Supplemental Resource: Brain and Cognitive Sciences
Statistics & Visualization for Data Analysis & Inference
January (IAP) 2009

# Statistics and Visualization for Data Analysis and Inference

Mike Frank & Ed Vul

IAP 2009

# Classes

1. **Visualization** – how can I see what my data show?
2. **Resampling** – what parts of my data are due to noise?
3. **Distributions** – how do I summarize what I believe about the world?
4. **The Linear Model** – how can I create a simple model of my data?
5. **Bayesian Modeling** – how can I describe the processes that generated my data?

# Classes

1. **Visualization** – how can I see what my data show?
2. **Resampling** – what parts of my data are due to noise?
3. **Distributions** – how do I summarize what I believe about the world?
4. **The Linear Model** – how can I create a simple model of my data?
5. **Bayesian Modeling** – how can I describe the processes that generated my data?

ALL I EVER WANTED TO KNOW ABOUT

# THE LINEAR MODEL

BUT WAS AFRAID TO ASK

# Outline

1. Introducing the linear model
   - the linear model as a model of data
   - what it is, how it works, how it's fit
   - inc. $r^2$, ANOVA, etc
2. A (very) worked example
   - india abacus data
   - logistic regression
   - multi-level/mixed models

# Caveats

- Not necessarily Bayesian
  - Not so many priors and likelihoods
  - Though compatible with this approach
- “Model-driven,” instead
  - making assumptions about where data came from
  - checking those assumptions
  - writing down models that fit data

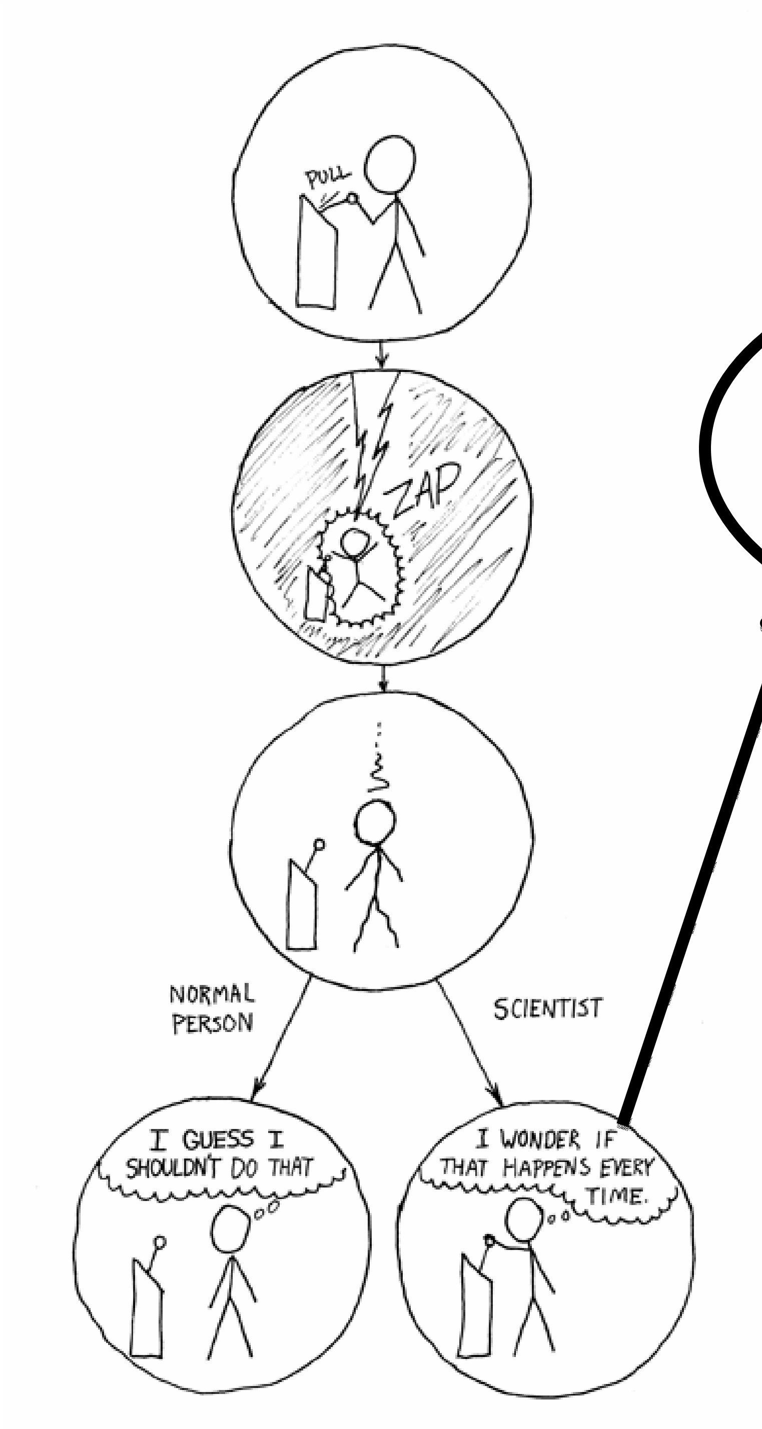
JUST

# THE LINEAR MODEL

# What you will learn

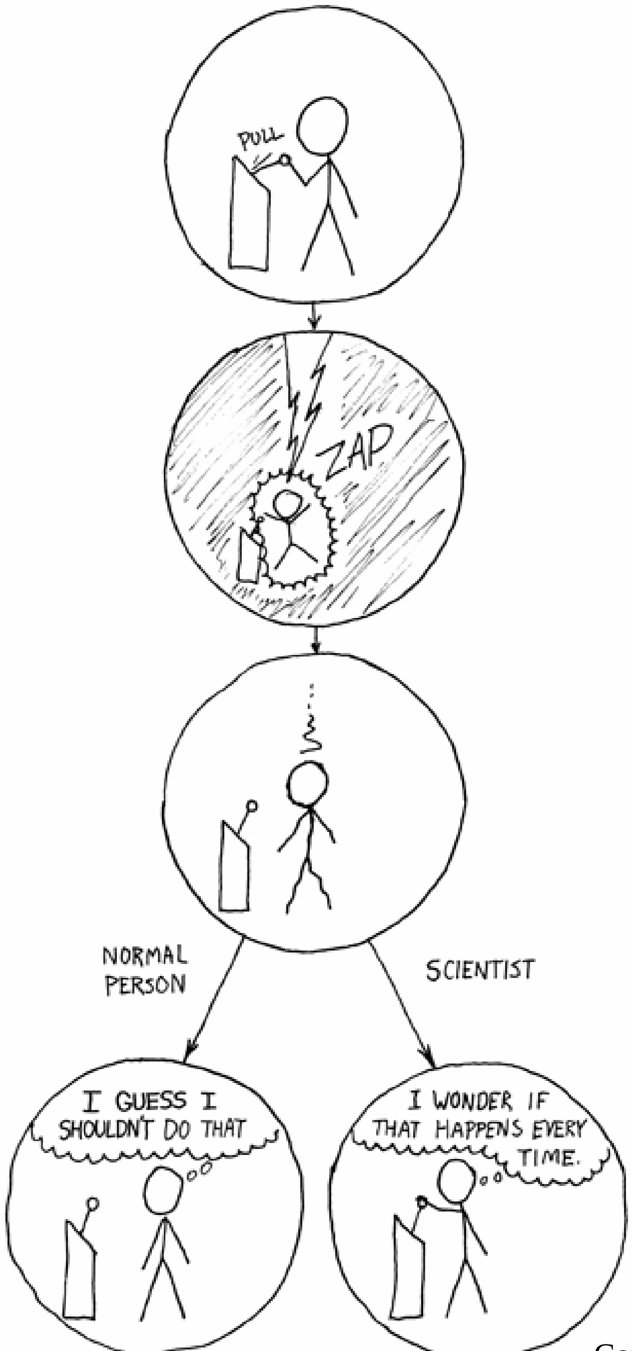
- The linear model is a model of data
  - Consider the interpretation of your model
  - Treat it as a model whose fit should be assessed
- The GLM allows links between linear models and data with a range of distributions
- Multilevel models can be effective tools for fitting data with multiple grains of variation
  - Especially important for subjects/items

# Data




Courtesy of xkcd.org

# Data




Courtesy of xkcd.org

**with hands**

ZAP ZAP ZAP ZAP
ZAP ZAP

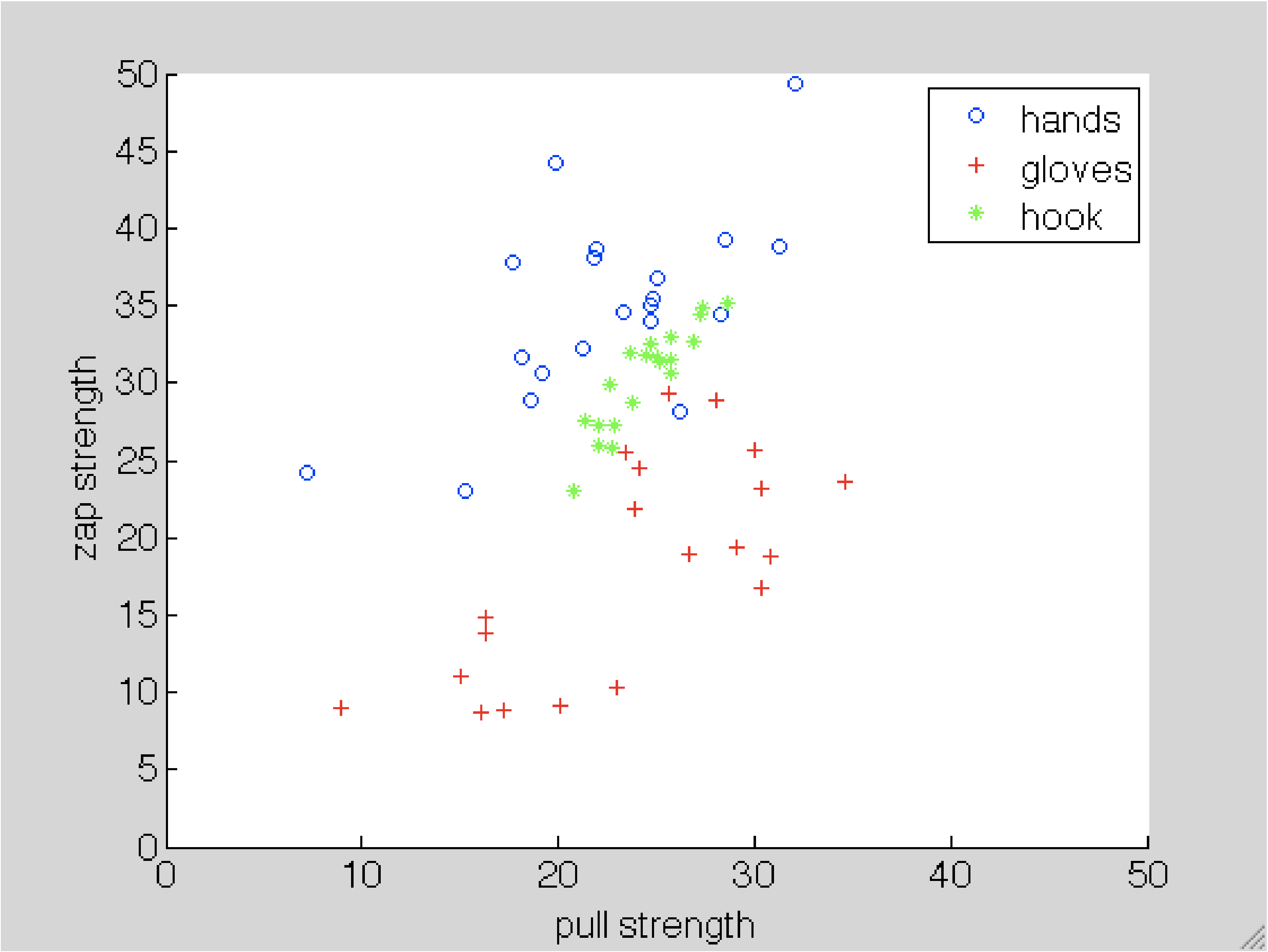
**with gloves**

ZAP ZAP ZAP ZAP ZAP
ZAP

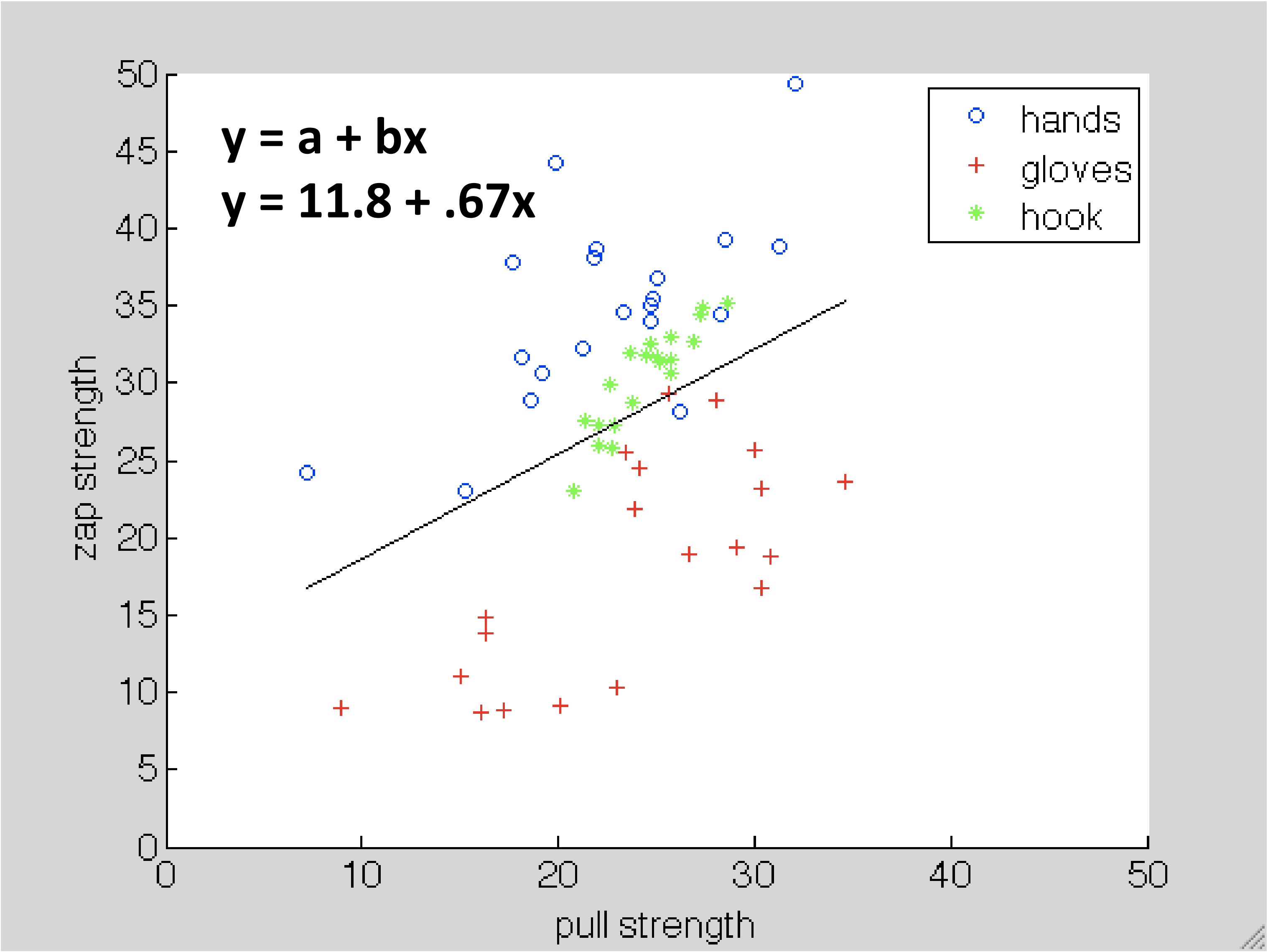
**with a wooden hook**

ZAP ZAP ZAP ZAP ZAP ZAP

# Plotting the data



# Regression, intuitively

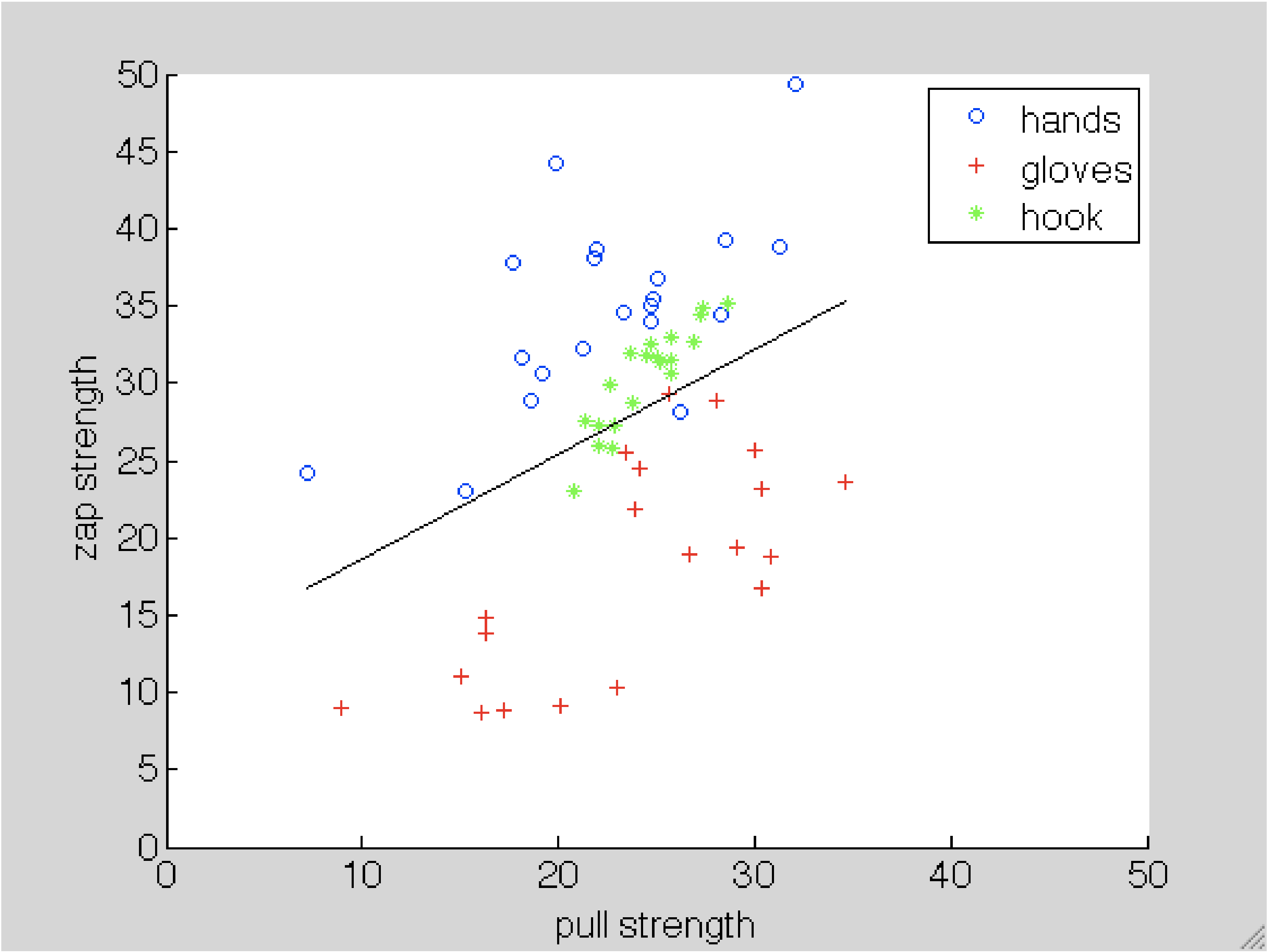


# Regression, computationally

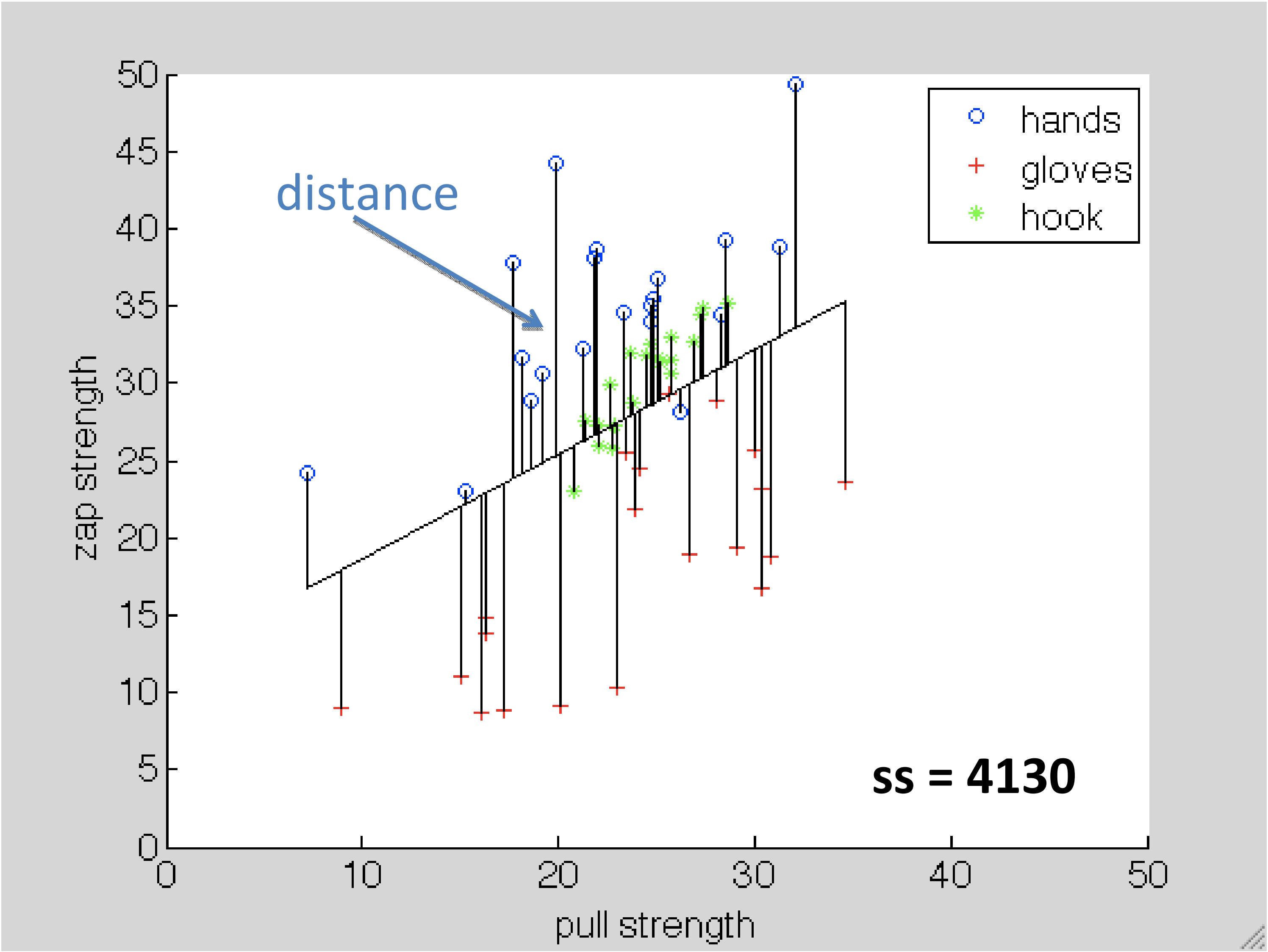
- Fitting a line to data: $y = a + bx$
- How do we fit it?

```
all_pulls = [hand_pulls; glove_pulls; hook_pulls];
all_zaps = [hand_zaps; glove_zaps; hook_zaps];

intercept = ones(size(all_pulls));
[b, b_int, r, r_int, stats] = …
regress(all_zaps,[interceptall_pulls]);

xs = [min(all_pulls) max(all_pulls)];
ys = b(1) + xs*b(2);

line(xs,ys,'Color',[0 0 0])
```

