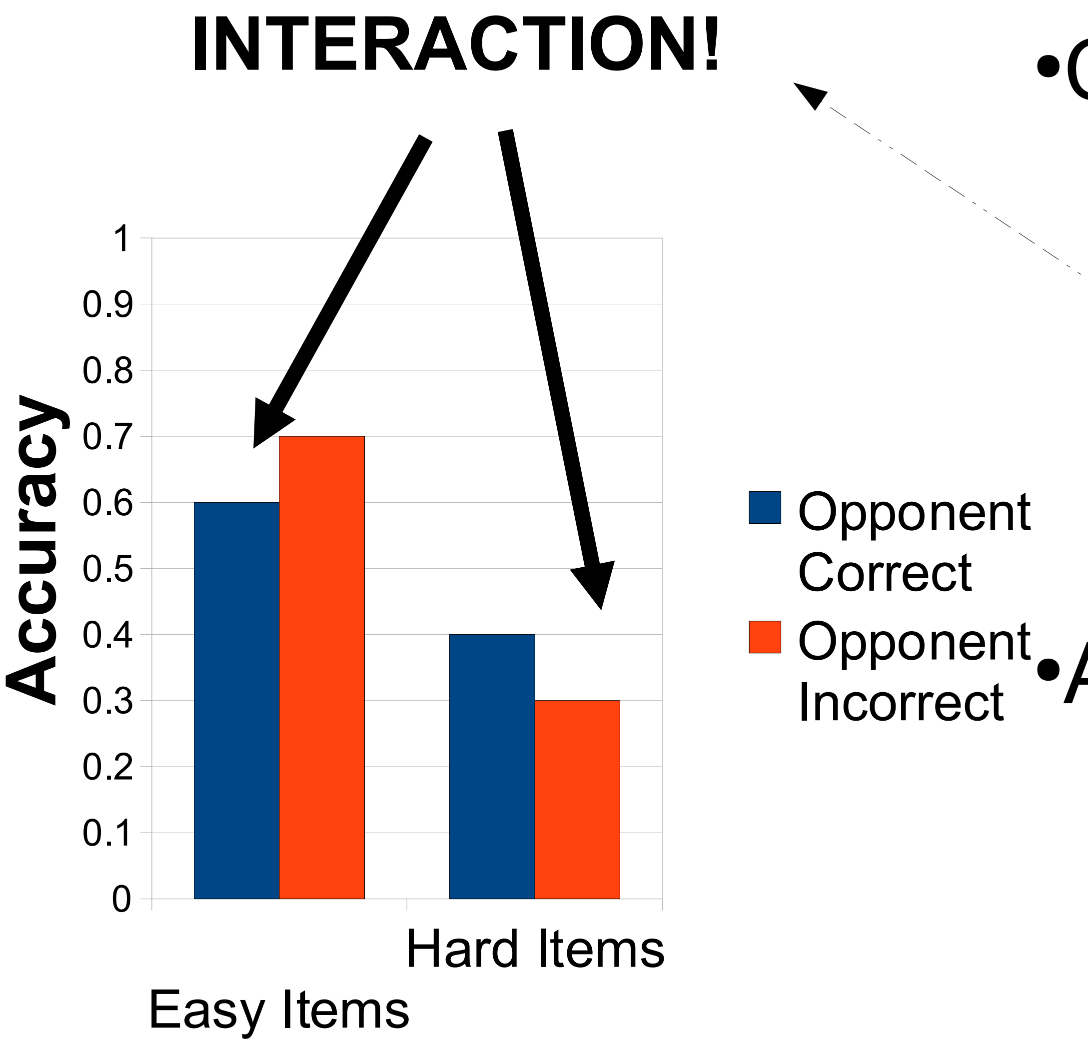
Correct : -0.5
Incorrect : 0.5

ITEM TYPE

Related : -0.5
Unrelated : 0.5

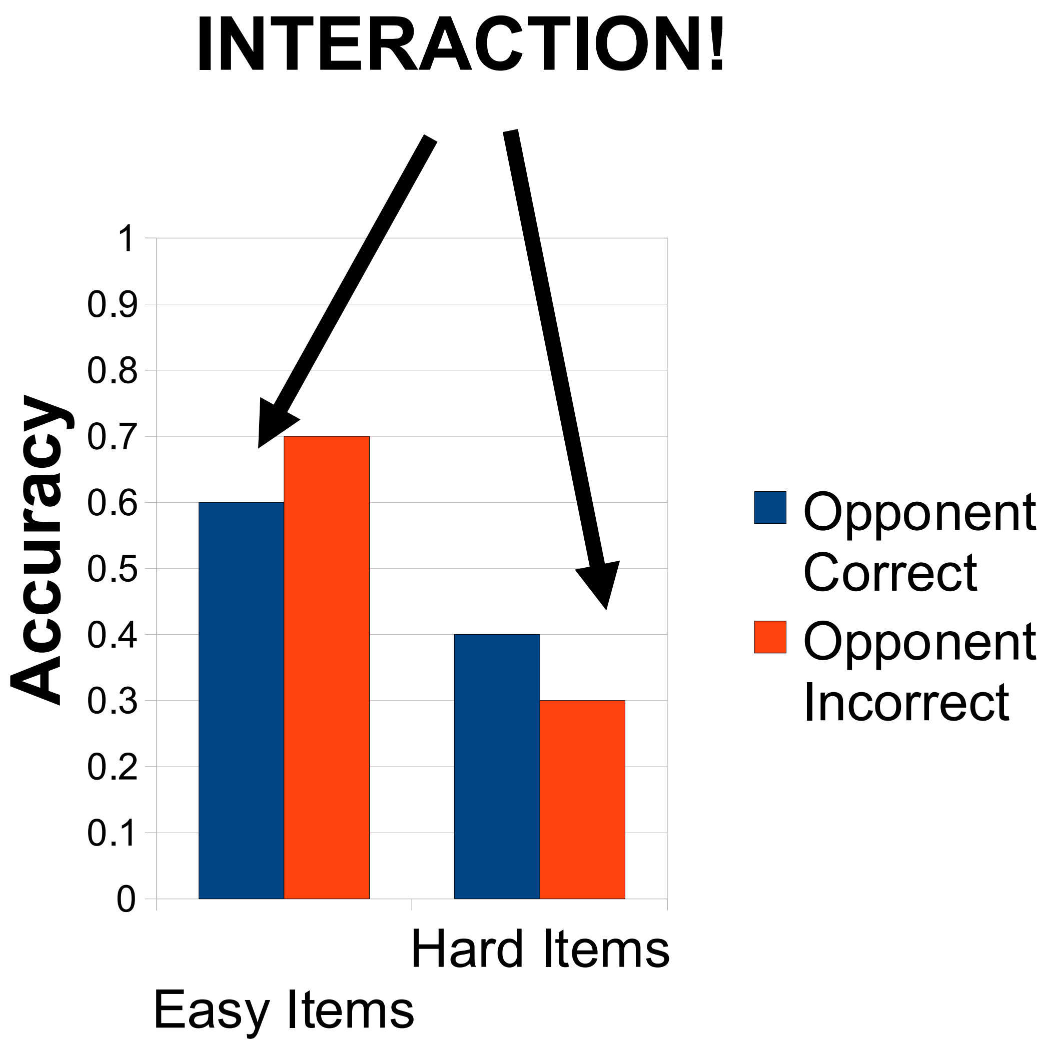
- **Dummy Coding -> Simple Effects**
  - Consider only *one level* of predictor X_2 in testing predictor X_1
- **Contrast Coding -> Main Effects**
  - Consider *all levels* of predictor X_2 in testing predictor X_1
- Both are legitimate statistical tests, but **they test different things**
  - **Simple effects** may be *appropriate* if you WANT to only test at one level of predictor X_2
  - e.g. that level is the baseline (*"opponent didn't see"* condition?)
  - Just make sure that your tests are testing what you say they are!

Some Other Notes...



- Coding differences **<u>do not</u>** affect the test of the **<u>interaction</u>**
  - Coding only changes the simple/main effect terms
- Also **<u>doesn't</u>** change **<u>overall fit</u>** of the model

Some Other Notes...



- If NO interaction, **simple effects** and **main effects** are the **<u>same</u>**
  - X_2 is irrelevant to X_1 effect
- But note that even if interaction isn't reliable at $\alpha = .05$, there can be a *numerical* interaction
  - Would still be *some* difference between simple effects & main effects

Outline

- ~~The Problem~~
- ~~Recap of Coding~~
- ~~Paramoter Testing~~
- ~~Simple Effects & Main Effects~~
- More Detailed Explanation
- How to Do Contrast Coding
- Continuous Predictors

Dummy Coding

- $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_{12} X_1 X_2$ (+ random effects, error)
- X_1 = 0 if related, 1 if unrelated
- X_2 = 0 if opponent right, 1 if opponent wrong
- Results:
  - Related, Right: $\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_{12} X_1 X_2$
  - $= \beta_0 + \beta_1(0) + \beta_2(0) + \beta_{12}(0)(0)$

    (substituting in 0s for X_1 and X_2)