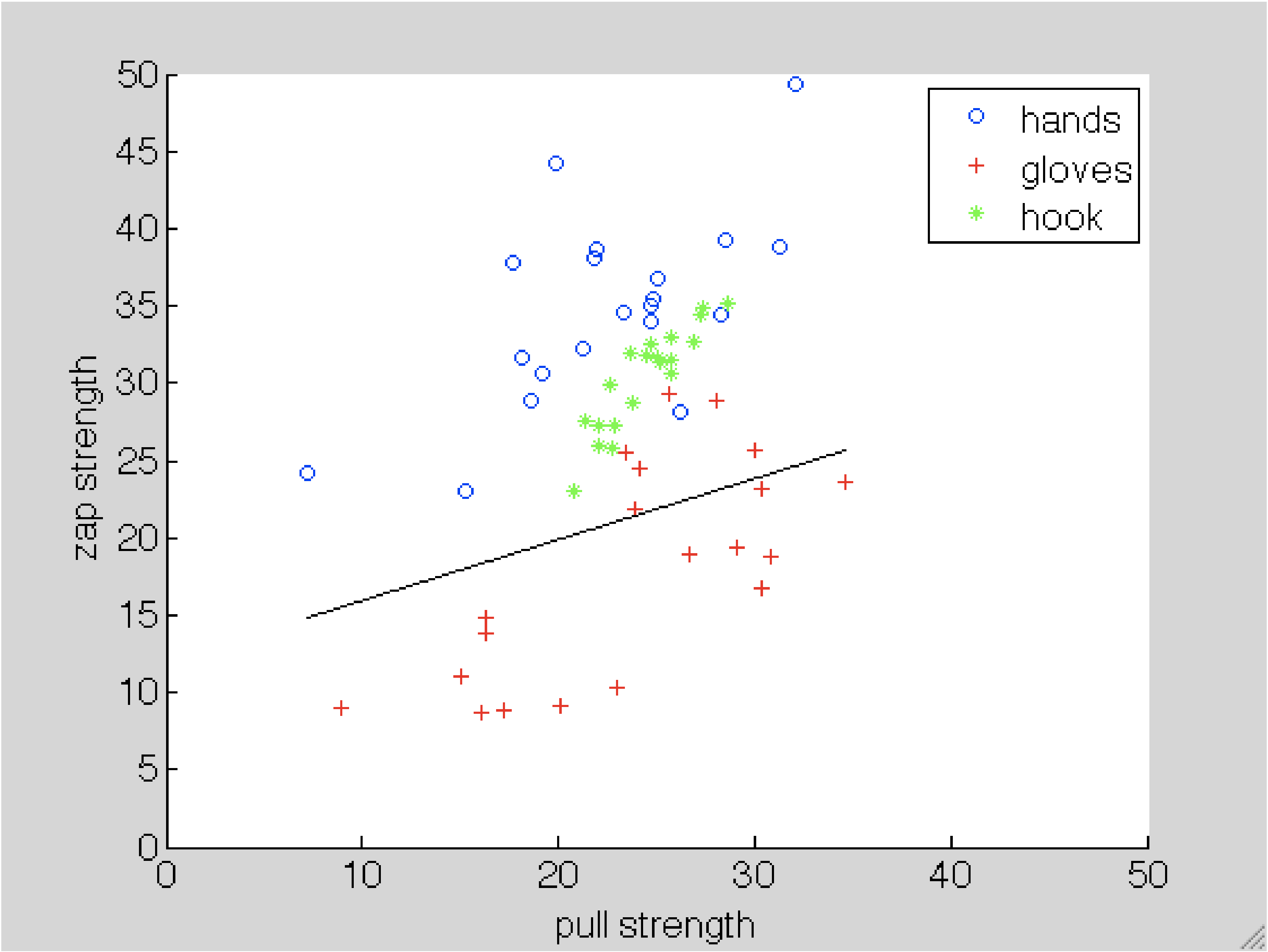
# Regression, really



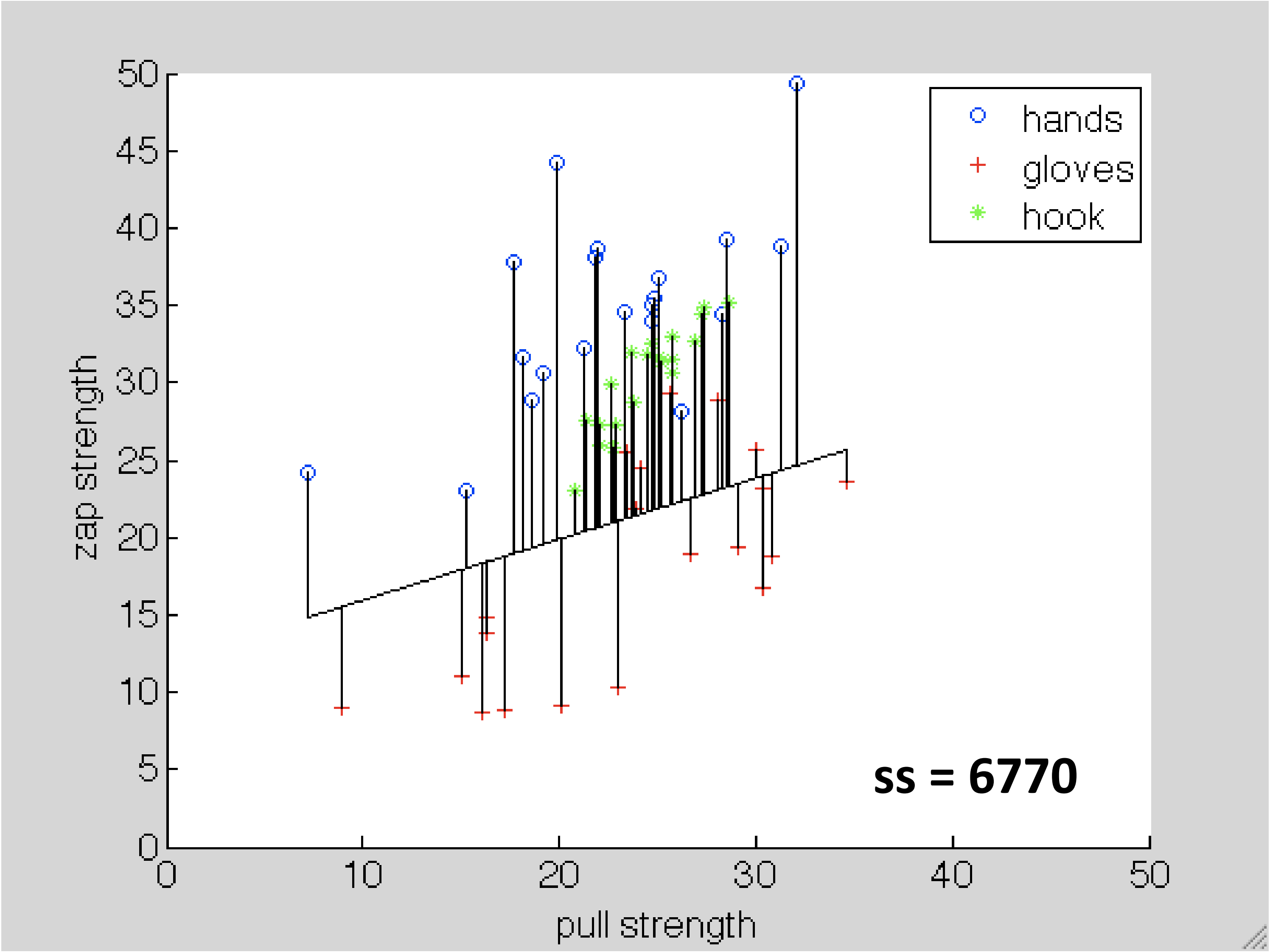
# Regression, really



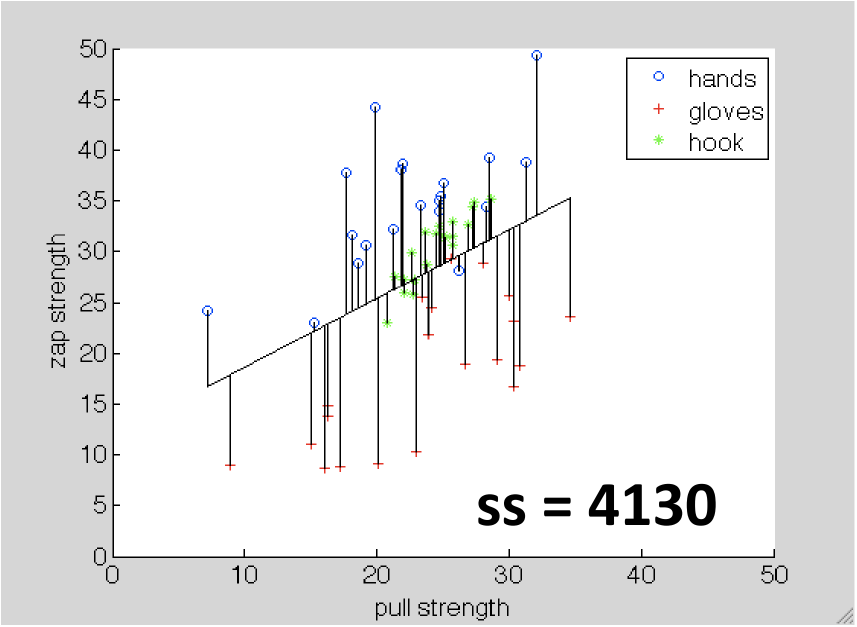
# Regression, really

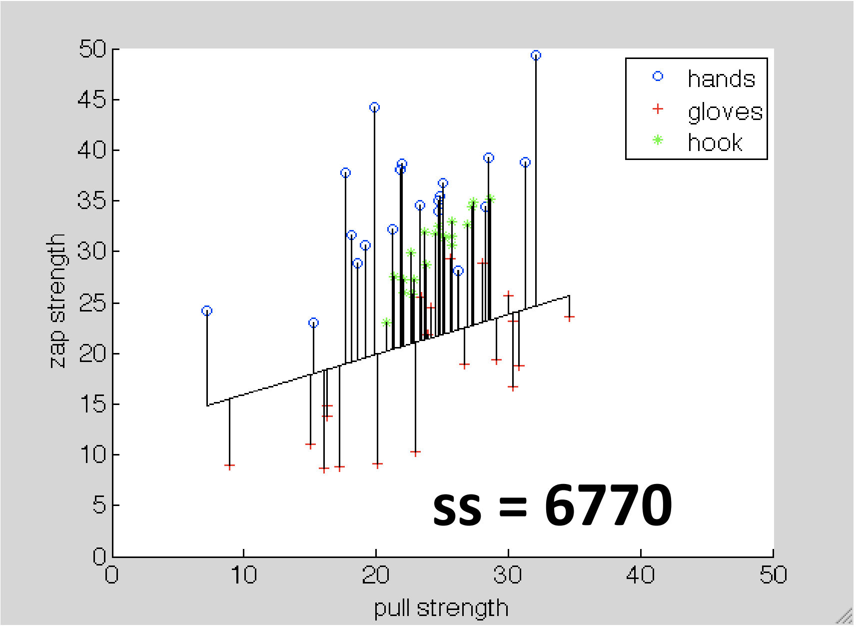


# Regression, really



# Regression, really

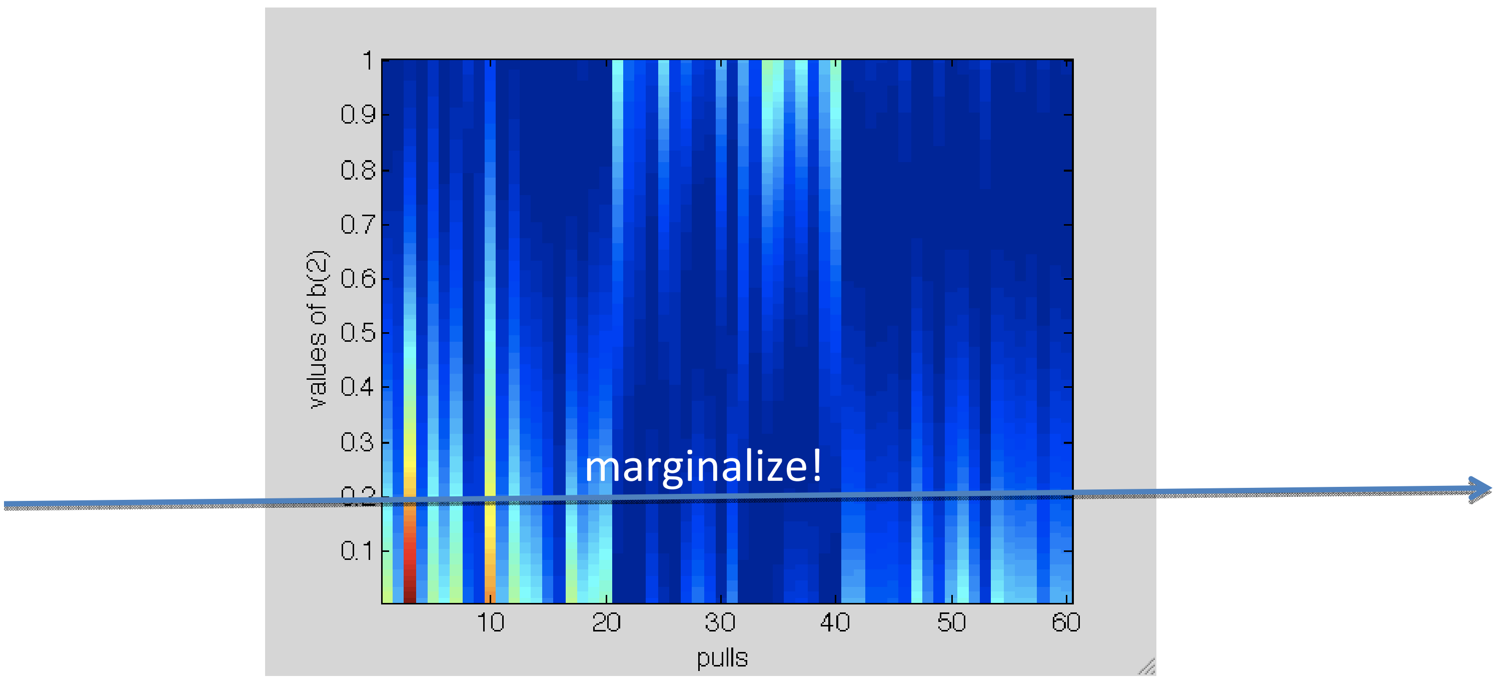


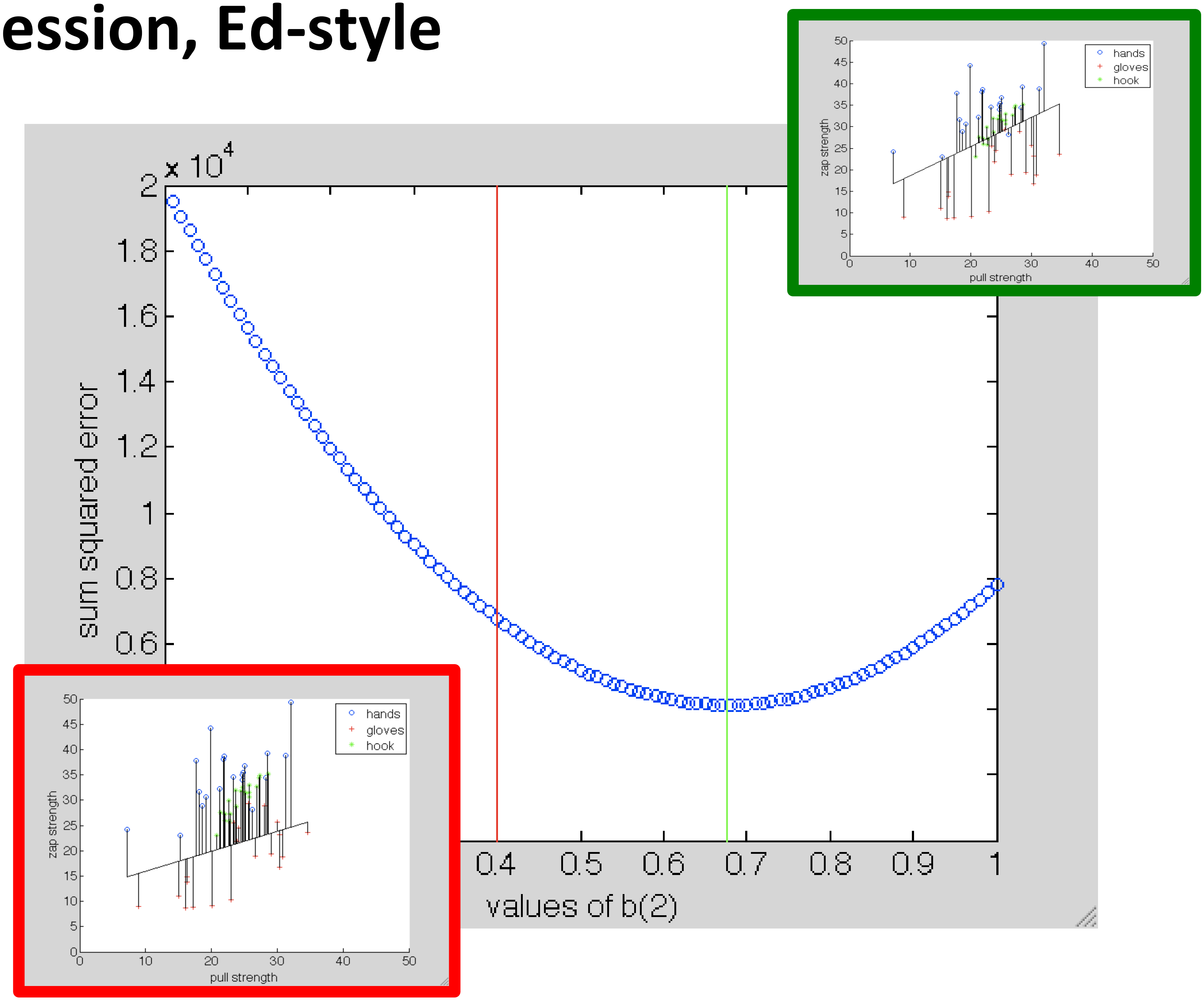

- The principle: minimize summed square error (**ss**)
- Why ss? We won't get into it.

# Regression, Ed-style

```
for i = 1:100
b(i) = i/100;

  for j = 1:length(all_pulls)
squared_error(i,j) = ...
        (all_zaps(j) - (a +b(i)*all_pulls(j)))^2;
  end
end
```



# Regression, Ed-style



# You don’t actually have to do that

- It turns out to be analytic, so the values of B are given by

$$\hat{\boldsymbol{\beta}} = (X^T X)^{-1} X^T \boldsymbol{y}$$

- What is sum squared error but a **likelihood function**?
  - Turns out that what we did was “maximum likelihood estimation”

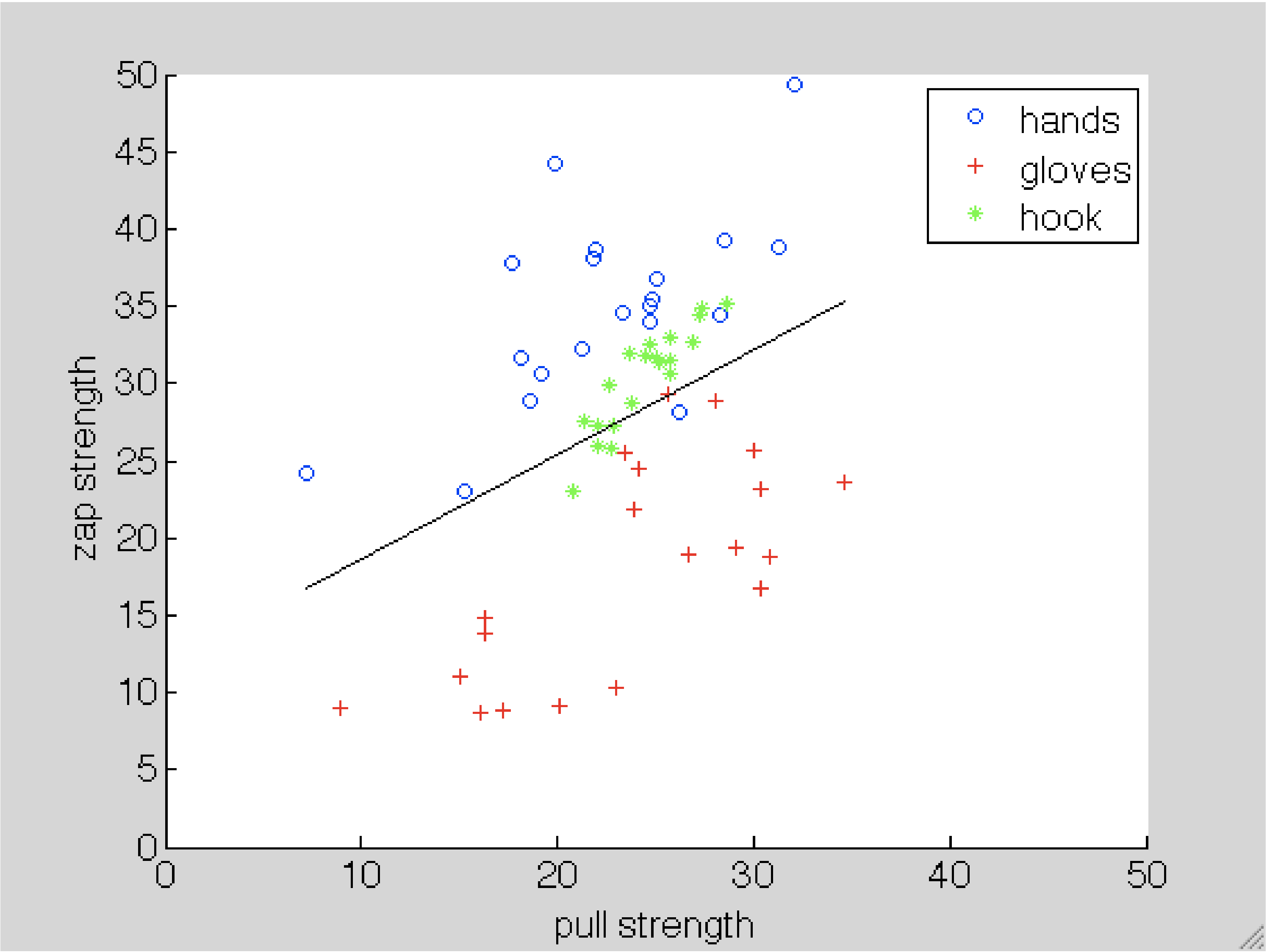
# Regression is a model of data

- I’ve been lying:
  - this error is assumed to be normal
- This is a model of data, in the sense of a **generative process**, a way of generating new data
  - so we can work backwards via Bayes and derivethe same likelihood function
  - least squares is (somewhat) Bayesian
  - and we could easily have put a prior on the coefficient if we had wanted to

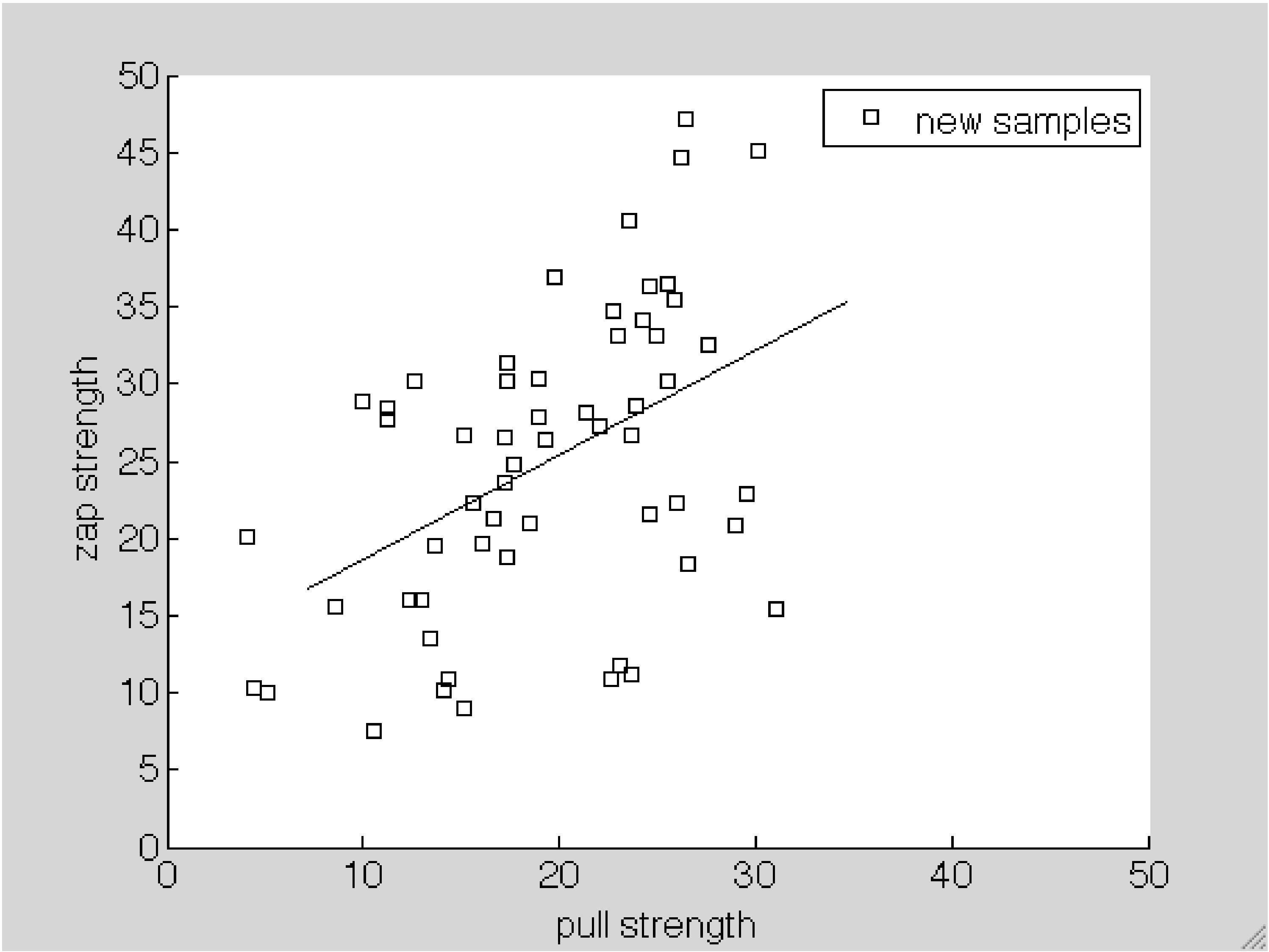
# When you have a model…

- Prediction
  - or, in Bayesian language, “forward sampling”
- Interpretation & evaluation
  - interpreting coefficients
  - $r^2$ (effect size)?
  - residuals
  - coefficient significance
  - ANOVA

# Plotting model and data!



# Prediction

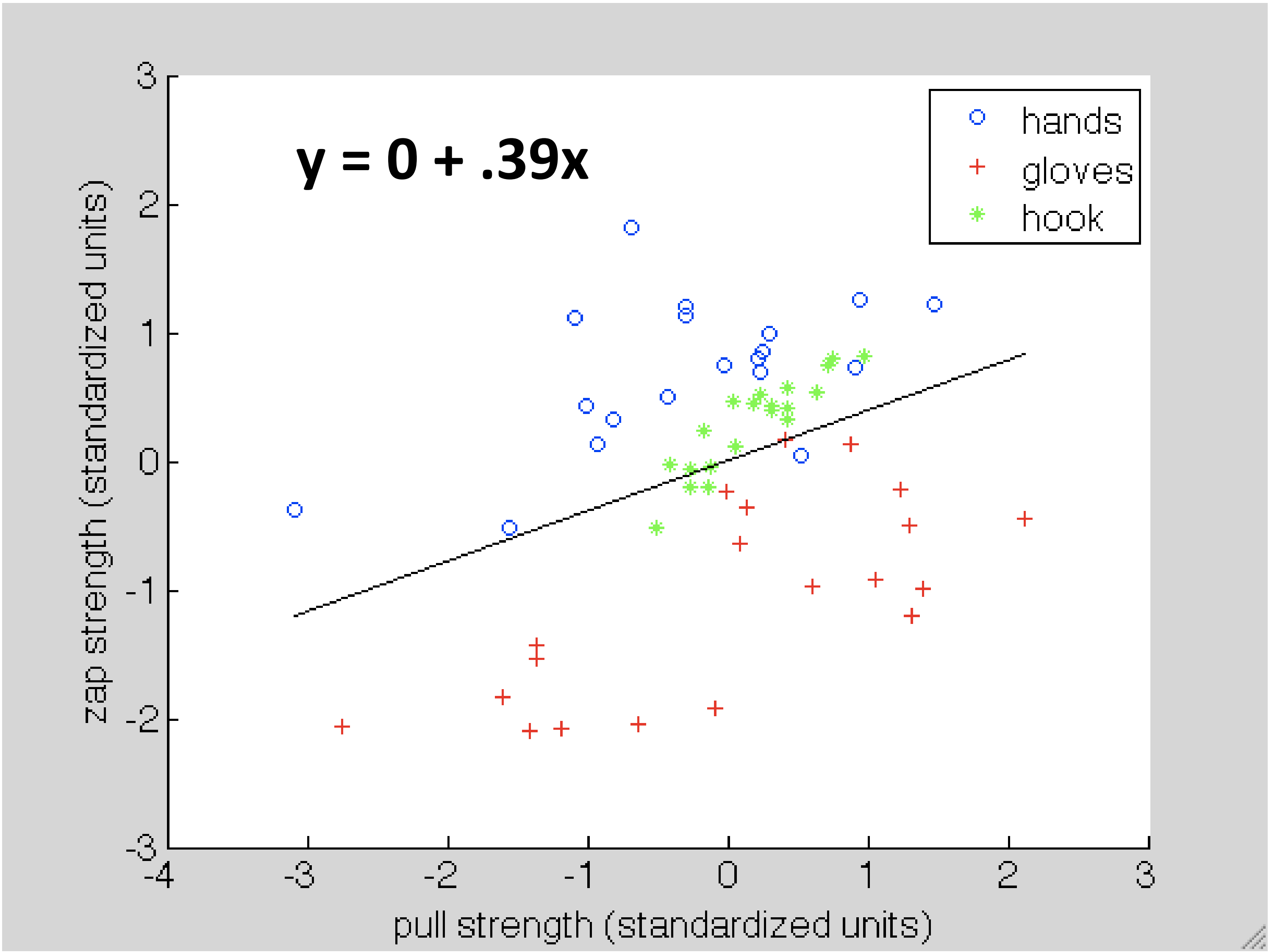


new samples
zap strength
pull strength
0
5
10
15
20
25
30
35
40
45
50
0
10
20
30
40
50

# Coefficients

- We found:
  - intercept = 11.87
    - So if you didn't pull at all, you'd get a ZAP ?
  - slope = 0.67
    - One unit of pulling strength makes the zap .67pts larger
- Standardizing coefficients
  - It can sometimes be useful to z-score your data so that you can interpret the units of the coefficients
  - z-score: (X – mean(X)) / stdev(X)

# Standardized coefficients



# How good is my model?

- In the univariate continuous case, we can compute a correlation coefficient:

$$\rho_{X,Y} = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E((X-\mu_X)(Y-\mu_Y))}{\sigma_X \sigma_Y},$$

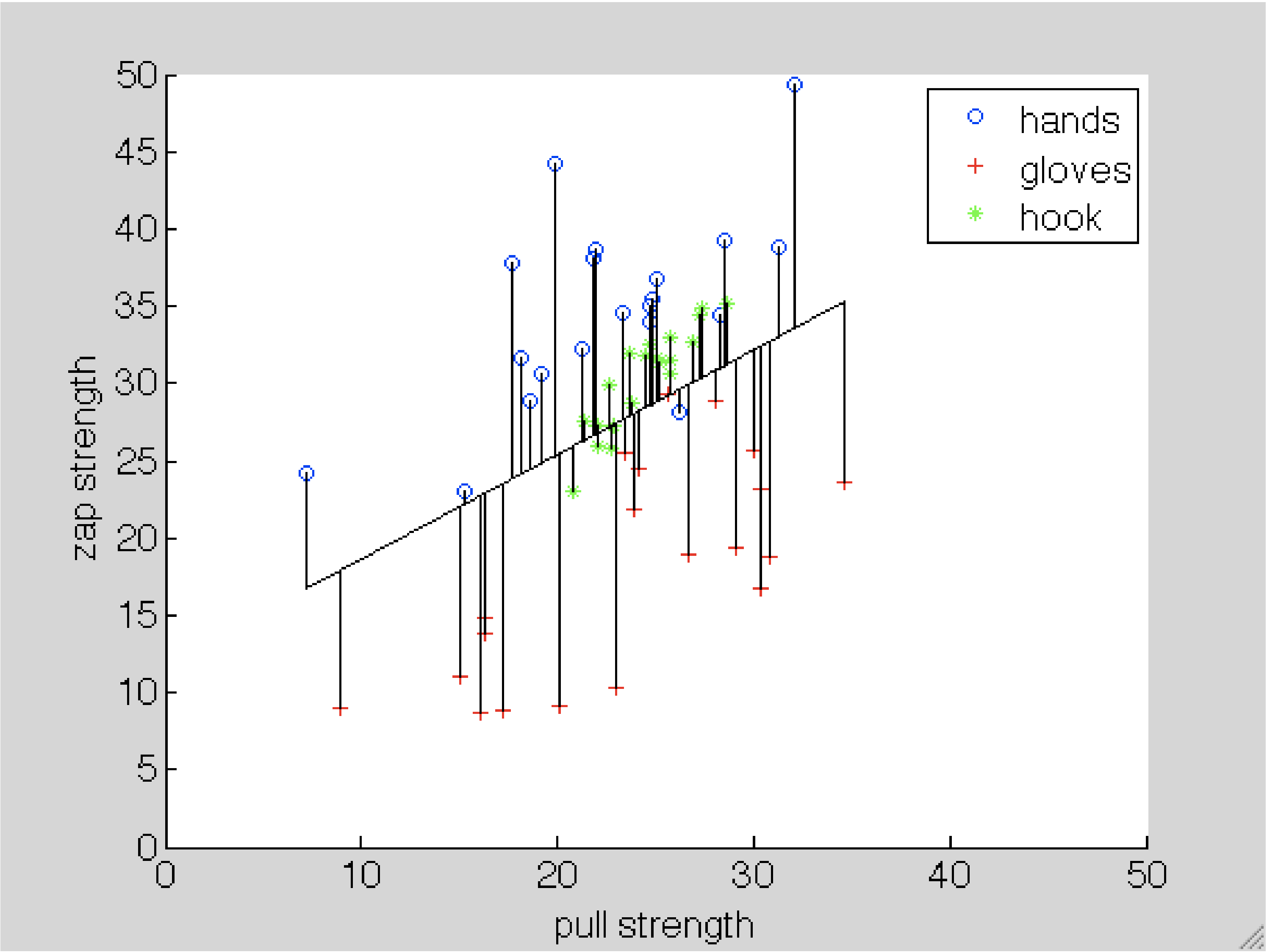
- And then Pearson's $r^2$ ("portion of variance explained) is the square of this number
- But more generally:

$$R^2 \equiv 1 - \frac{SS_{\mathrm{err}}}{SS_{\mathrm{tot}}}.$$

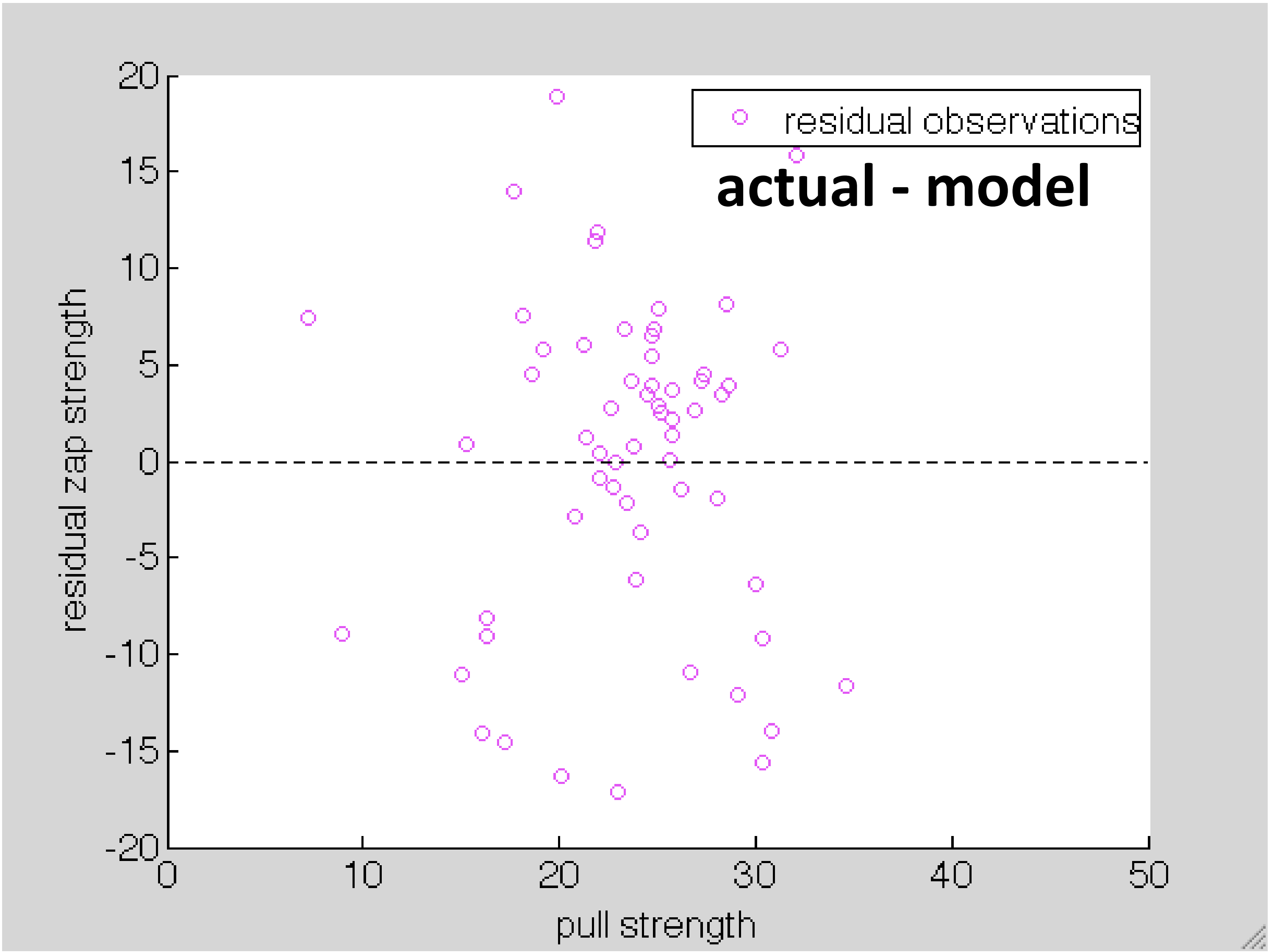
sum of squares for the residuals

sum of squares for the data

# Assessing model fit: residuals



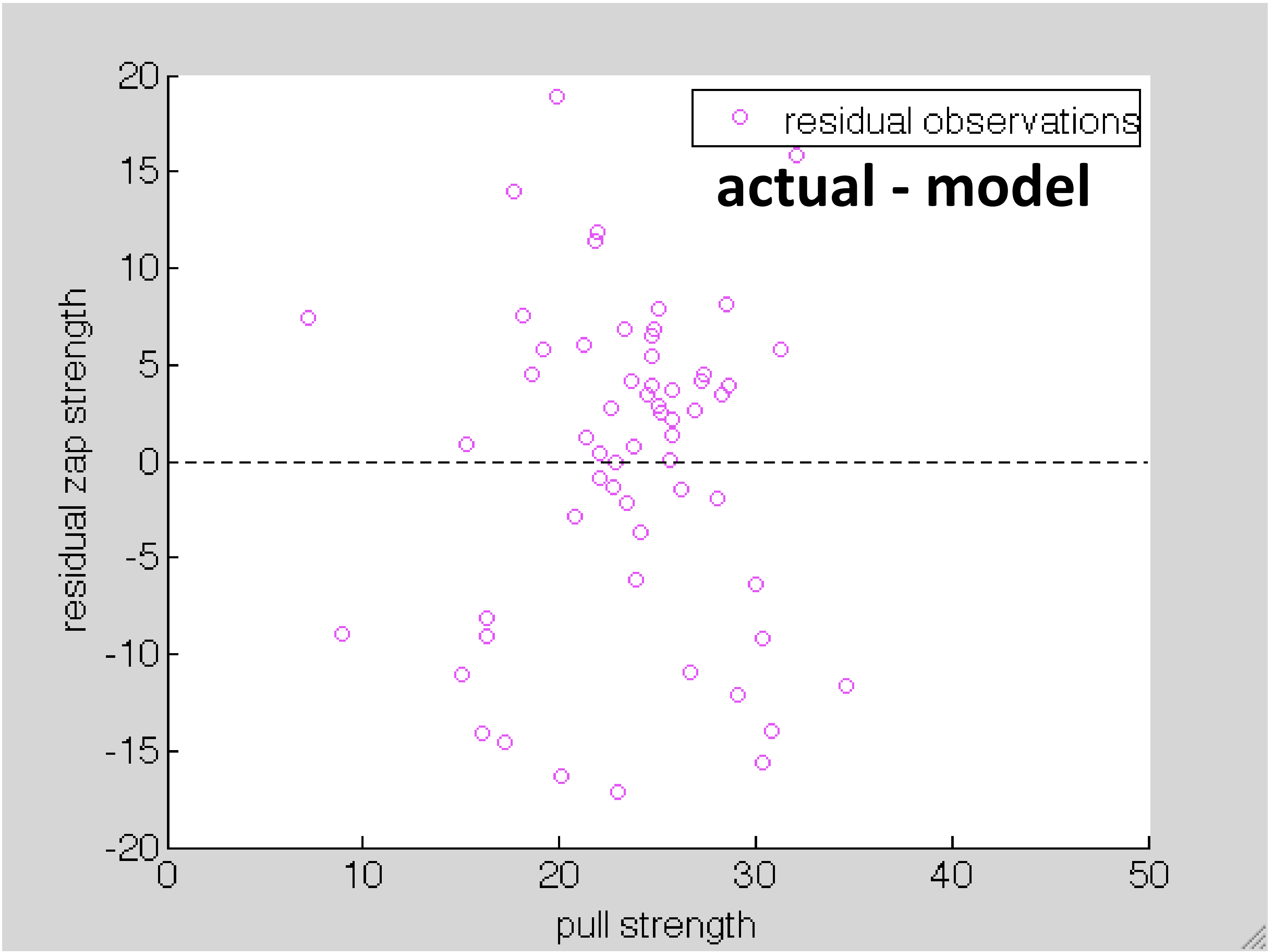
# Assessing model fit: residuals



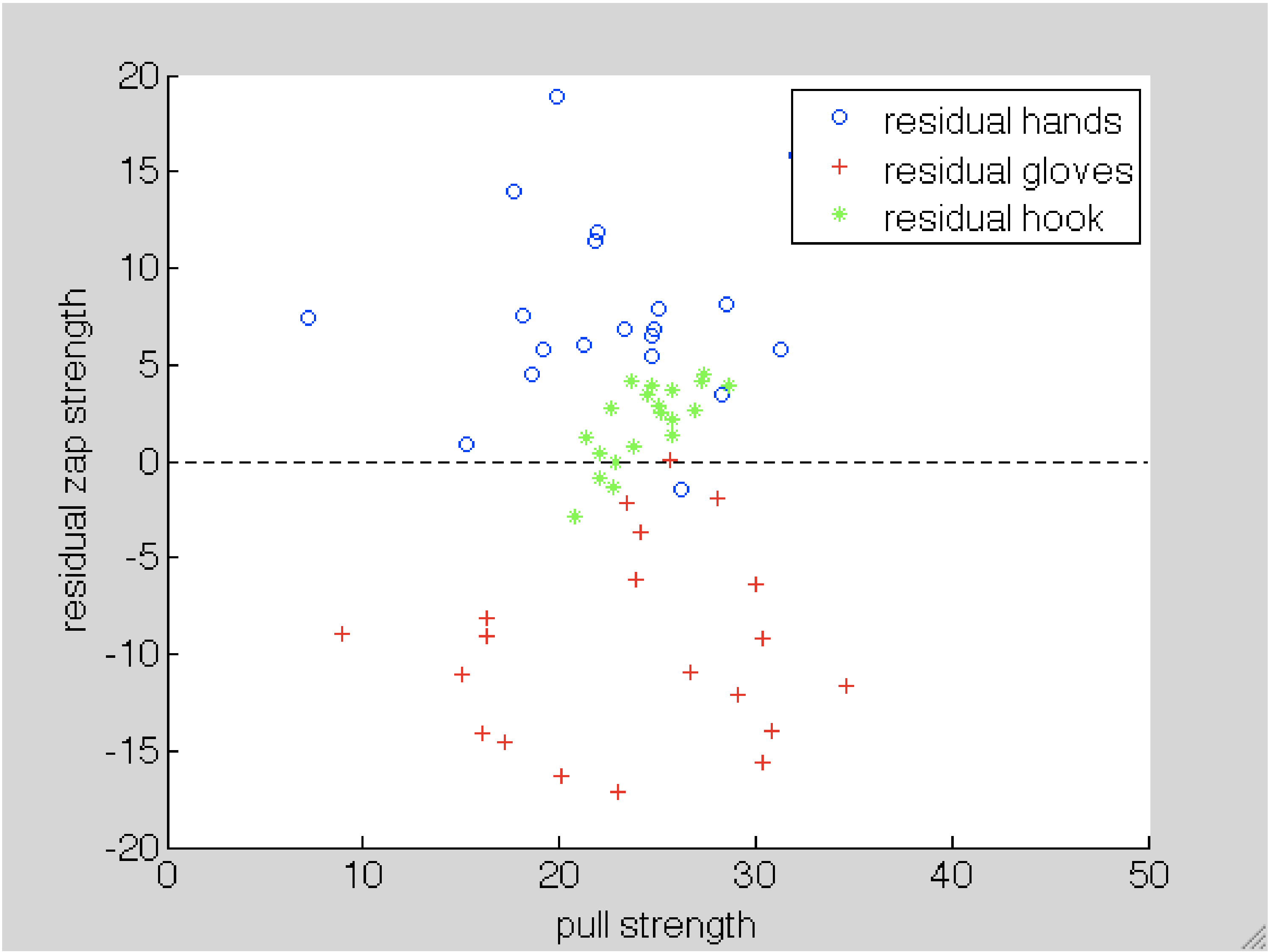
# Assessing model fit

- So now we know $SS_{err}$ (and $SS_{total}$ is easy to find)
  - So $r^2$ is .15, meaning that r = .39 (wait…)
- Another way of looking at it
  - How much better would you predict y if you knew x?
- Why is this important?
  - $r^2$ is a very easily interpretable measure of **effect size**
  - e.g., proportion of variance that is explained (since $SS_{total}$ is “variance”)

# Assessing model fit: residuals



# Assessing model fit: residuals



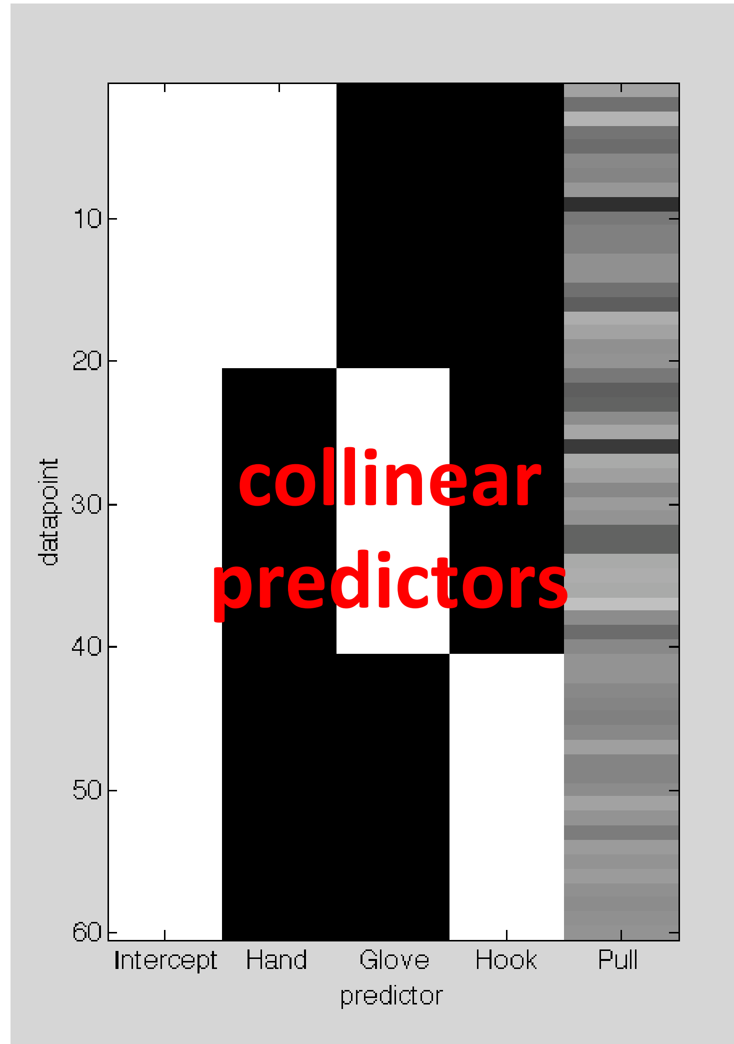
# When it's broke…

- Adding another predictor to the model (pull type)

- This is the beauty of the linear model!