Dummy Coding

- $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_{12} X_1 X_2$ (+ random effects, error)
- X_1 = 0 if related, 1 if unrelated
- X_2 = 0 if opponent right, 1 if opponent wrong
- Results:
  - Related, Right: $\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_{12} X_1 X_2$
  - $= \beta_0 + \beta_1(0) + \beta_2(0) + \beta_{12}(0)(0)$
  - Most of this is 0 and drops out

Dummy Coding

- $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_{12} X_1 X_2$ (+ random effects, error)
- X_1 = 0 if related, 1 if unrelated
- X_2 = 0 if opponent right, 1 if opnonent wrong
- Results:
  - Related, Right: β_0
  - Unrelated, Right: $\beta_0 + \beta_1$
  - Related, Wrong: $\beta_0 + \beta_2$
  - Unrelated, Wrong: $\beta_0 + \beta_1 + \beta_2 + \beta_{12}$

If we create the equation for all 4 conditions...

Dummy Coding

- $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_{12} X_1 X_2$ (+ random effects, error)
- X_1 = 0 if related, 1 if unrelated
- X_2 = 0 if opponent right, 1 if opnonent wrong
- Results:
  - Related, Right: β_0
  - Unrelated, Right: $\beta_0 + \beta_1$
  - Related, Wrong: $\beta_0 + \beta_2$
  - Unrelated, Wrong: $\beta_0 + \beta_1 + \beta_2 + \beta_{12}$

We see that, here, β_1 = Difference between **Related, Right** and **Unrelated, Right**

Contrast Coding

- $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_{12} X_1 X_2$ (+ random effects, error)
- X_1 = -0.5 if related, 0.5 if unrelated
- X_2 = -0.5 if opponent right, 0.5 if opponent wrong
- Results:
  - Related, Right: $\beta_0 - 0.5\beta_1 - 0.5\beta_2 + \beta_{12}(-0.5)(-0.5)$
  - Unrelated, Right: $\beta_0 + 0.5\beta_1 - 0.5\beta_2 + \beta_{12}(0.5)(-0.5)$
  - Related, Wrong: $\beta_0 - 0.5\beta_1 + 0.5\beta_2 + \beta_{12}(-0.5)(0.5)$
  - Unrelated, Wrong: $\beta_0 + 0.5\beta_1 + 0.5\beta_2 + \beta_{12}(0.5)(0.5)$

Contrast Coding

- $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_{12} X_1 X_2$ (+ random effects, error)
- X_1 = -0.5 if related, 0.5 if unrelated
- X_2 = -0.5 if opponent right, 0.5 if opponent wrong
- Results:
  - Related, Right: $\beta_0 - 0.5\beta_1 - 0.5\beta_2 + \beta_{12}(-0.5)(-0.5)$
  - Related, Wrong: $\beta_0 - 0.5\beta_1 + 0.5\beta_2 + \beta_{12}(-0.5)(0.5)$
  - Unrelated, Right: $\beta_0 + 0.5\beta_1 - 0.5\beta_2 + \beta_{12}(0.5)(-0.5)$
  - Unrelated, Wrong: $\beta_0 + 0.5\beta_1 + 0.5\beta_2 + \beta_{12}(0.5)(0.5)$

I switched the order of these rows to make the next step easier to see

Contrast Coding

- $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_{12} X_1 X_2$ (+ random effects, error)

β_1 = Difference between **2 related conditions** and **2 unrelated conditions**

- …l, 0.5 if unrelated
- …ent right, 0.5 if opponent wrong
- Results:
  - Related, Right: $\beta_0 - 0.5\beta_1 - 0.5\beta_2 + \beta_{12}(-0.5)(-0.5)$
  - Related, Wrong: $\beta_0 - 0.5\beta_1 + 0.5\beta_2 + \beta_{12}(-0.5)(0.5)$
  - Unrelated, Right: $\beta_0 + 0.5\beta_1 - 0.5\beta_2 + \beta_{12}(0.5)(-0.5)$
  - Unrelated, Wrong: $\beta_0 + 0.5\beta_1 + 0.5\beta_2 + \beta_{12}(0.5)(0.5)$

Same between 2 related and 2 unrelated conditions:

Outline

- ~~The Problem~~
- ~~Recap of Coding~~
- ~~Paramoter Testing~~
- ~~Simple Effects & Main Effects~~
- ~~More Detailed Explanation~~
- How to Do Contrast Coding
- Continuous Predictors