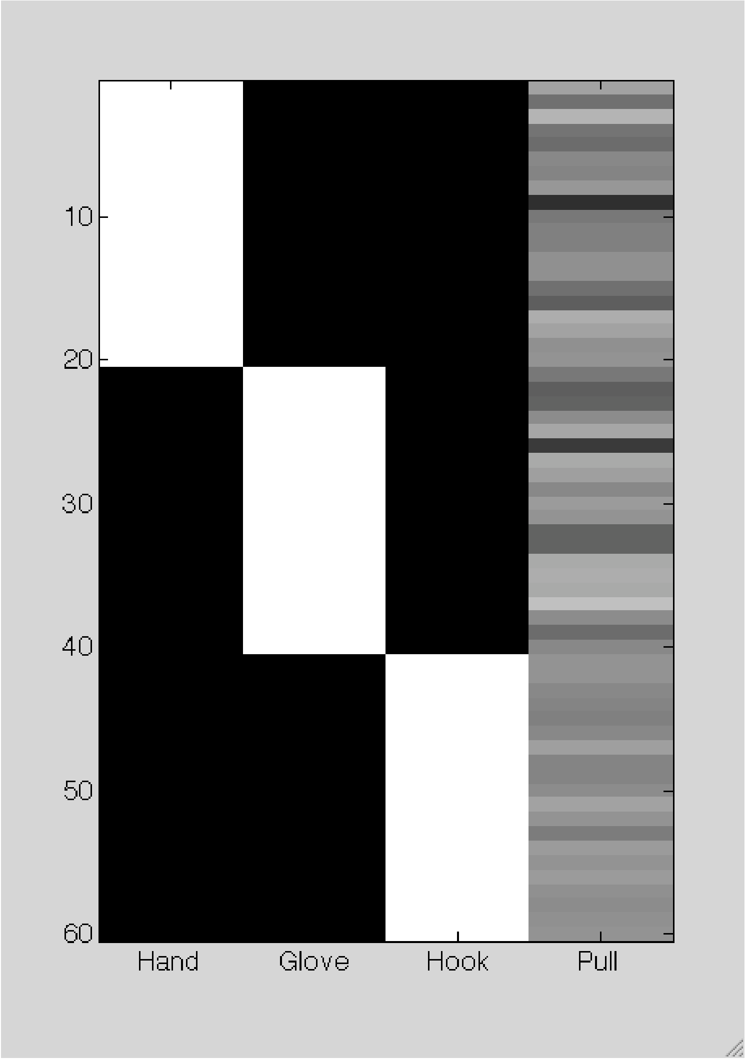
# So how do we do it?

```
intercept = ones(size(all_pulls));
all_pulls = [hand_pulls;
glove_pulls; hook_pulls];
all_zaps = ...
    [hand_zaps; glove_zaps; ...
    hook_zaps];
pull_type = zeros(60,3);
pull_type(1:20,1) = 1;  % hand
pull_type(21:40,2) = 1; % glove
pull_type(41:60,3) = 1; % hook

X1 = [intercept pull_type ...
    all_pulls];
X2 = [pull_typeall_pulls];

% bad
[b, b_int, r, r_int, stats] = ...
   regress(all_zaps,X1);

% good
[b, b_int, r, r_int, stats] = ...
  regress(all_zaps,X2);
```

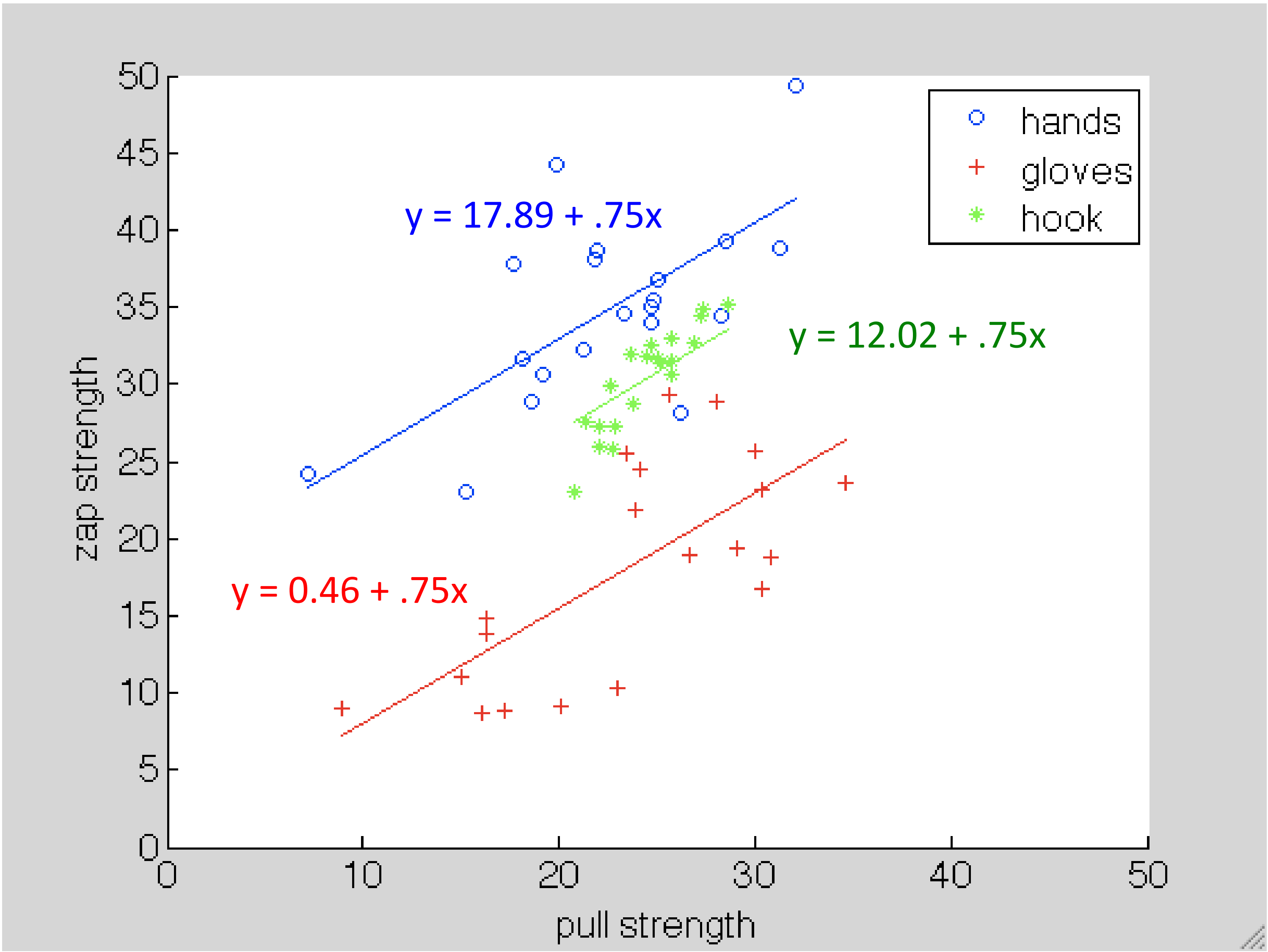


# So how do we do it?

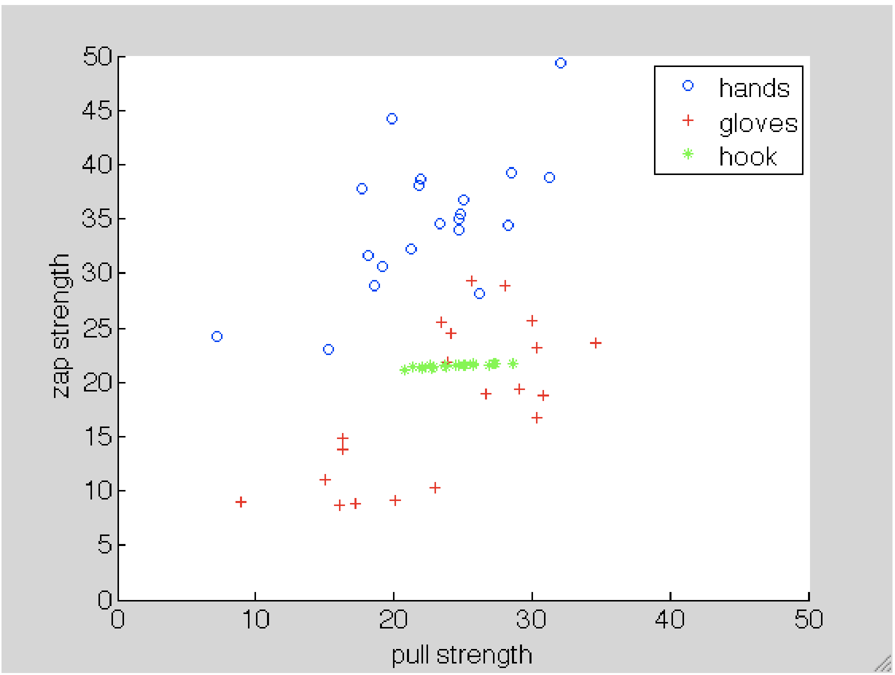
```
intercept = ones(size(all_pulls));
all_pulls = [hand_pulls;
glove_pulls; hook_pulls];
all_zaps = ...
    [hand_zaps; glove_zaps; ...
    hook_zaps];
pull_type = zeros(60,3);
pull_type(1:20,1) = 1;  % hand
pull_type(21:40,2) = 1; % glove
pull_type(41:60,3) = 1; % hook

X1 = [intercept pull_type ...
    all_pulls];
X2 = [pull_typeall_pulls];

% bad
[b, b_int, r, r_int, stats] = ...
   regress(all_zaps,X1);

% good
[b, b_int, r, r_int, stats] = ...
  regress(all_zaps,X2);
```

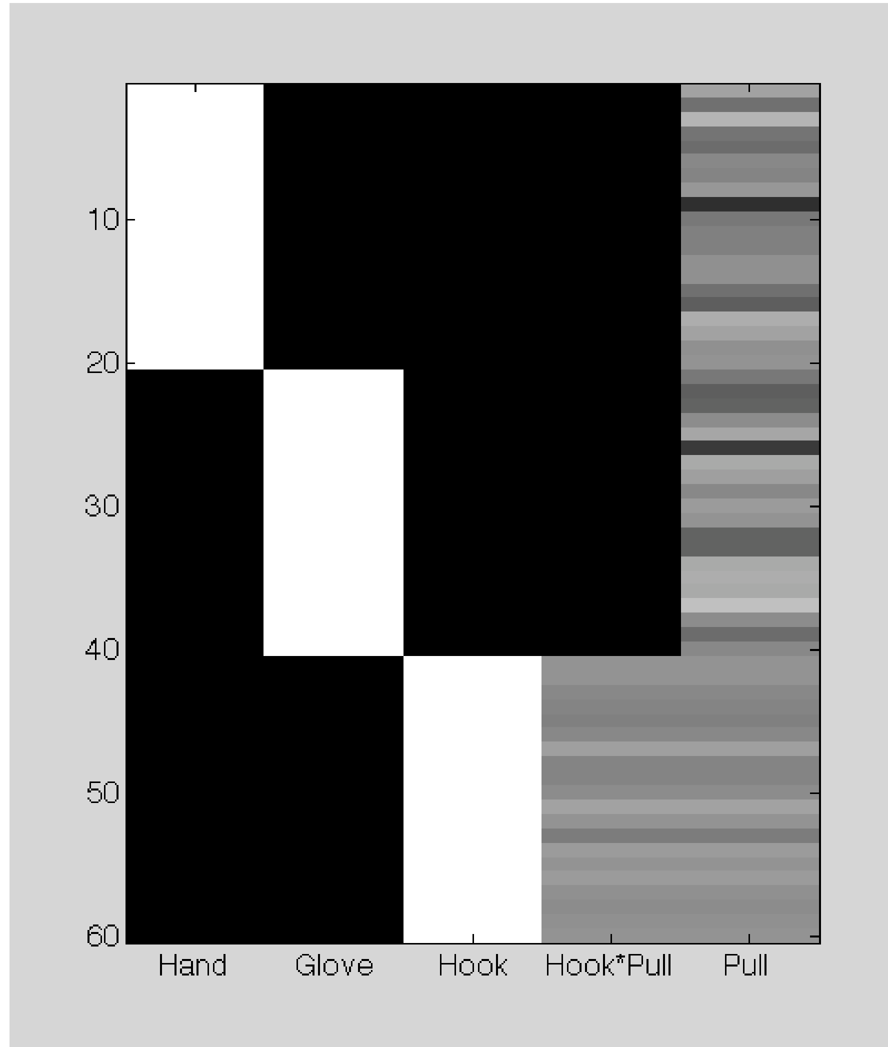


# The resulting model

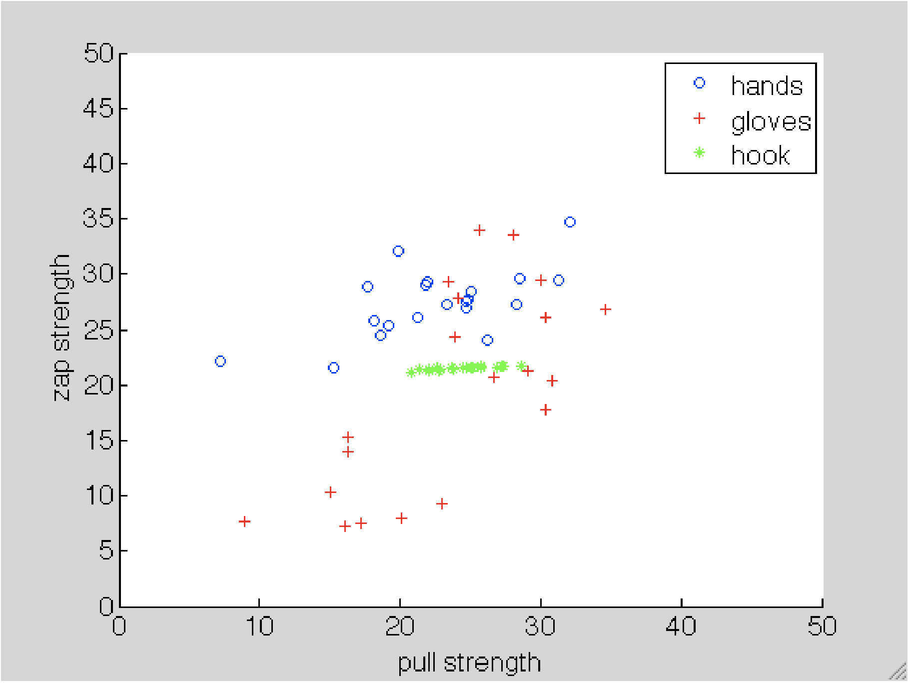


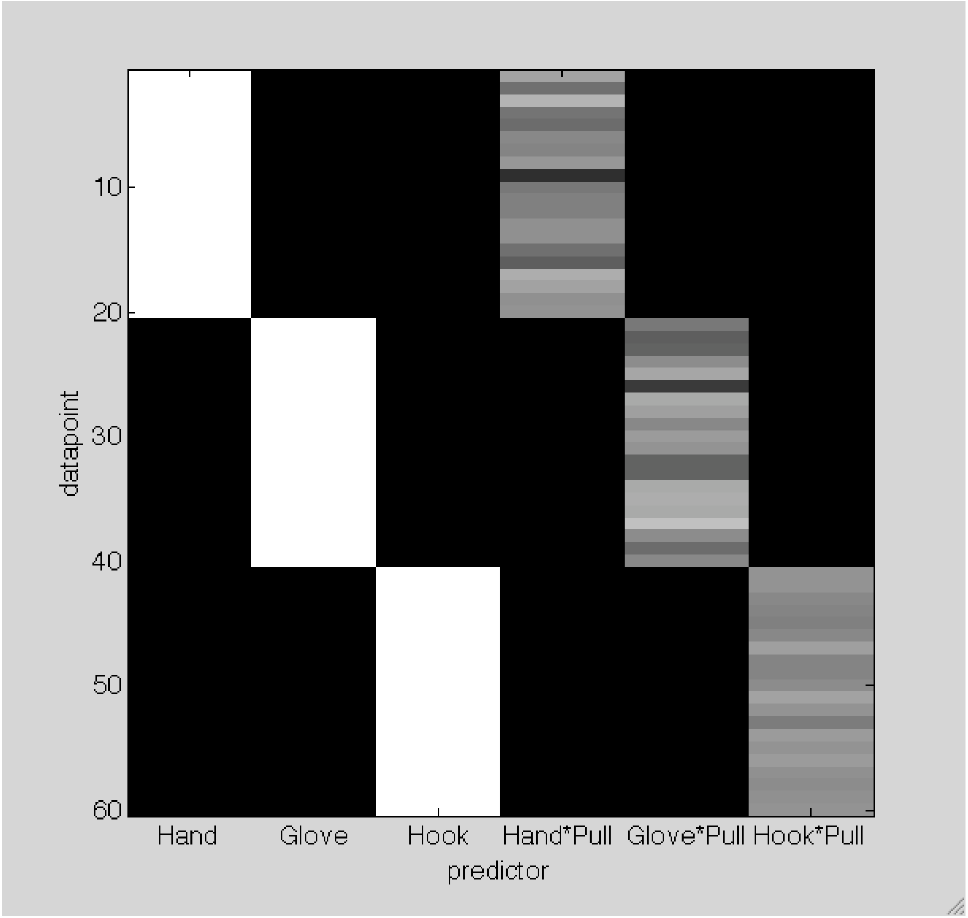
# Aside: interactions



# Aside: Interactions



# Interpretation redux

- What does my model say?
  - each predictor's coefficient is now an intercept value that can be quantified
- How good is my model?
  - $r^2$ for the whole model is now .79
  - "is it significant"? – not a great question
  - is this coefficient/factor/model related to the data?
    - well, $r^2$ is really big
    - in a way that didn't happen by chance?

# Statistical significance and the LM

- Coefficient significance
  – Easy, general, and useful
- Factor significance
  – ANOVA as a way to pool across different coefficients
  – Only applicable in special cases
- Model significance
  – F-test
- Caveat: I'm not really going to tell you how any of these work, just why they work

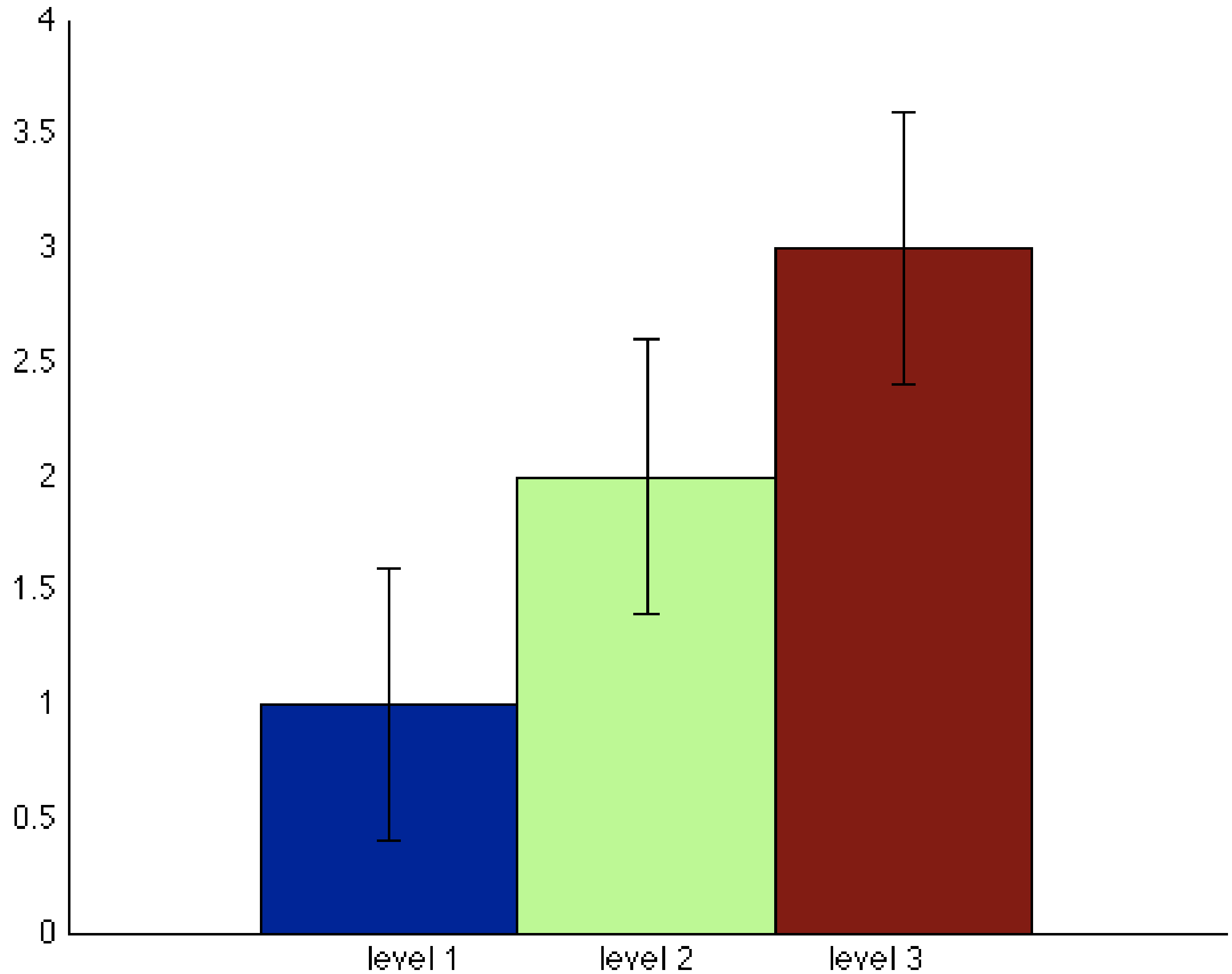
# Coefficient significance

- How can I tell if a particular predictor is statistically significant?
- Look at the error in the model fit
  - In the specific case of a simple linear model, it's analytic
  - SE: $$\hat{\sigma}_j = \sqrt{\frac{S}{n-p-1}\left[(\mathbf{X}^\mathbf{T}\mathbf{X})^{-1}\right]_{jj}}.$$
  - 95% confidence: $\hat{\beta}_j \pm t_{\frac{\alpha}{2},n-p-1}\hat{\sigma}_j.$
  - More generally, you can use simulation to get empirical 95% confidence intervals
  - Remember: you can always grid the model parameters and get bounds on estimates

# Factor significance

- ANOVA (analysis of variance)
  - A method for partitioning the explanatory power of a variable
  - with multiple categorical variables
  - basically this same old sum of squares trick
- ANOVA often treated as a statistical hypothesis test
  - Not as a way of assessing the fit of the underlying model
- When is it useful?
  - When there are multiple categorical factors
    - e.g. multiple coefficients, each with their own error

# For example

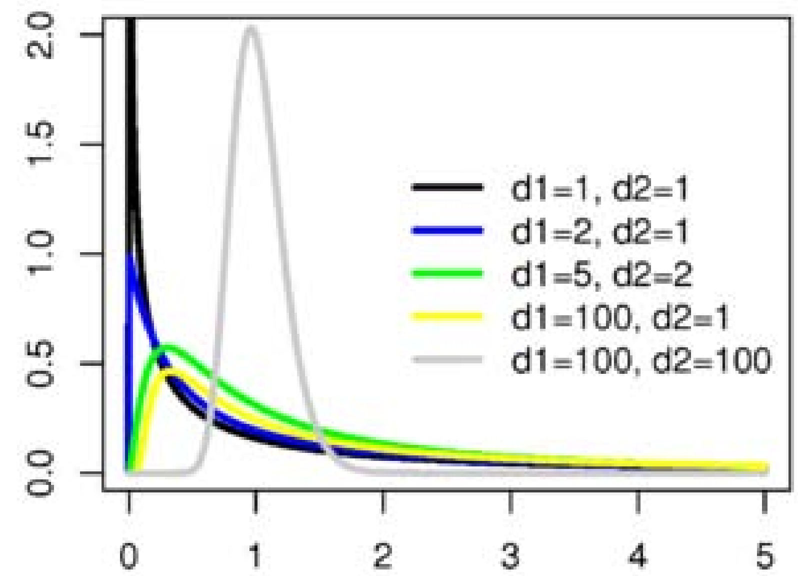


# ANOVA

$$F^* = \frac{\text{MSTR}}{\text{MSE}}$$

$$\text{MSTR} = \frac{\text{SSTR}}{I-1}$$

$$\text{MSE} = \frac{\text{SSE}}{n_T - I}$$




where I is the number of treatments, and $n_T$ is the number of cases

also, F = $$\frac{R^2/(m-1)}{(1-R^2)/(n-m)}$$

# Model significance

- Just a question of whether the overall error explained by the model differs from
- Happens also to be an F distribution
- So you can just do the same test with all of the treatments
- Interpretation is “having the whole model makes you know more than you would if you didn’t have any model”

What now?

# A (VERY) WORKED EXAMPLE

# Worked example outline

- India addition interference
  - Paradigm
  - Dataset and visualization
- Logistic regression
  - Motivation
  - Link function etc.
- Multi-level/mixed models

# Addition demo

33

out of

56

time!

19

# Addition demo

12

35

43

79

95

11

56

81

out of time!

# Adaptive arithmetic

## 10 second time limit, staircased number of addends

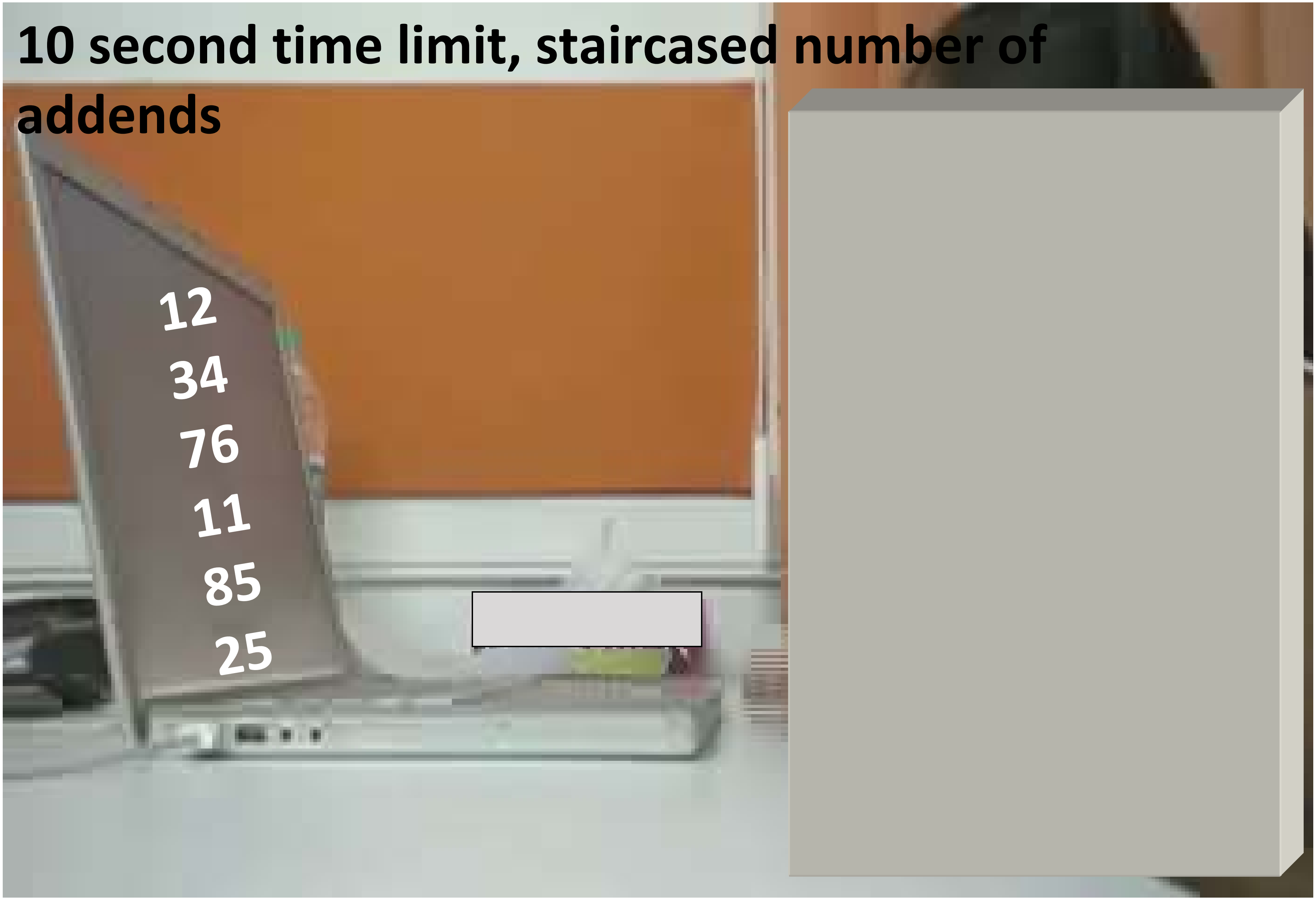
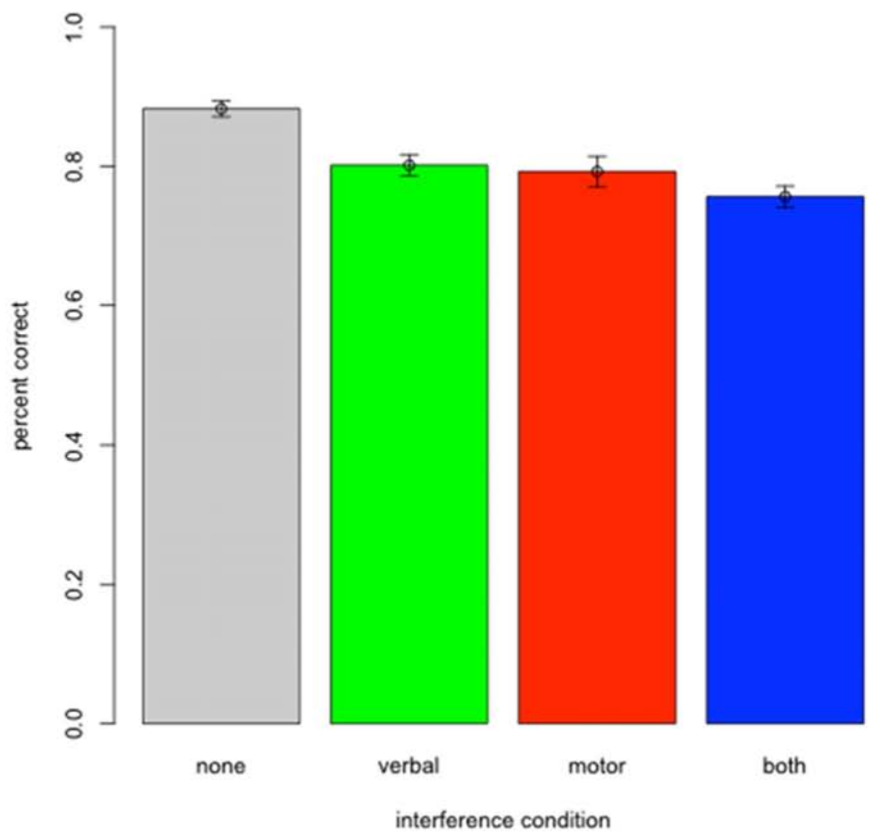
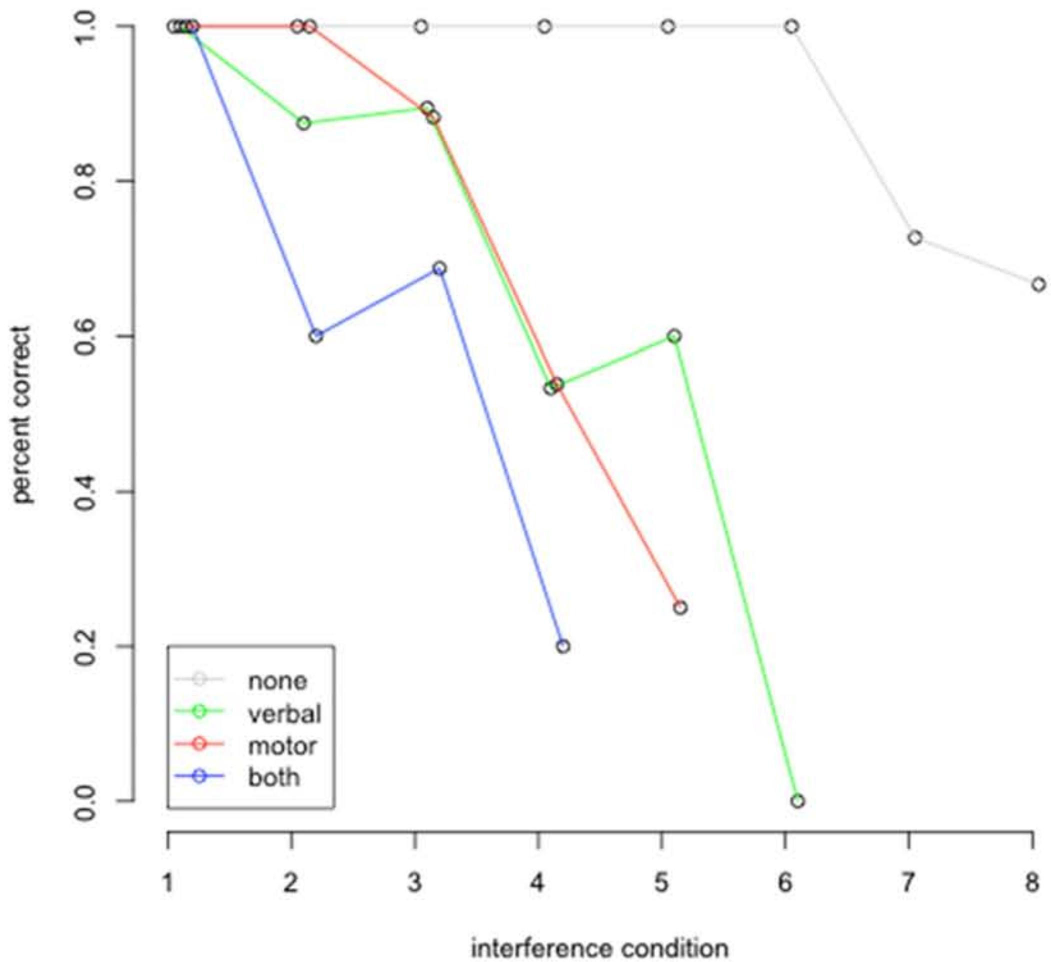



Figure by MIT OpenCourseWare.

# Aggregate data

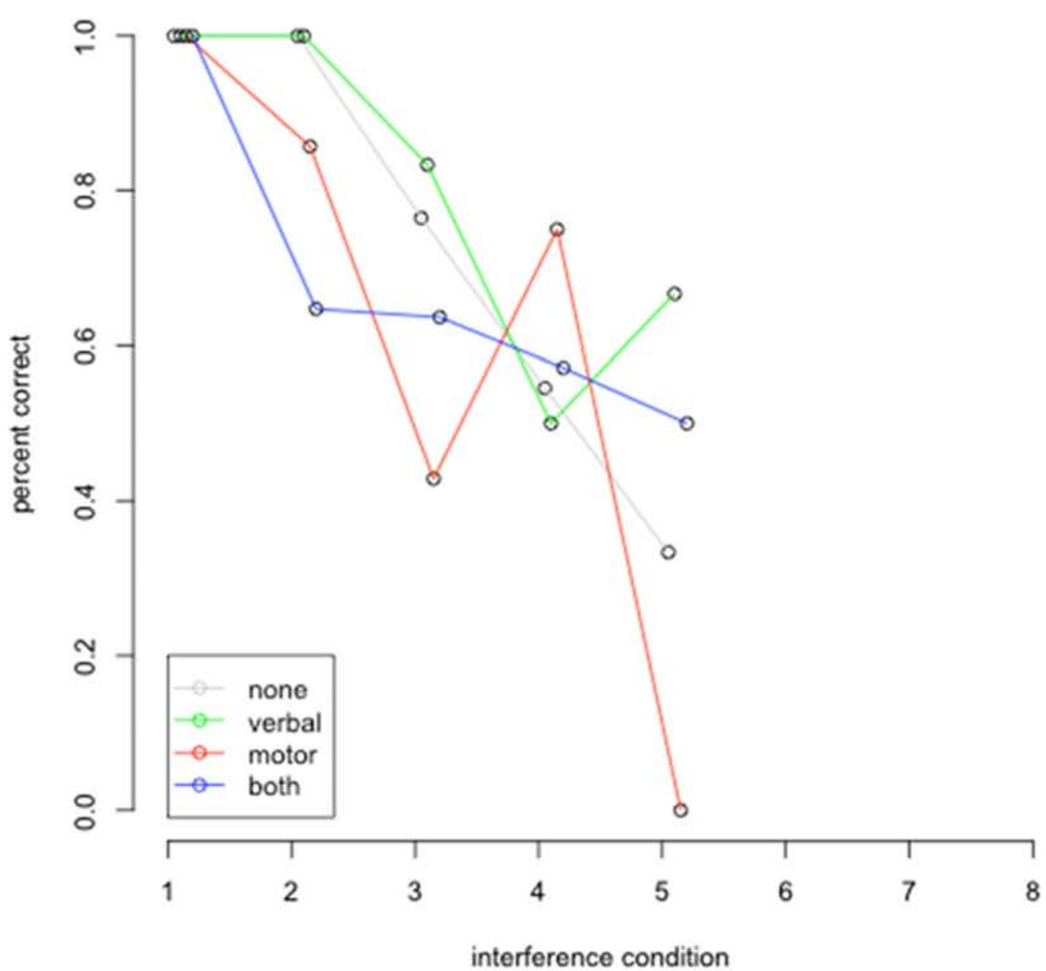


# One subject

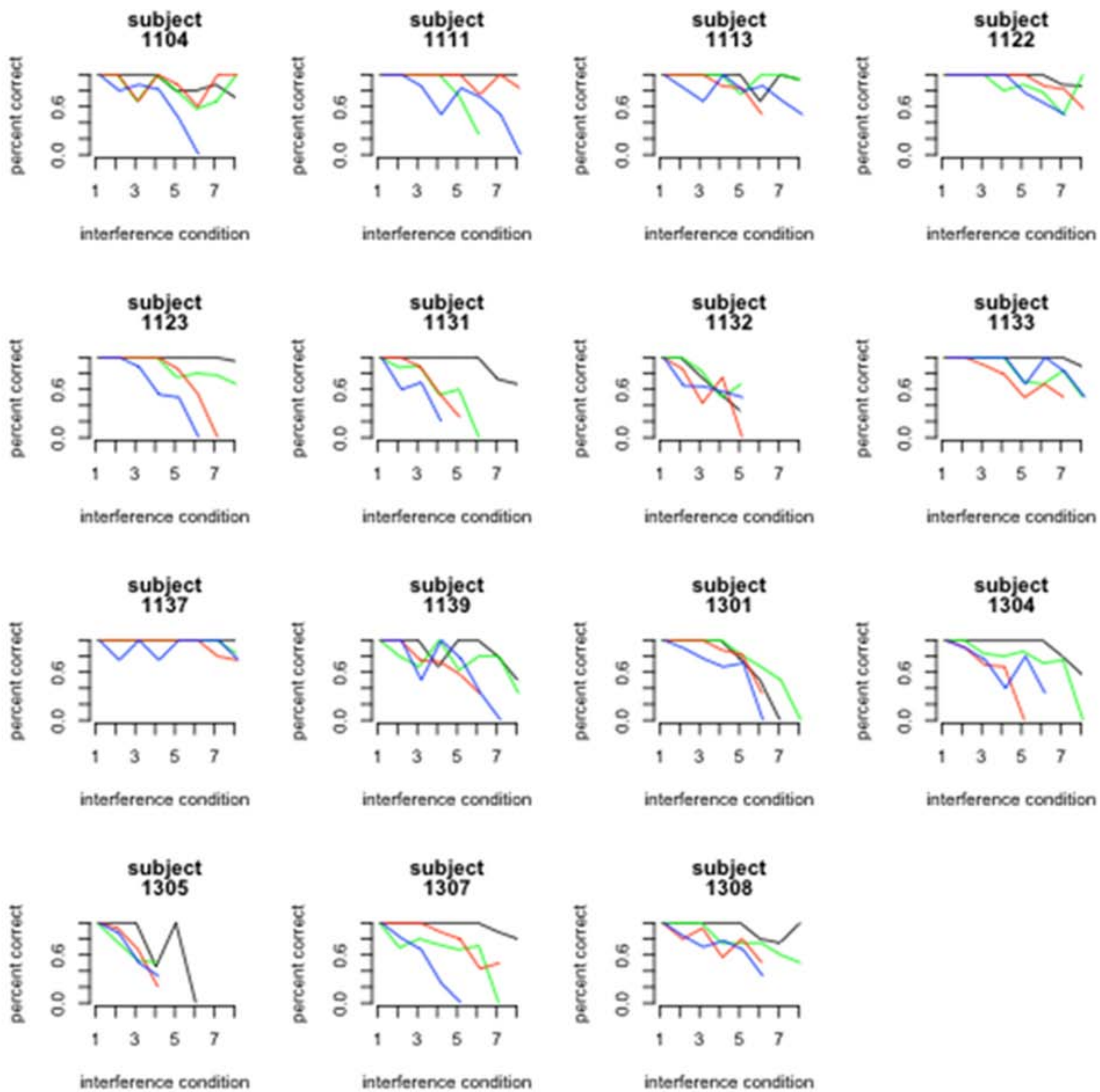


percent correct
1.0
0.8
0.6
0.4
0.2
0.0
none
verbal
motor
both
1
2
3
4
5
6
7
8
interference condition

# Another subject



# All the subjects



# An individual's data

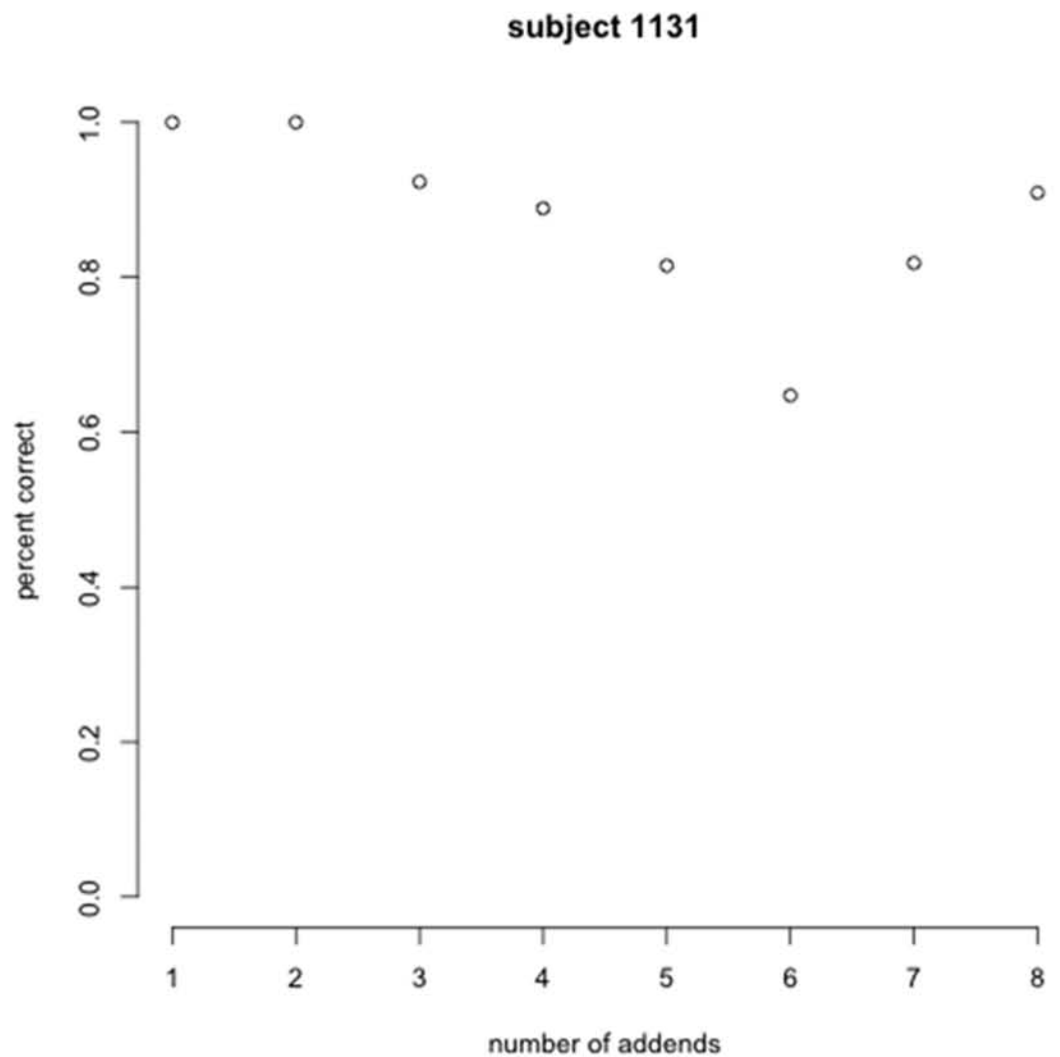
| trial | num | corr | addends | cond |
|---|---|---|---|---|
| 161 | 1 | 1 | 1 | none |
| 162 | 2 | 1 | 1 | none |
| 163 | 3 | 1 | 2 | none |
| 164 | 4 | 1 | 2 | none |
| 165 | 5 | 1 | 3 | none |
| 166 | 6 | 1 | 3 | none |
| | ... | ... | ... | |
| 294 | 32 | 1 | 5 | both |
| 295 | 33 | 0 | 5 | both |
| 296 | 34 | 0 | 4 | both |
| 297 | 35 | 1 | 3 | both |

# How do we model an individual?

Linear model of their performance looks great, right?

```
lm(formula =corr~ addends)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.94788    0.07759  12.217   <2e-16 ***
sub.addends -0.01654    0.01370  -1.207     0.230
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
```

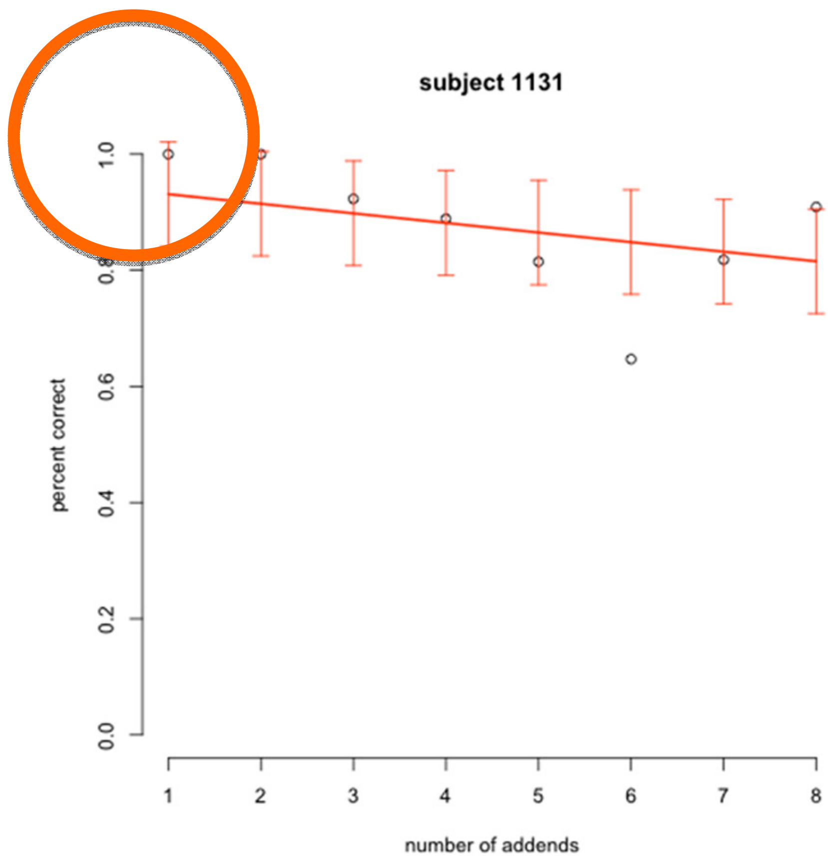
# Oops



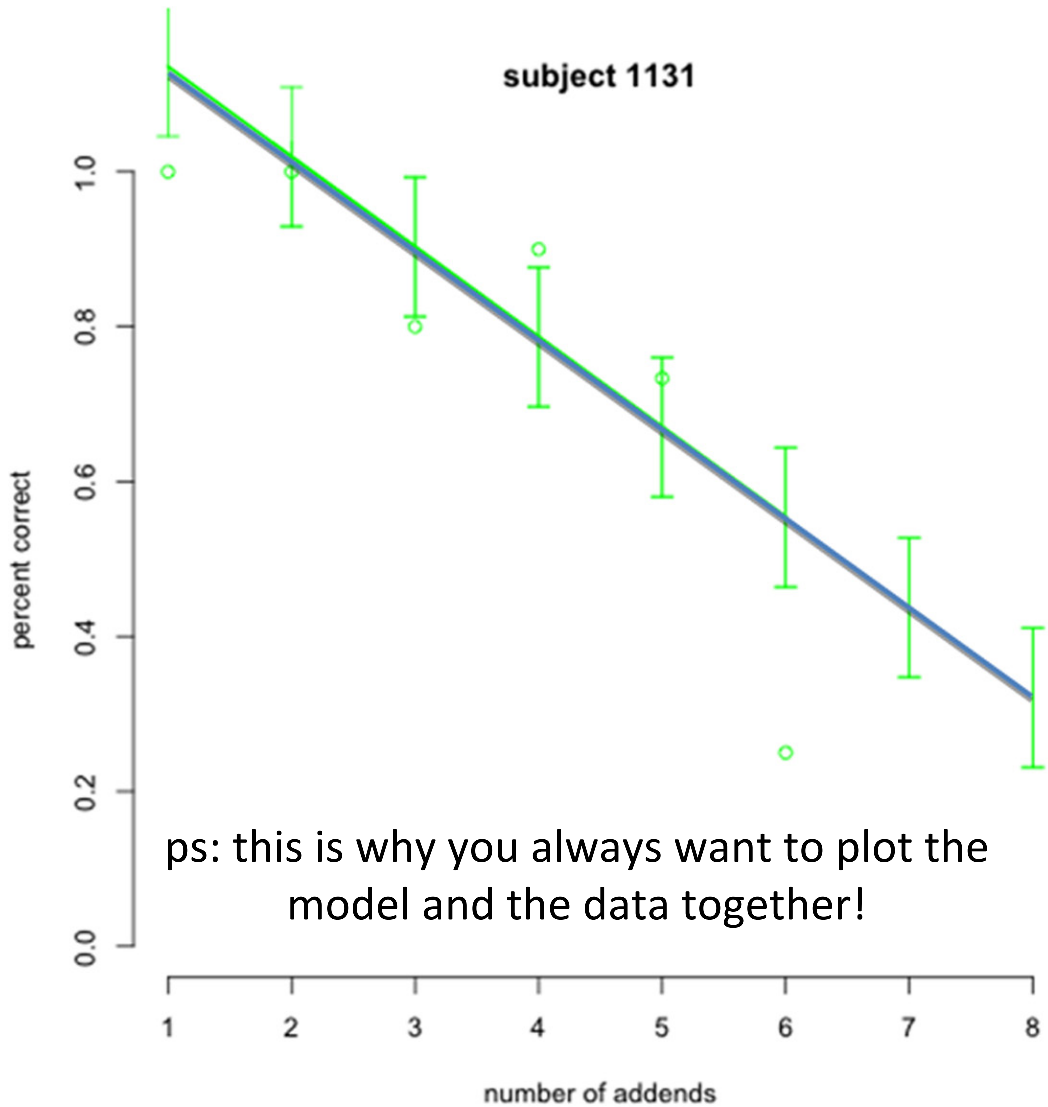
subject 1131
percent correct
0.0
0.2
0.4
0.6
0.8
1.0
1
2
3
4
5
6
7
8
number of addends