How to Do Contrast Coding

- SEE your current coding:
```
contrasts(Dataframe$Variable)
```

- CHANGE your coding:
```
contrasts(Dataframe$Variable) =
c(-0.5,0.5)
```

- With more than 2 levels, set *multiple* contrasts:
```
contrasts(Dataframe$Variable) =
cbind(c(-0.33,-0.33,0.66),
c(-0.5,0.5,0))
```

How to Do Contrast Coding

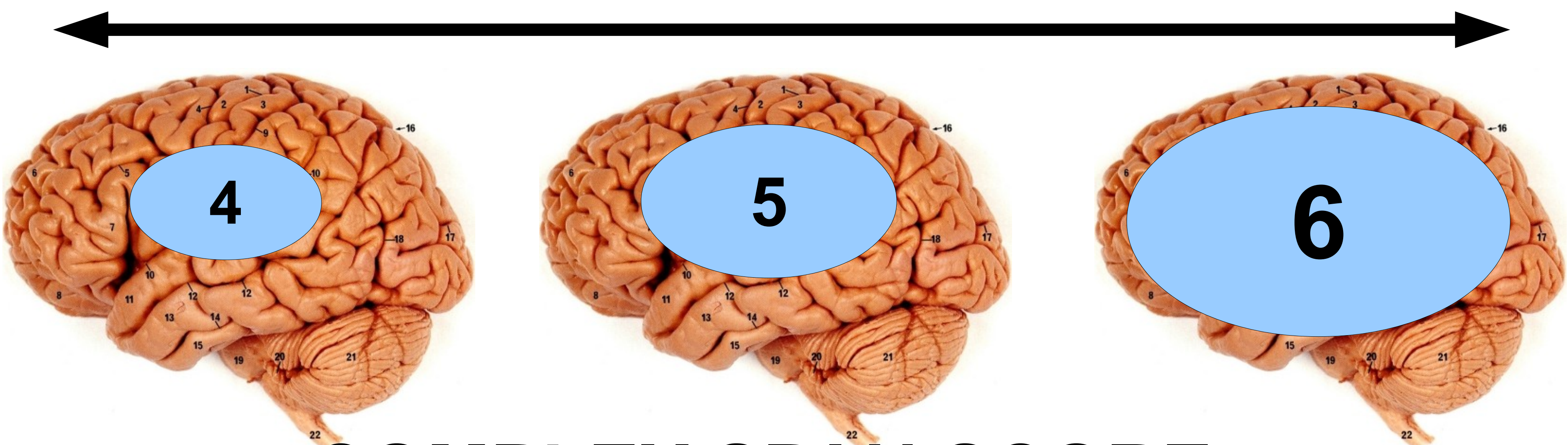
- To get back to **dummy** coding...
- Could set the coding *manually*
  `contrasts(Dataframe$Variable)= c(0,1)`
- SHORTCUT!
  `contrasts(Dataframe$Variable) = contr.treatment`

Outline

- ~~The Problem~~
- ~~Recap of Coding~~
- ~~Paramoter Testing~~
- ~~Simple Effects & Main Effects~~
- ~~More Detailed Explanation~~
- ~~How to Do Contrast Coding~~
- Continuous Predictors

Continuous Predictors

- So far, we've looked at *categorical* predictors
- What about *continuous* predictors?
  - e.g. do **online processing resources** predict use of pitch accenting information in discourse comprehension?



COMPLEX SPAN SCORE

Continuous Predictors

- Again, **by default**, **pitch accent** is evaluated when **span score** = 0

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots$$

- So **main effect** of **pitch accent** represents what pitch accent does when you have no working memory
  - May be **uninformative** (as in this case)
  - Nobody has span score of 0

Continuous Predictors

- Alternative: **CENTER** the **continuous predictor**
- So 0 is now the *mean* span score

Subject 23's span score: 4.43

Mean: 5.0

→ **CENTERING** →

Subject 23's span score: -0.57

Mean: 0.0

- Now, **main effect** of **pitch accent** represents what pitch accent does for you if you have *average* span score
  - "Jane Average"'s pitch accenting effect
  - More informative!

Centering a Variable in R

- **Replace** the original variable w/ mean-centered version

```
Dataframe$Variable =
Dataframe$Variable -
mean(Dataframe$Variable)
```

- **Keep** the original variable and create a new mean-centered one called `Variable.c`:

```
Dataframe$Variable.c =
Dataframe$Variable -
mean(Dataframe$Variable)
```

- **Default**
  - Main effect of predictor X_1 is when predictor X_2 is at 0
- **Mean Centering**
  - Main effect of predictor X_1 is when predictor X_2 is at its mean

- Again...
  - Both are legitimate statistical tests, but **they test different things**
  - No difference between these 2 when there's no interaction
  - **Doesn't** change the **test of the interaction itself**