# Oops

**standard error shouldn't extend outside the bounds of the measure!**



# Oops

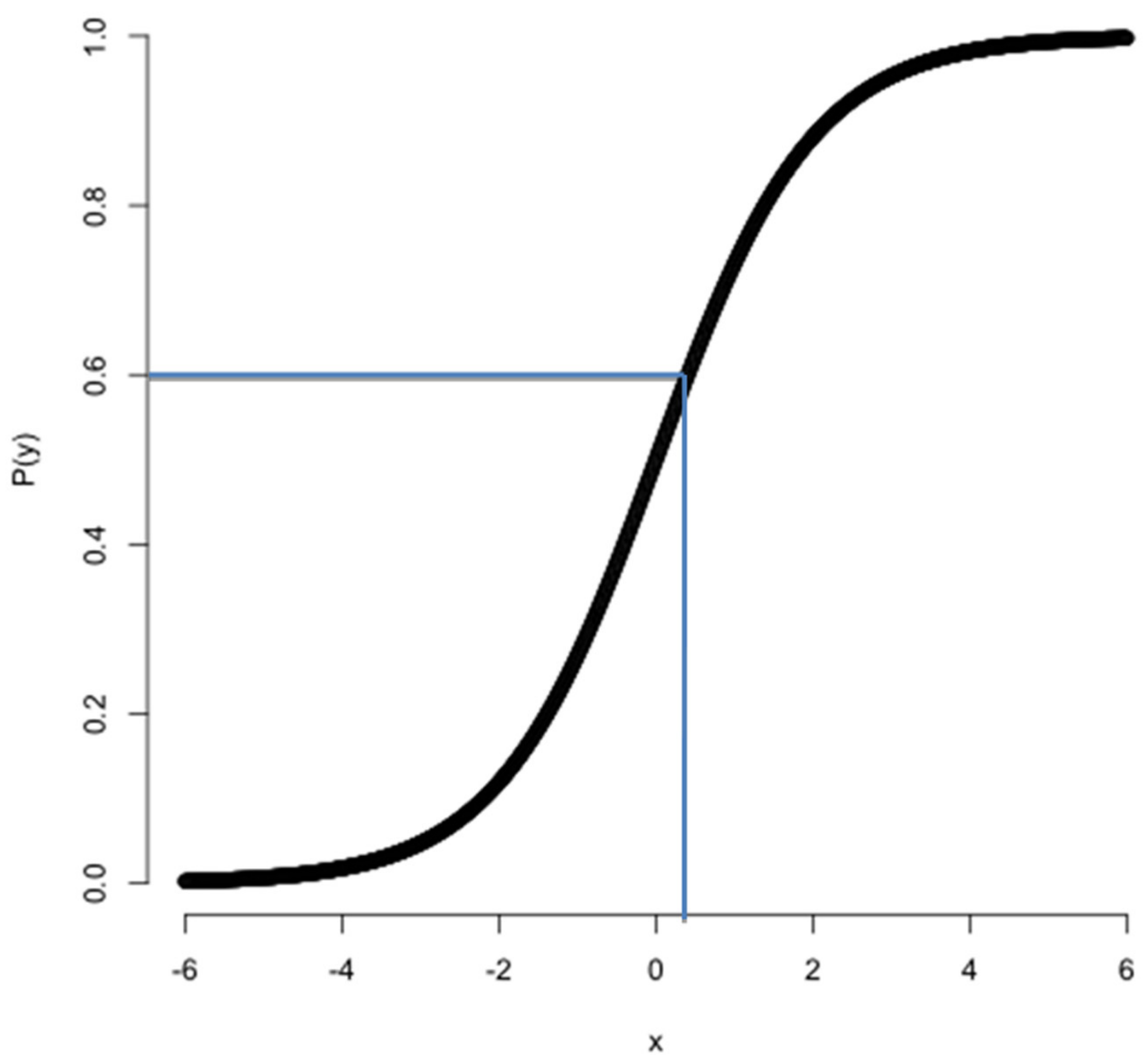



ps: this is why you always want to plot the model and the data together!

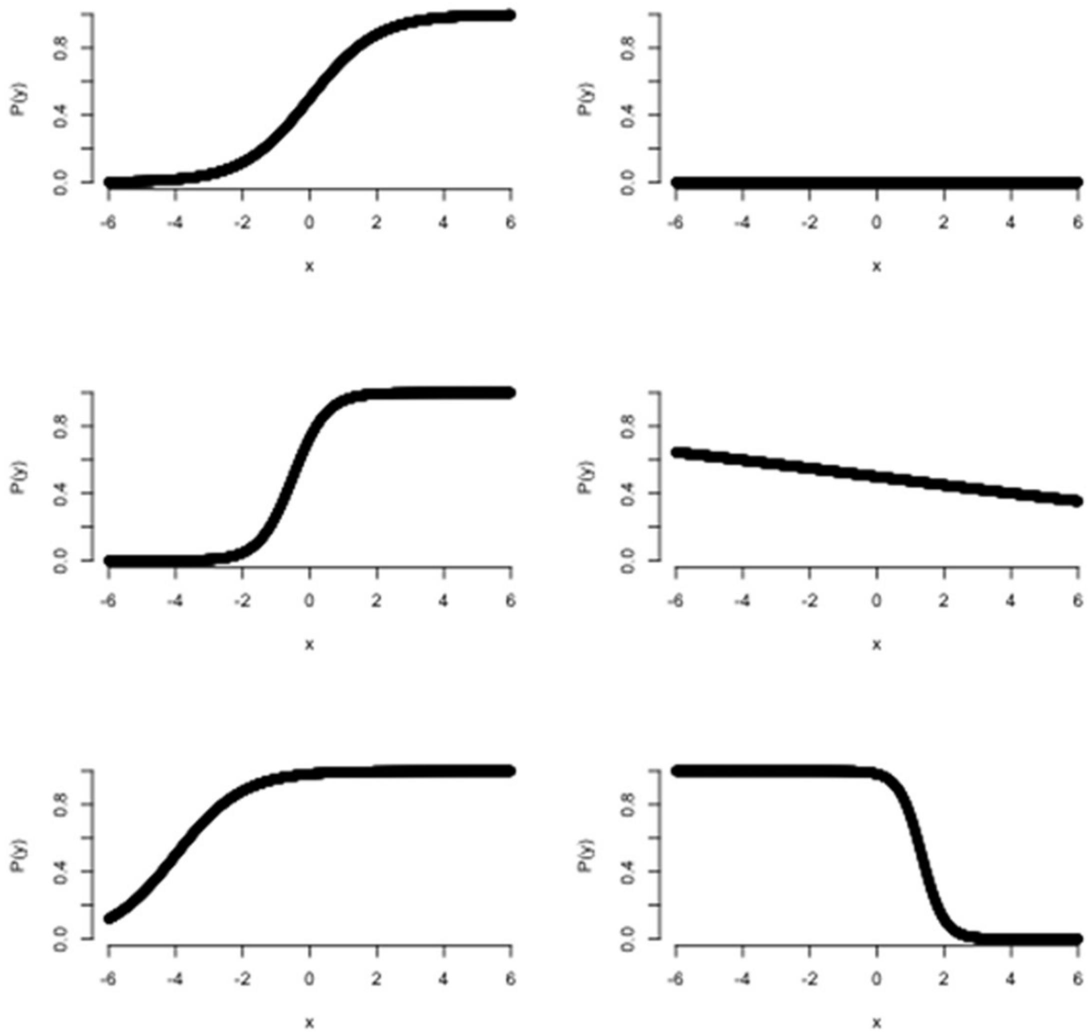
# Introducing: logistic regression

- In an ordinary linear model:
  - y = a + bx
  - Just add stuff up—y is (-Inf,Inf)
- In a logistic regression:
  - P(correct) = $\text{logit}^{-1}(bx)$, where $\text{logit}^{-1}(z) = \frac{1}{1+e^{-z}}$
- What does this do?
  - Turns real valued predictors into [0,1] valued probabilities! (our response format)
  - This is what a **generalized linear model** is: a way of linking a linear model to a particular response format

# The inverse logit function



# The varieties of logit experience



# Error rate reduction intuition

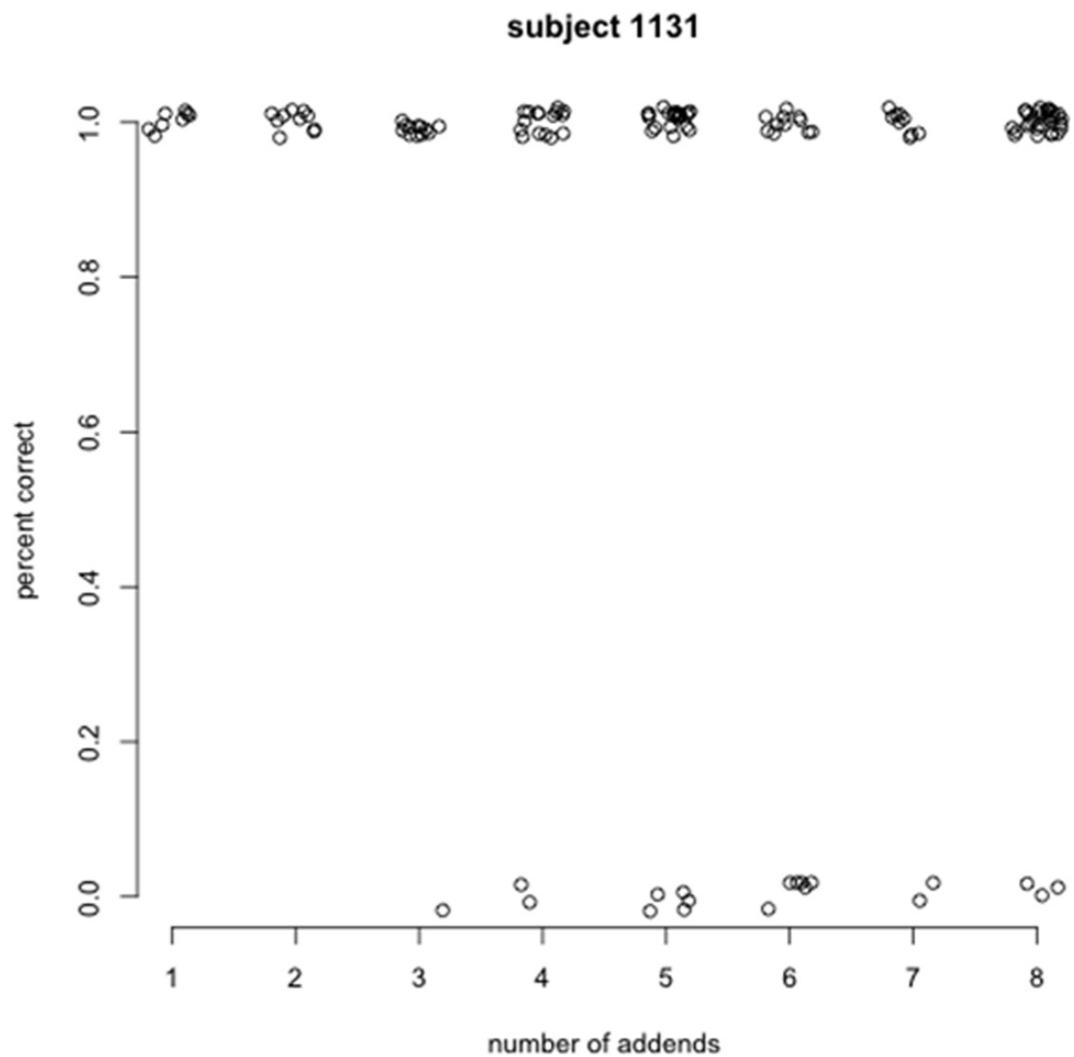
- Inverse logistic is curved
  - Difference in y corresponding to difference in x is not constant
  - Steep change happens in the middle of the curve
- From 50% to 60% performance is about as far as 90% to 93%
  - This is why those error bars were not right!
  - And this is why ANOVA/LM over percent correct is a big problem!

# Doing it right

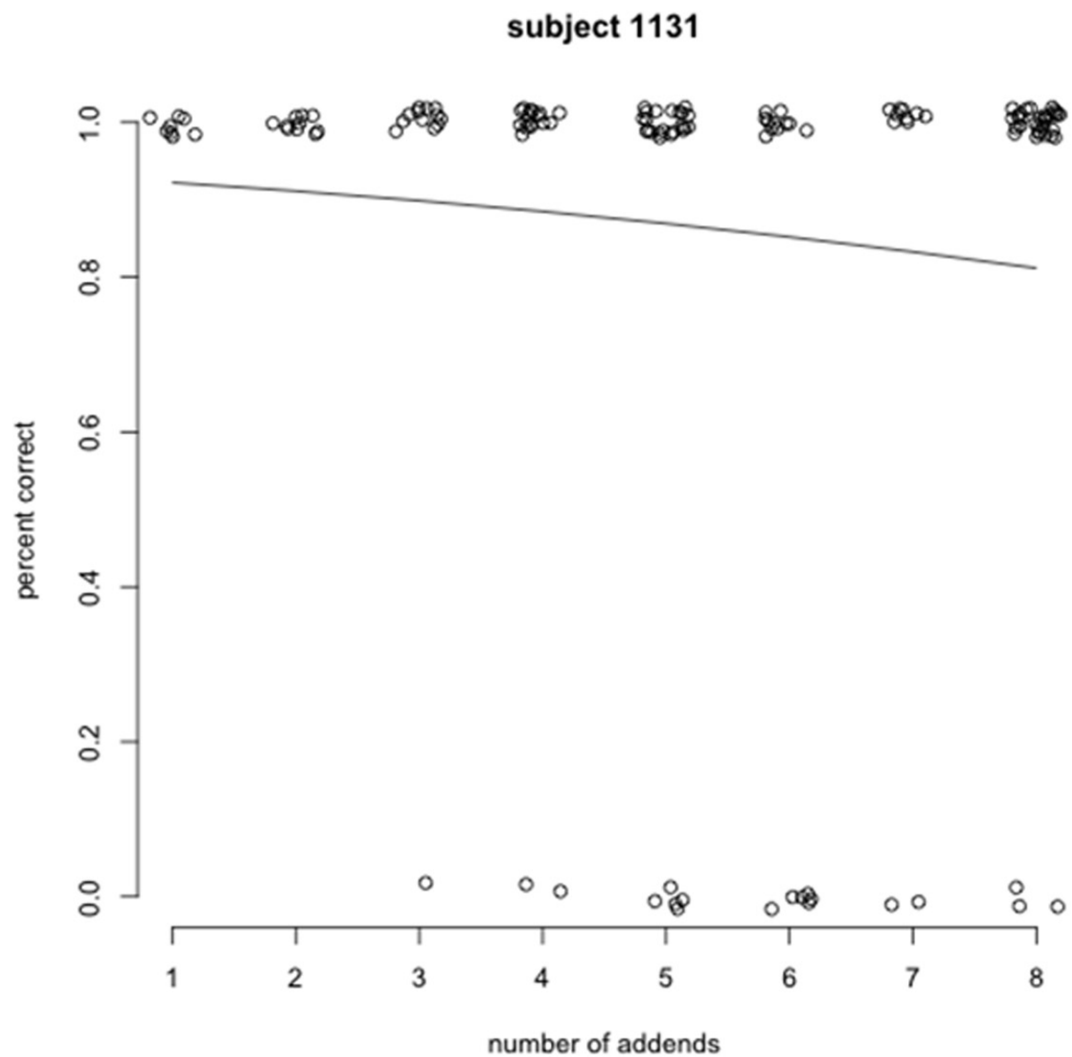
**glm(formula =corr~ addends, family = "binomial")**

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   2.6188     0.7322   3.577 0.000348 ***
addends      -0.1448     0.1207  -1.200 0.230231
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1
```

# What do the data actually look like?



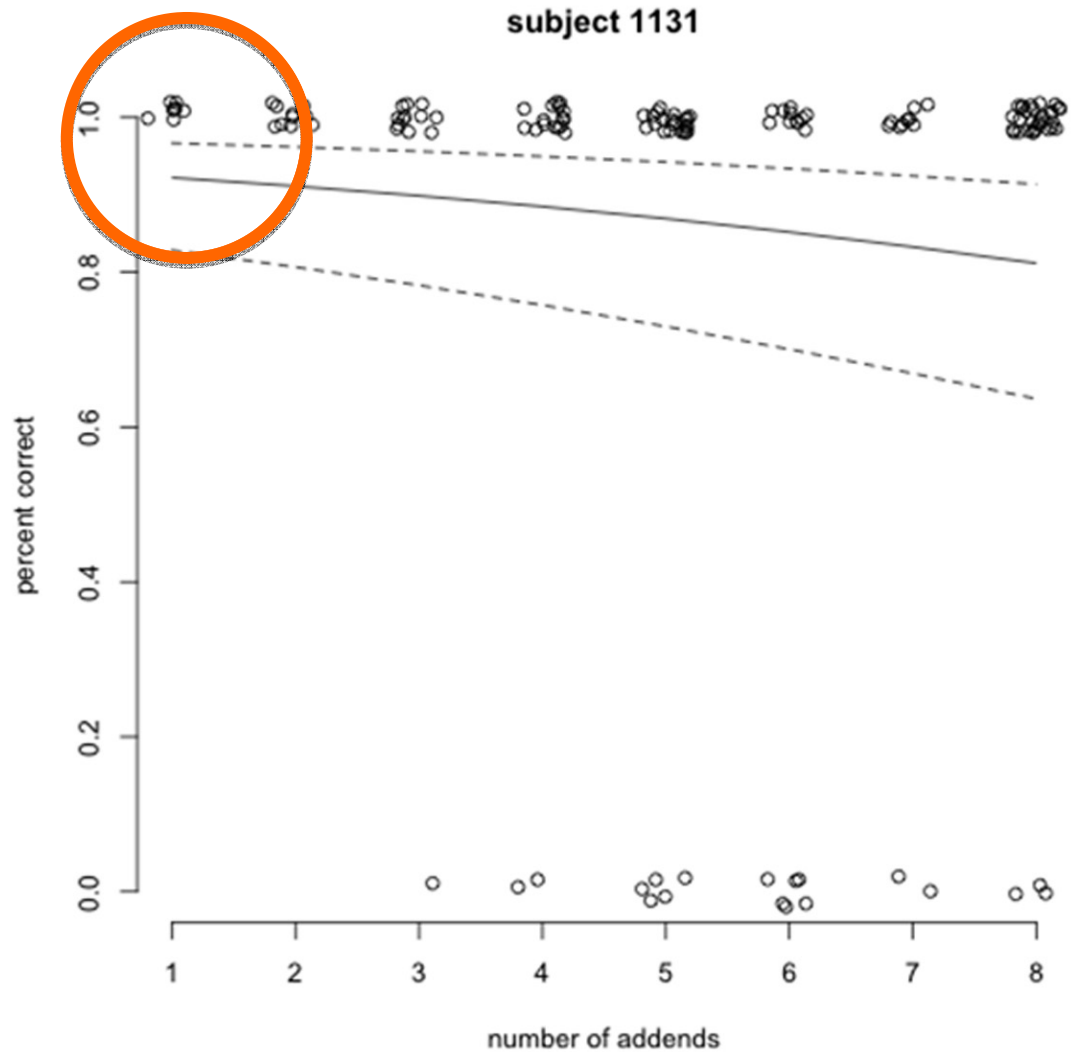
# Data + model



subject 1131
percent correct
0.0
0.2
0.4
0.6
0.8
1.0
number of addends
1
2
3
4
5
6
7
8

# Data + model + errors

**logit compresses with the scale**



# Aside: interpreting logistic coefficients

```
glm(formula =corr~ addends, family = "binomial")

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   2.6188     0.7322   3.577 0.000348 ***
addends      -0.1448     0.1207  -1.200 0.230231
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
```

- What does an estimate mean?
  - Well, different changes in probability at different points in the scale

# Which logistic regression?

**glm(formula =corr~cond, family = "binomial")**

Coefficients:

| | Estimate | Std. Error | z value | Pr(>\|z\|) | |
|---|---|---|---|---|---|
| (Intercept) | 1.0609 | 0.3867 | 2.743 | 0.00609 | ** |
| cond.motor | 1.2417 | 0.7185 | 1.728 | 0.08395 | . |
| cond.none | 18.5052 | 1844.2980 | 0.010 | 0.99199 | |
| cond.verbal | 0.3254 | 0.5728 | 0.568 | 0.56998 | |

---

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1

# Which logistic regression?

**glm(formula = corr ~ cond - 1, family = "binomial")**

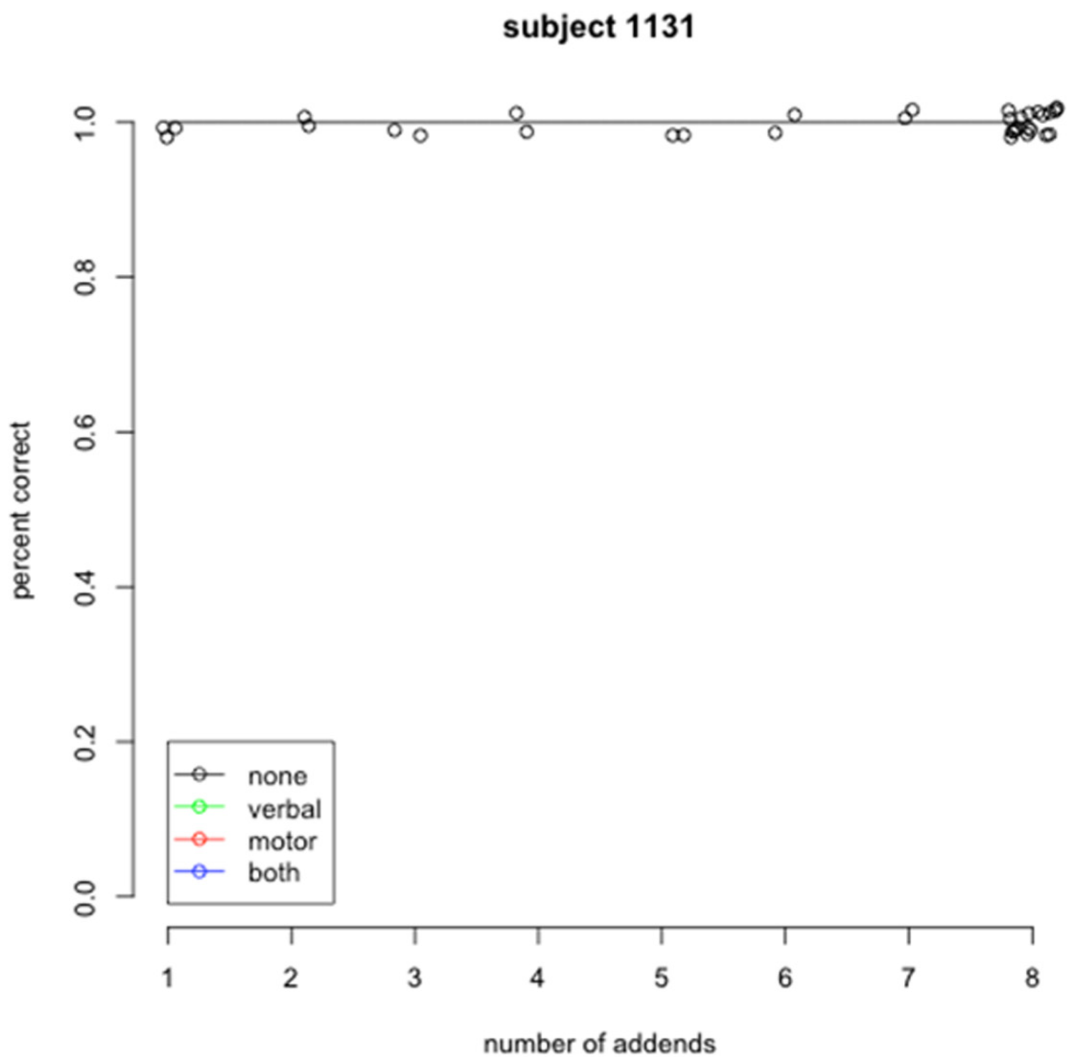
```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
cond.both     1.0609     0.3867   2.743 0.006087 **
cond.motor    2.3026     0.6055   3.803 0.000143 ***
cond.none    19.5661  1844.2980   0.011 0.991535
cond.verbal   1.3863     0.4226   3.281 0.001036 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1
```

# Which logistic regression?

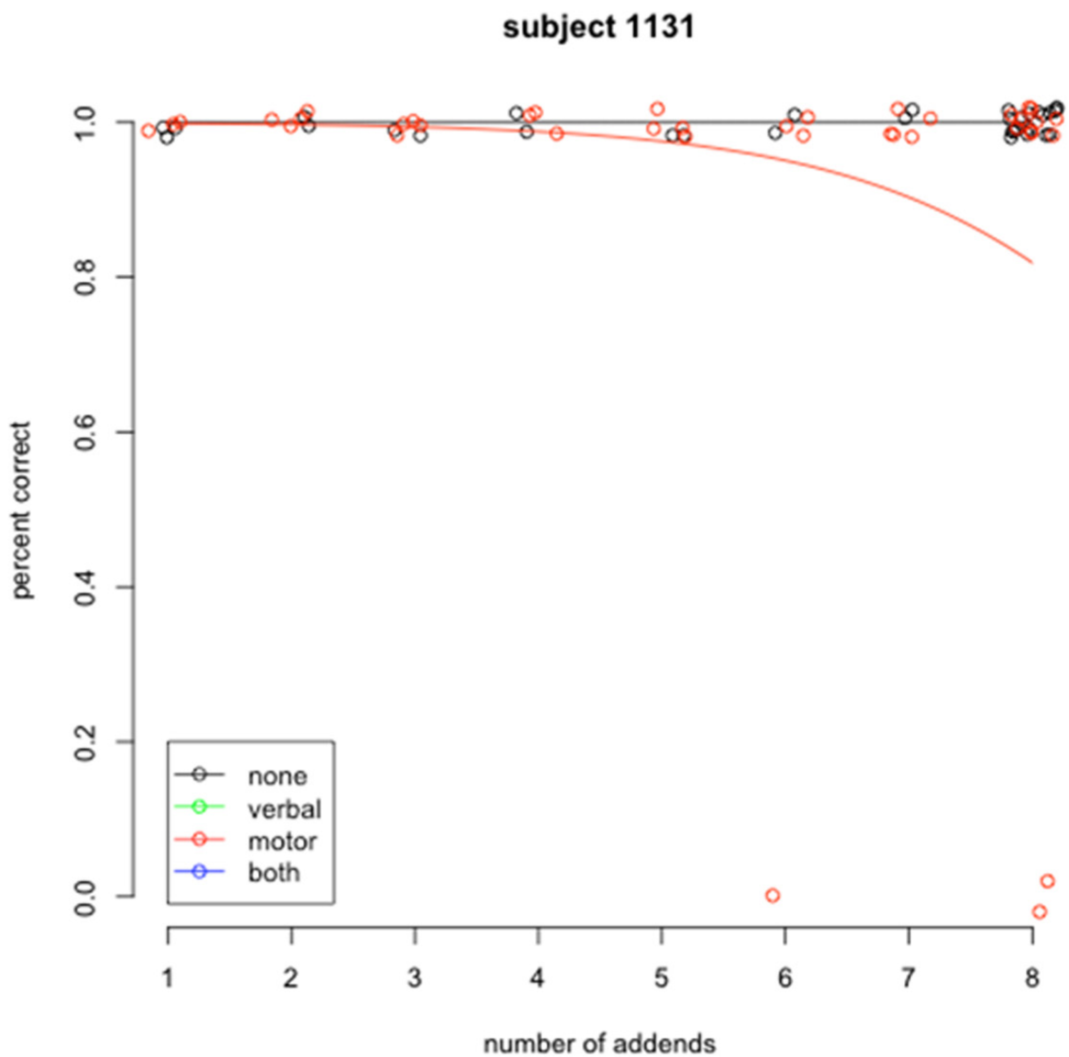
**glm(formula = sub.corr ~ sub.addends + sub.cond - 1, family = "binomial")**

```
Coefficients:
              Estimate Std. Error z value Pr(>|z|)
addends        -0.7267     0.2435  -2.985 0.002839 **
cond.both4.6731     1.3872   3.369 0.000755 ***
cond.motor7.3203     1.9279   3.797 0.000146 ***
cond.none24.7699  1695.5890   0.015 0.988345
cond.verbal      4.7366     1.2569   3.768 0.000164 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

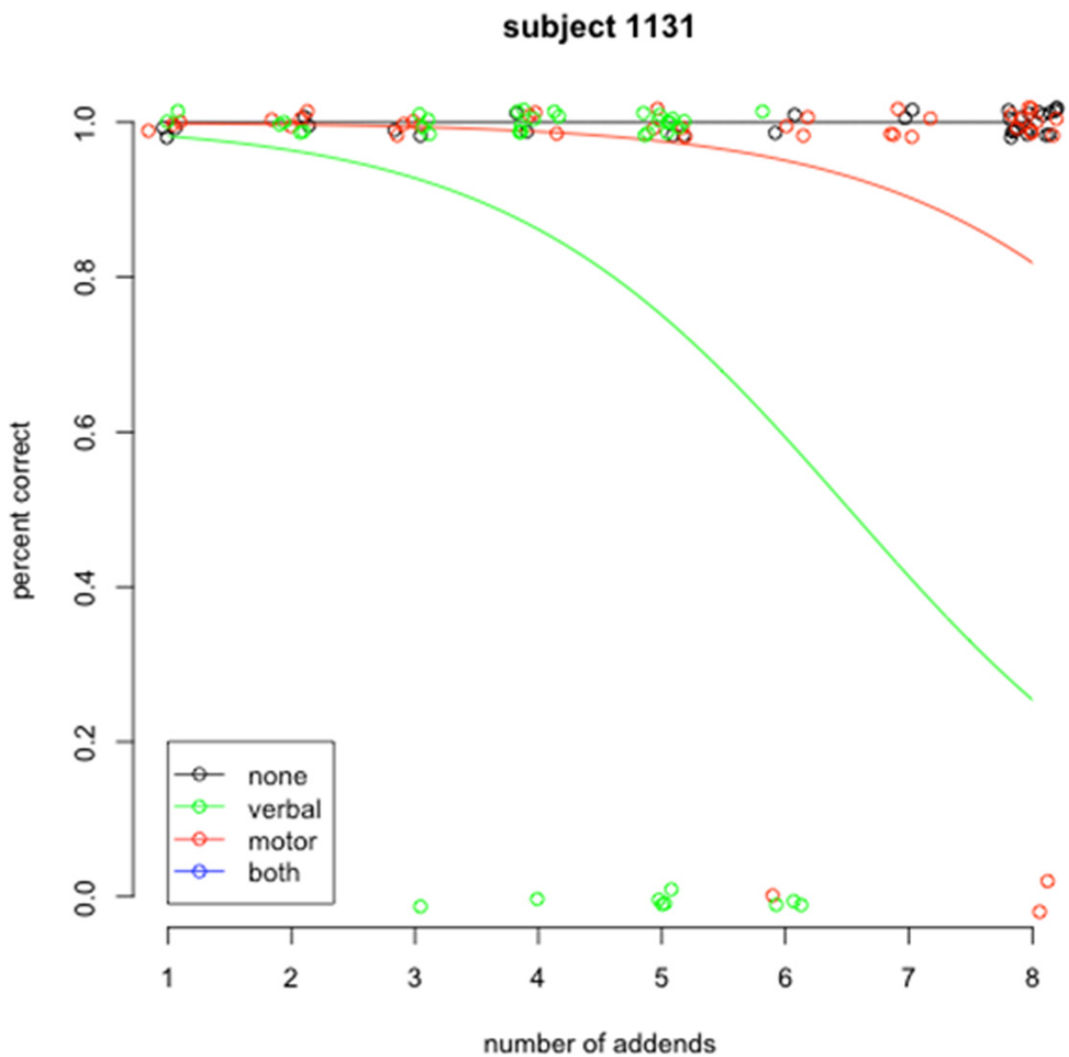
# Now plot model + data



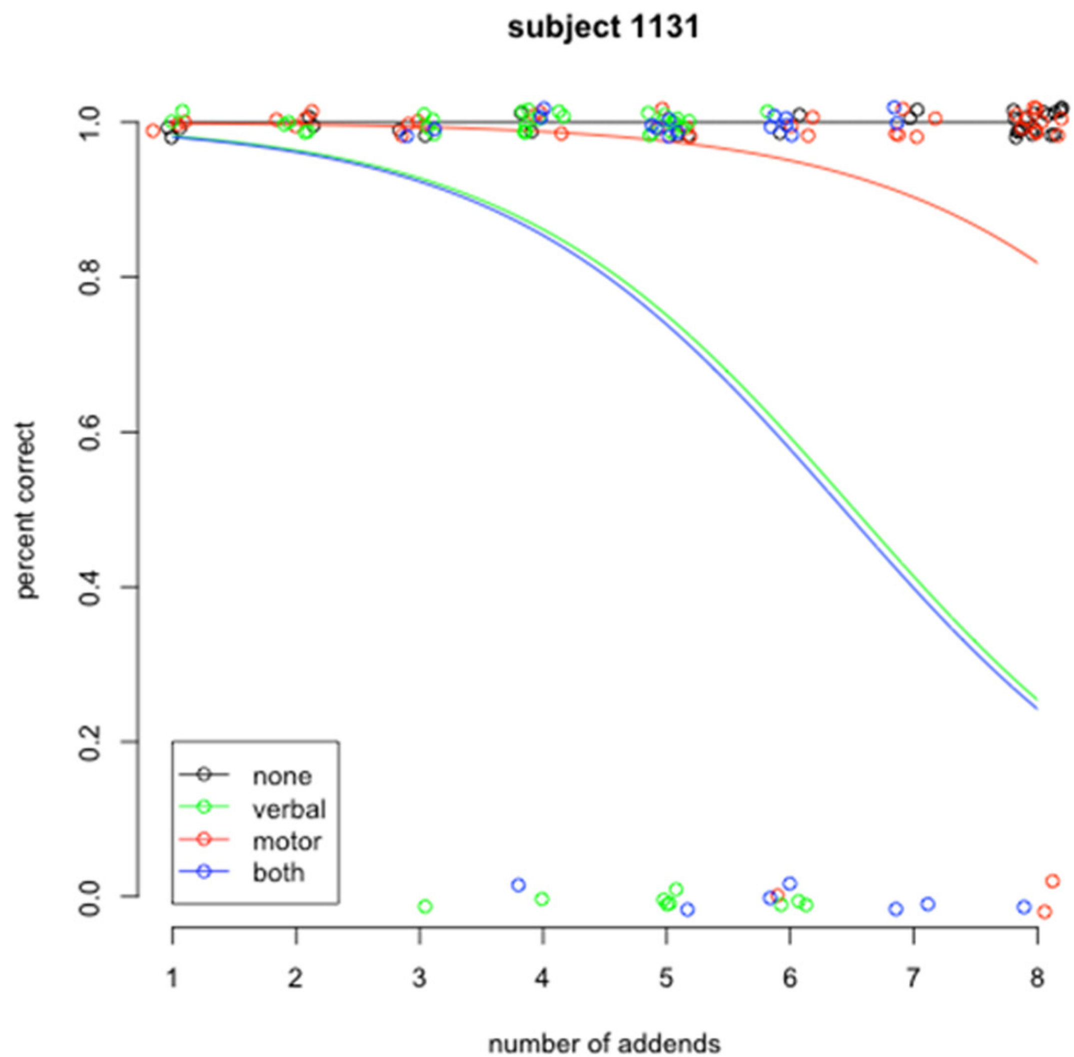
# Now plot model + data



# Now plot model + data



# Now plot model + data



# Which logistic regression?

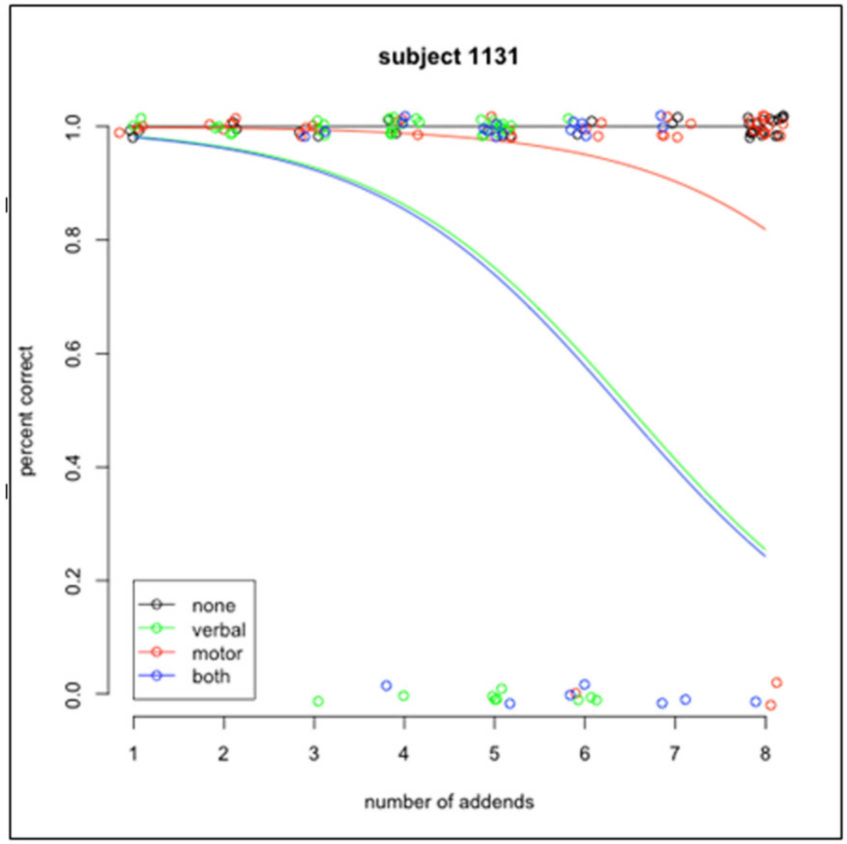
**glm(formula = sub.corr ~ sub.addends + sub.cond - 1, family = "binomial")**

Coefficients:

| | Estimate | Std. Error | z value | Pr(>\|z\|) | |
|---|---|---|---|---|---|
| addends | -0.7267 | 0.2435 | -2.985 | 0.002839 | ** |
| cond.both | 4.6731 | 1.3872 | 3.369 | 0.000755 | *** |
| cond.motor | 7.3203 | 1.9279 | 3.797 | 0.000146 | *** |
| cond.none | 24.7699 | 1695.5890 | 0.015 | 0.988345 | |
| cond.verbal | 4.7366 | 1.2569 | 3.768 | 0.000164 | *** |

---

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '



# Analyzing the whole dataset

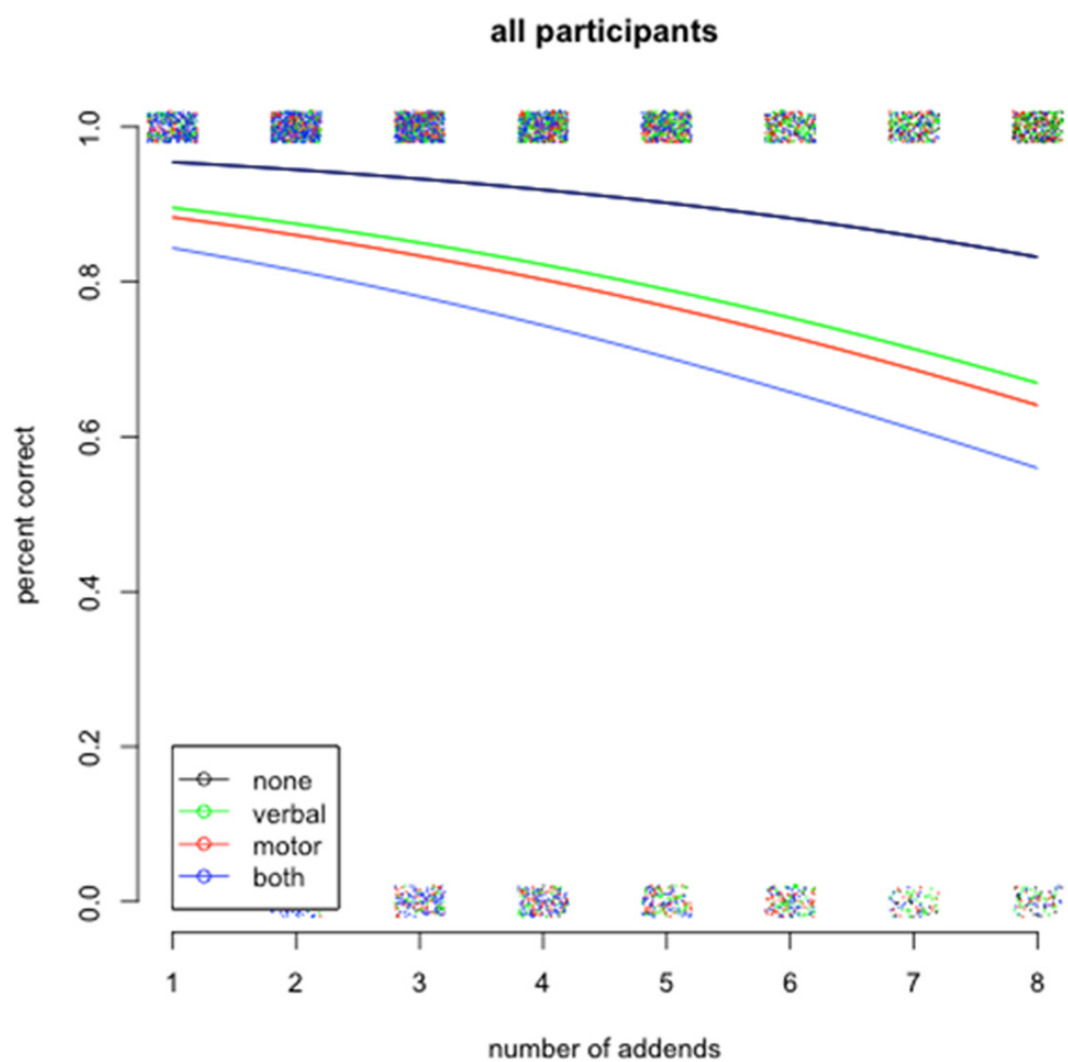
- We can do a logistic model for all the data!
  - Averaging across subjects
  - "Just gets rid of noise"?

**glm(formula = correct ~ addends + cond - 1)**

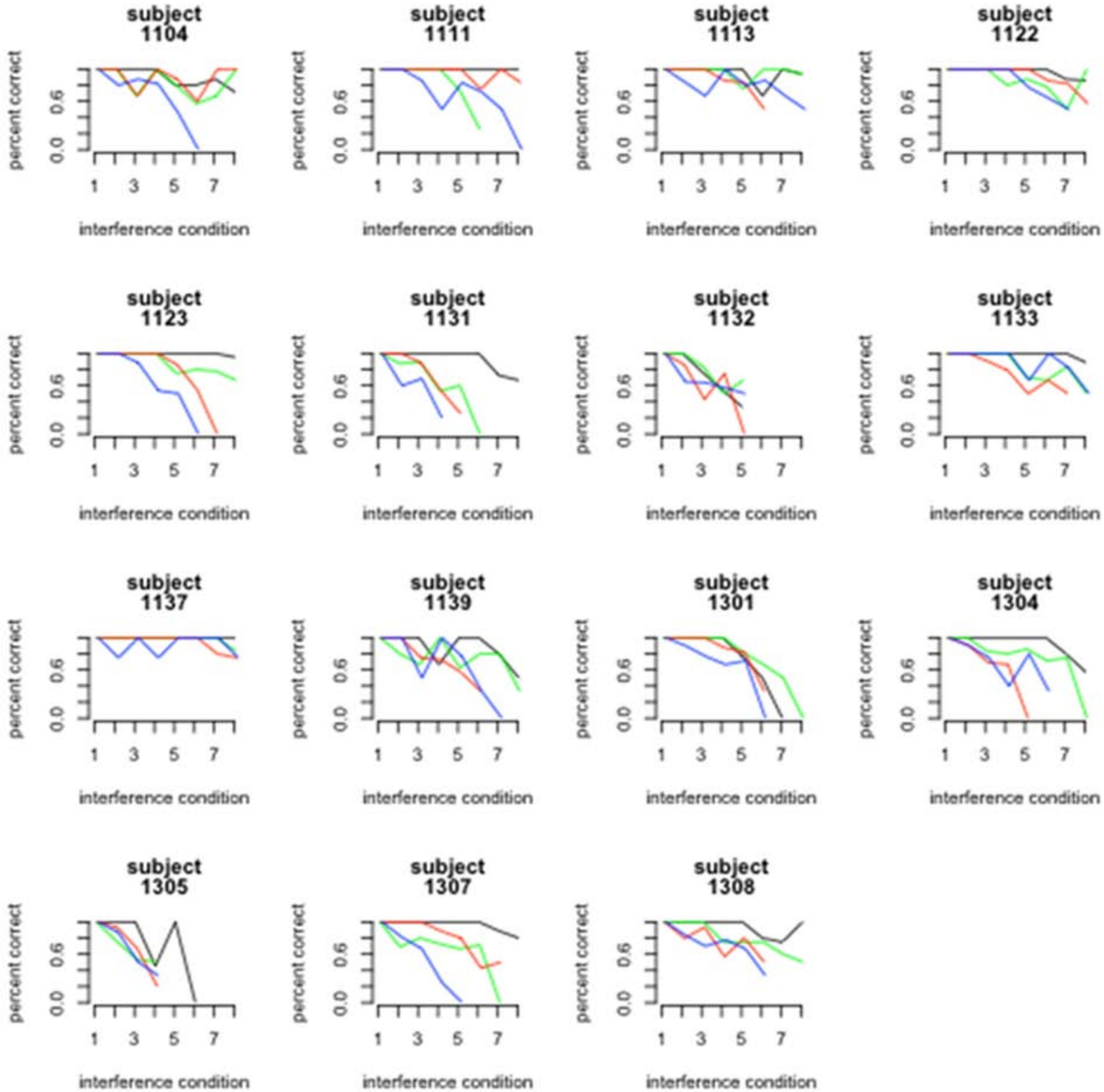
Coefficients:

| | Estimate | Std. Error | z value | Pr(>\|z\|) | |
|---|---|---|---|---|---|
| addends | -0.20663 | 0.02682 | -7.706 | 1.30e-14 | *** |
| cond.both | 1.89223 | 0.13839 | 13.673 | < 2e-16 | *** |
| cond.motor | 2.23117 | 0.15908 | 14.025 | < 2e-16 | *** |
| cond.none | 3.25219 | 0.21618 | 15.044 | < 2e-16 | *** |
| cond.verbal | 2.35800 | 0.16720 | 14.103 | < 2e-16 | *** |

# Plotting everything

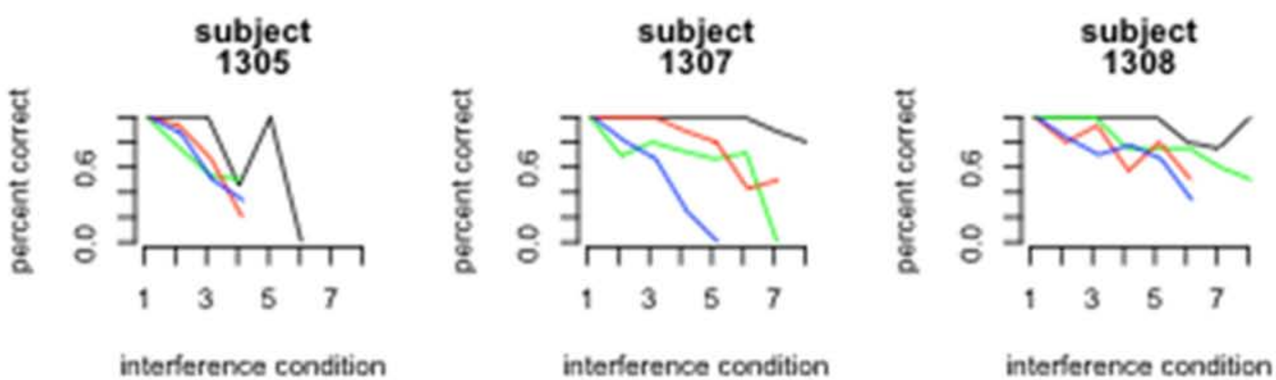


# All the subjects



# What's going on?

- Every participant didn't do every trial
  - participants only did trials they did (relatively) well on
  - good participants contributed all the trials for the higher numbers of addends



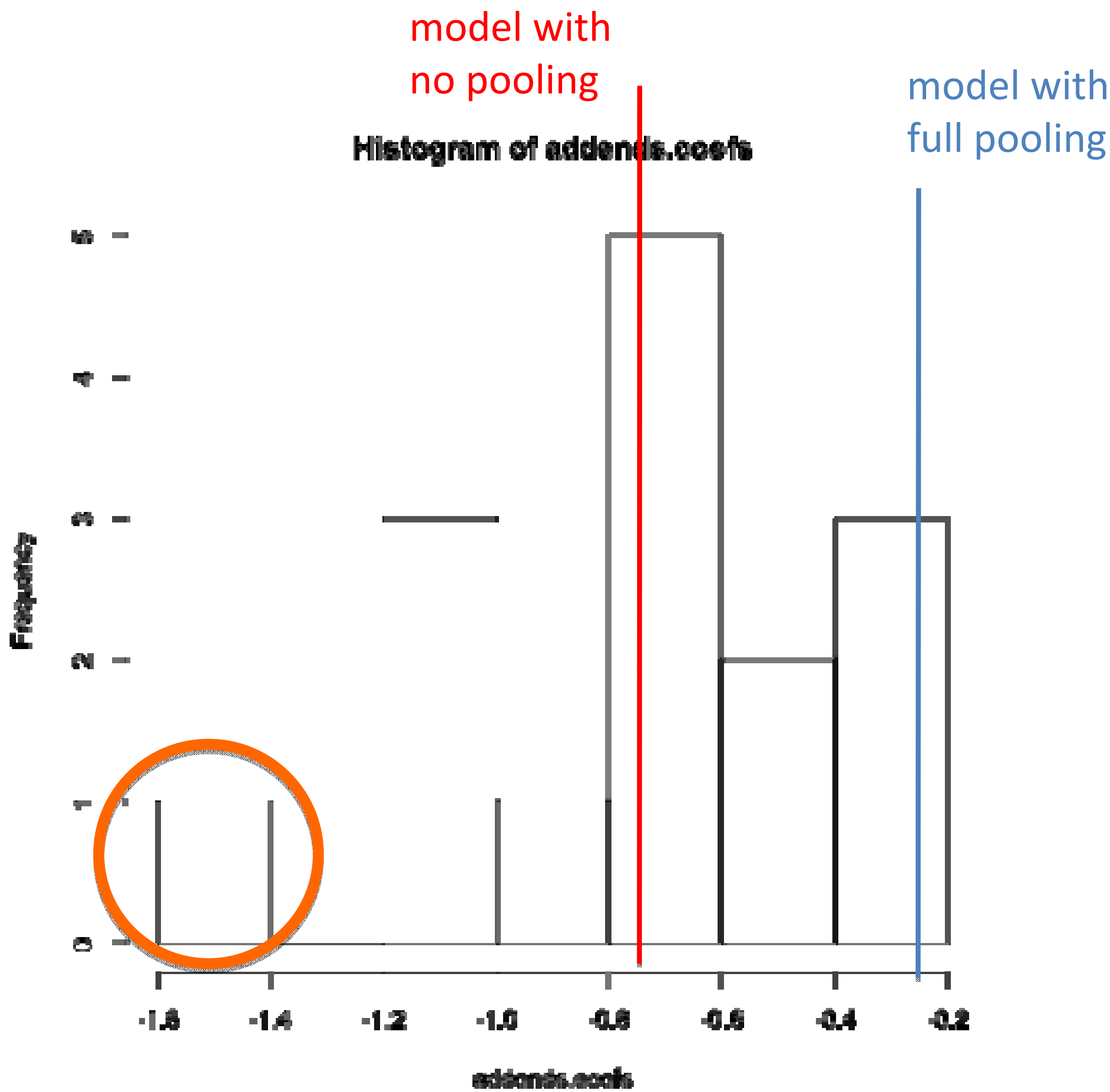
# Multilevel linear modeling

- You have data at two levels
  - Group level information about condition
  - Subject level information about identities
- You want to aggregate information
- There are three options
  1. Full pooling: throw out info about identities
  2. No pooling: analyze each subject separately
  3. Partial pooling: try to factor out unique contribution of subject identities

# No pooling

- Estimate a separate GLM for each subject
  - Then make inferences about the robustness of coefficients across GLM
  - But if you have sparse or noisy information for a participant, you can't use group data to correct
- "Whereas complete pooling ignores variation between individuals, the no-pooling analysis overstates it. To put it another way, the no-pooling analysis overfits the data within each individual." (Gelman& Hill, 2006)

# Addend coefficients




**Were they that bad?**

# Multilevel models for partial pooling

- The solution: fit a model which assigns some variation to individual participants and some to group level coefficients
- standard LM: $y = a + bx$
- simple multilevel LM: $y = a_j + bx + \ldots$
  - different intercept for each participant
  - but same slope
- more complex model: $y = a_j + b_j x + \ldots$
- we won't talk about *how* to fit any of these

# Mixed logistic regression

```
Generalized linear mixed model fit by the Laplace approximation
Formula: correct ~ addends + cond - 1 + (1 | subnum)

Random effects:
 Groups Name        Variance Std.Dev.
subnum (Intercept) 0.62641  0.79146
Number of obs: 2410, groups: subnum, 15

Fixed effects:
          Estimate Std. Error z value Pr(>|z|)
addends    -0.42775    0.03611  -11.84   <2e-16 ***
condboth    2.84159    0.26983   10.53   <2e-16 ***
condmotor   3.30181    0.28705   11.50   <2e-16 ***
condnone    4.76725    0.34824   13.69   <2e-16 ***
condverbal  3.54185    0.29771   11.90   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

**Here's the varying intercept term**

**methods note**: this is using R with the lme4 package, also new version of matlab can do this

# Mixed logistic regression

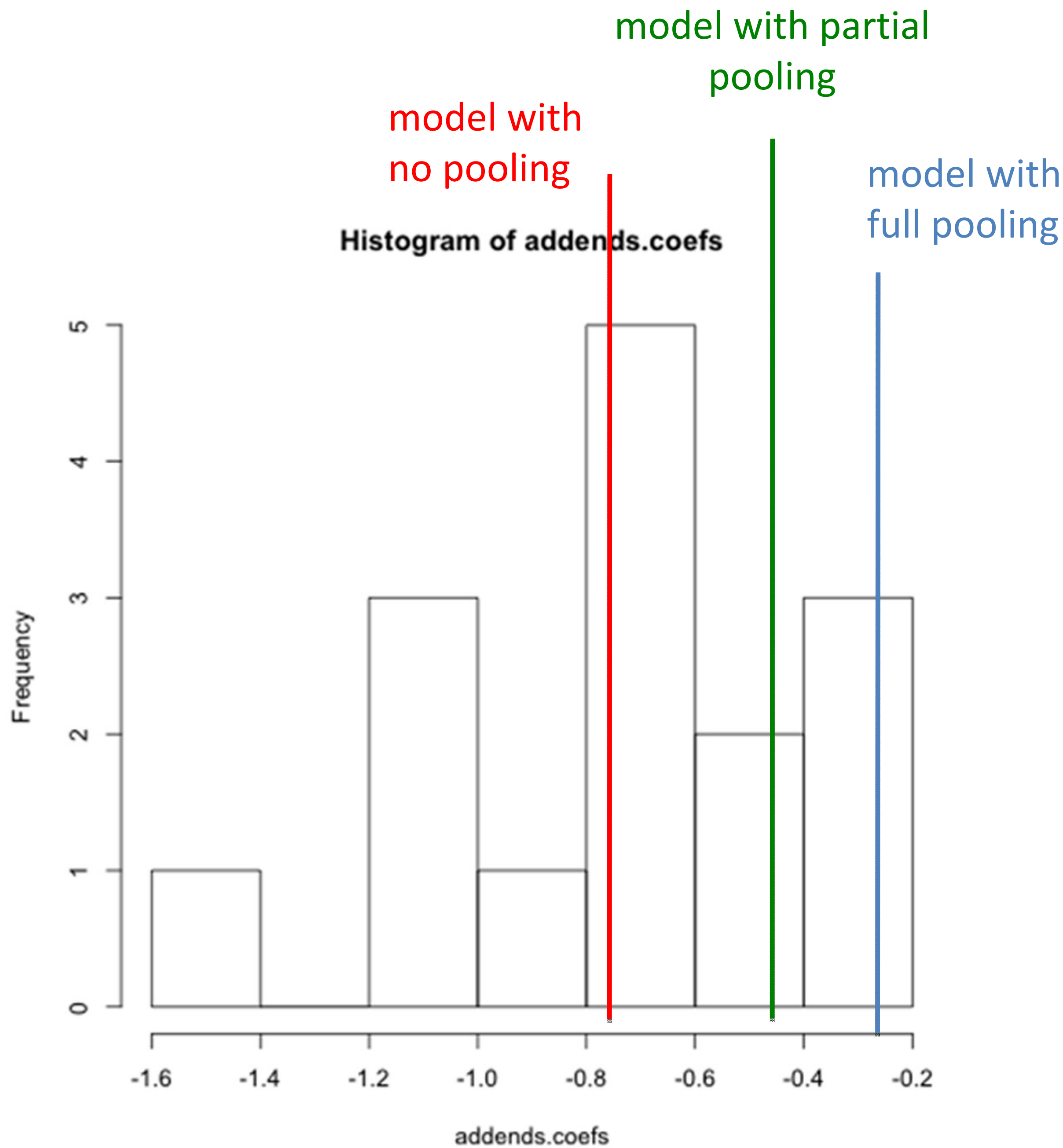
**group level predictors**

| | |
|---|---|
| addends | -0.4277491 |
| cond.both | 2.8415901 |
| cond.motor | 3.3018099 |
| cond.none | 4.7672475 |
| cond.verbal | 3.5418503 |

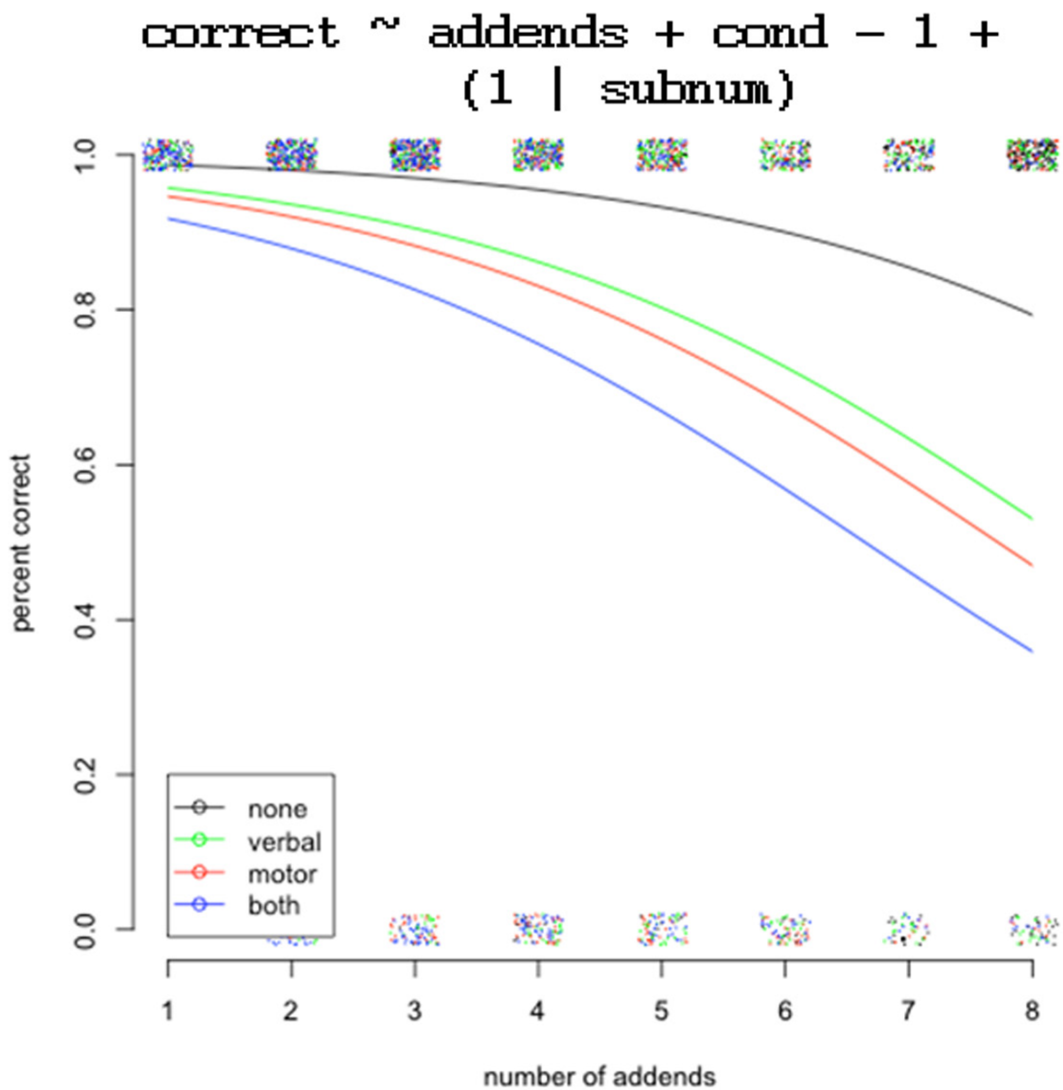
**subject-level predictors**

| | |
|---|---|
| 1104 | 0.2716444 |
| 1111 | 0.6886913 |
| 1113 | 0.8022382 |
| 1122 | 0.6276071 |
| 1123 | 0.2784564 |
| 1131 | -0.8091596 |
| 1132 | -1.1941759 |
| 1133 | 0.4858048 |
| 1137 | 1.5881314 |
| 1139 | -0.2907816 |
| 1301 | -0.3750832 |
| 1304 | -0.4040512 |
| 1305 | -1.2366471 |
| 1307 | -0.4758867 |
| 1308 | -0.1190724 |

# Addend coefficients

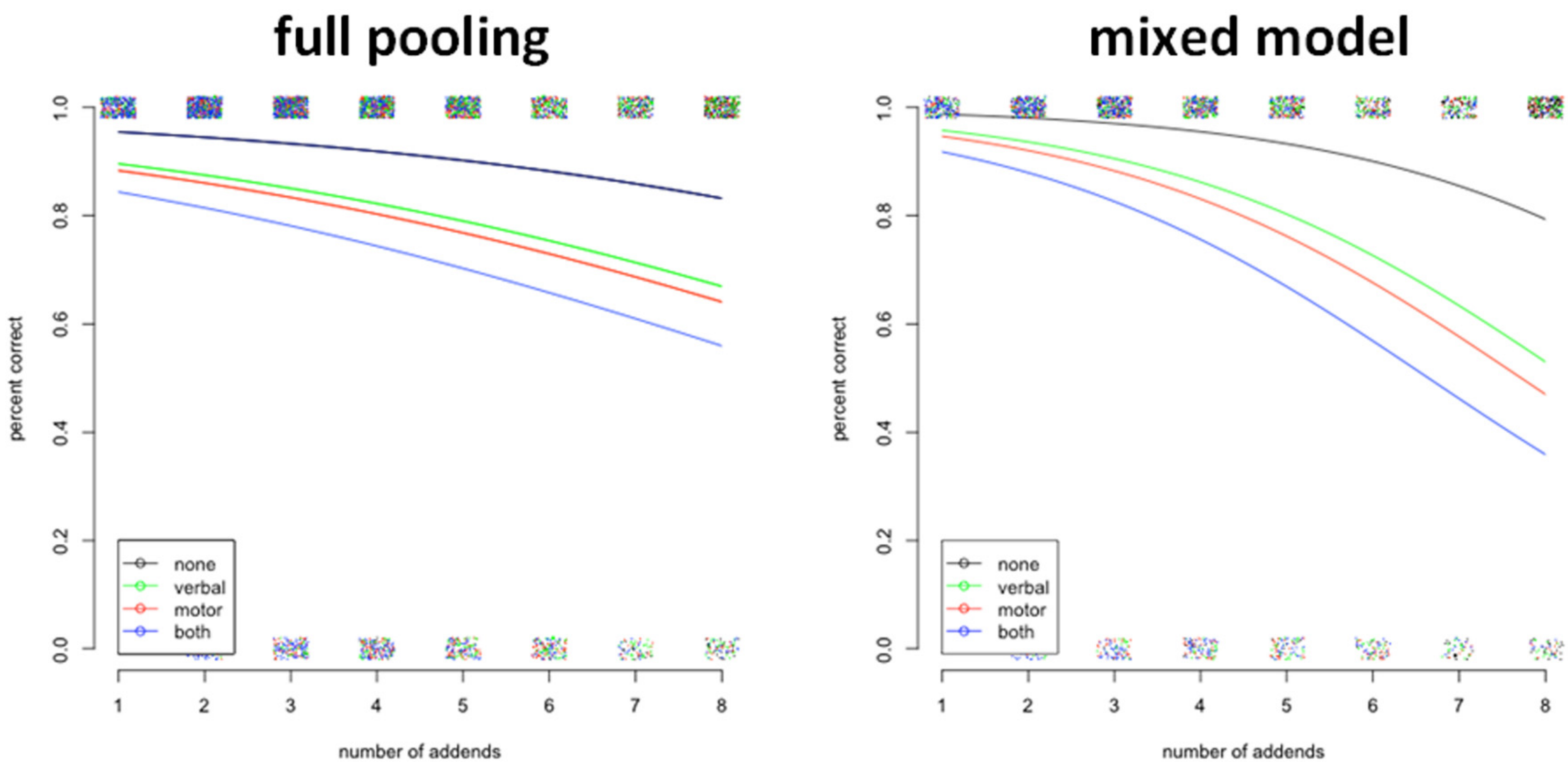


# Mixed model

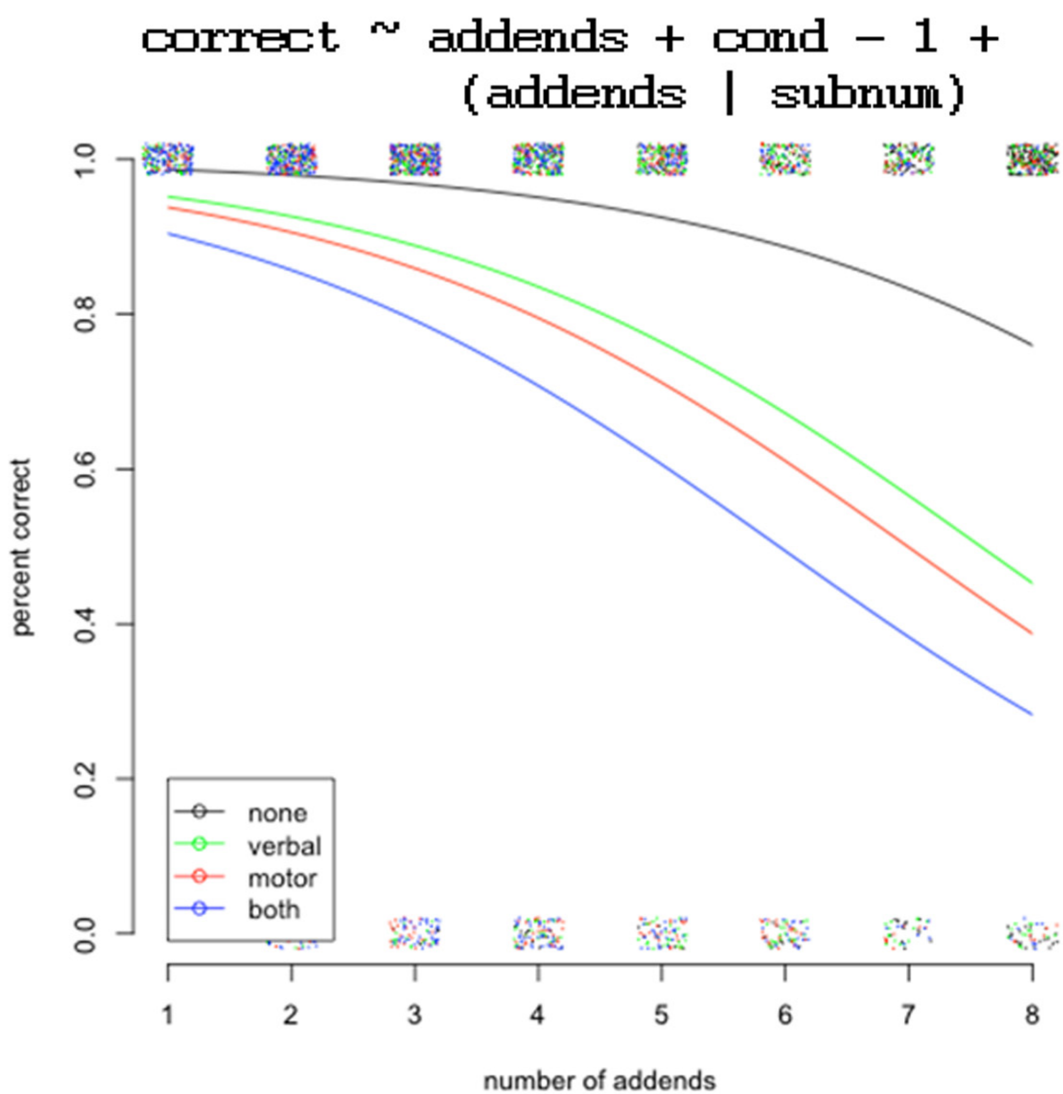


correct ~ addends + cond − 1 +
(1 | subnum)
percent correct
0.0
0.2
0.4
0.6
0.8
1.0
none
verbal
motor
both
1
2
3
4
5
6
7
8
number of addends

# Side by side

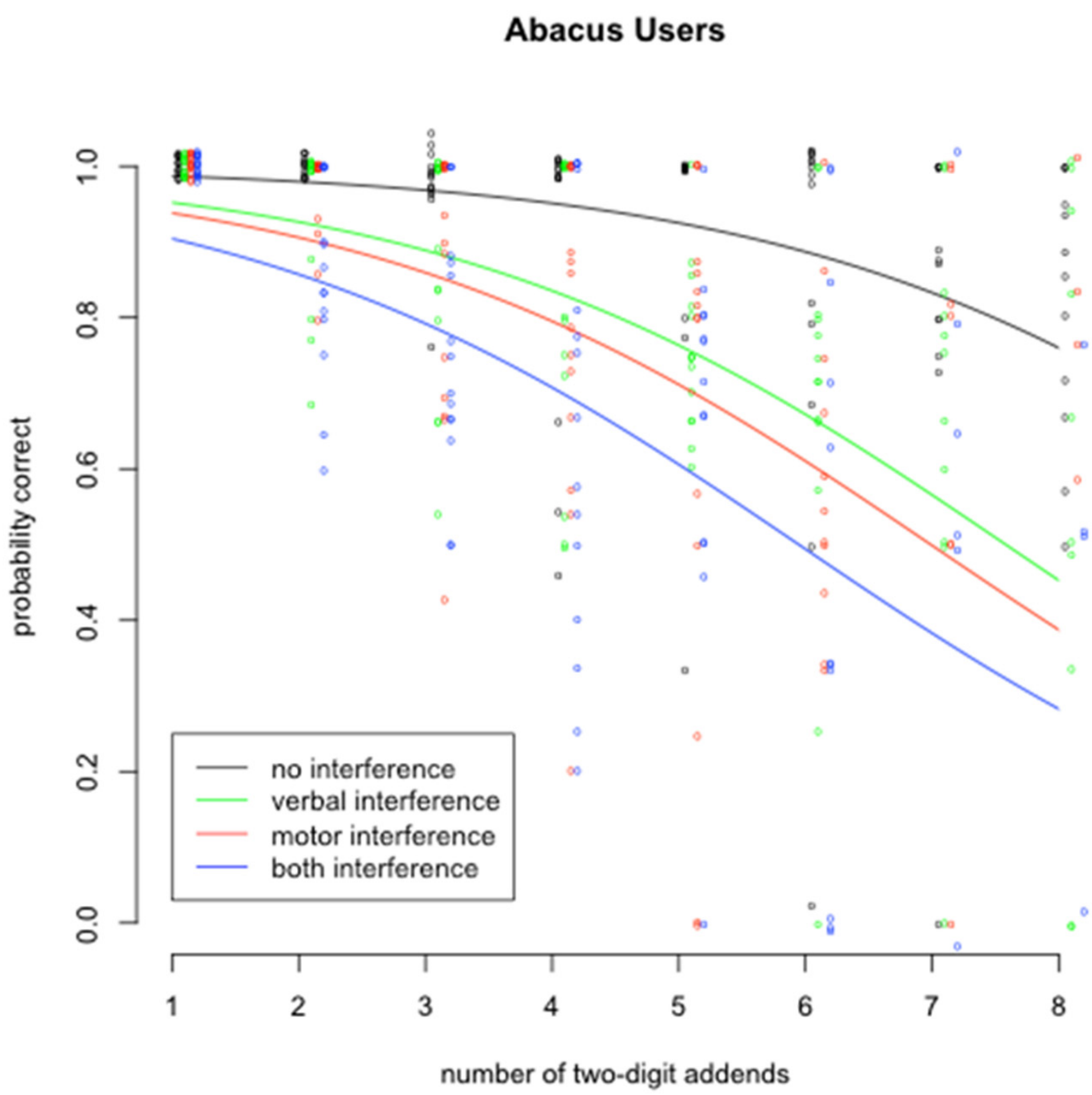

full pooling
mixed model
percent correct
number of addends
1.0
0.8
0.6
0.4
0.2
0.0
1
2
3
4
5
6
7
8
none
verbal
motor
both

# Mixed model



correct ~ addends + cond - 1 +
(addends | subnum)
percent correct
0.0
0.2
0.4
0.6
0.8
1.0
none
verbal
motor
both
1
2
3
4
5
6
7
8
number of addends

# The eventual visualization



# Generalizing from this example

- Not all studies have this dramatic problem
  - Part of the reason for the big change with the mixed model was the fact that not all subjects did all trials
- When do I choose a mixed model
  - When DON'T you want a mixed model?
  - If you've got logistic data, you don't want to use a regular LM over means
    - std. errors don't work out, e.g.
    - but full pooling is anti-conservative (violates independence)
  - So use the mixed logistic model

# CONCLUSIONS

# Summary

- The linear model is a model of data
  - Consider the interpretation of your model
  - Treat it as a model whose fit should be assessed
- The GLM allows links between linear models and data with a range of distributions
- Multilevel models can be effective tools for fitting data with multiple grains of variation
  - Especially important for subjects/items

# More generally

**Statistics as a "bag of tricks"**

- Tests and assumptions
  - check assumptions
  - apply test
- Significance testing
  - without looking for meaningfulness

**Statistical tools for modeling data**

- Modeling
  - fit model
  - check fit
- Meaningful interpretation
  - significance quantifies belief in parameter estimates