Multilevel Linear Models

Measurement & Evaluation of HCC Systems

Multilevel models

Today's goal:

Evaluate regression models in the presence of multilevel data

Outline:

- Theory of multilevel models
- Multilevel models in R
- Growth models in R
- Generalized linear mixed effect models in R

$x \rightarrow y$

Theory

of multilevel models

Multiple levels

Let’s say I want to test the effect of gender on performance in this class…

- Ankur (M)
- Kevin (M)
- Matias (M)
- Paritosh (M)
- Yifang (F)

Multiple levels

In two classes...

Treat class as X variable

2016:

- Ankur (M)
- Kevin (M)
- Matias (M)
- Paritosh (M)
- Yifang (F)

2017:

- Adam (M)
- Brian (M)
- Chen (M)
- Daphne (F)
- Elisa (F)
- Fiona (F)
- Grant (M)

Multiple levels

In many classes…

repeated measures!

2016:

- Ankur (M)
- Kevin (M)
- Matias (M)
- Paritosh (M)
- Yifang (F)

2017:

- Adam (M)
- Brian (M)
- Chen (M)
- Daphne (F)
- Elisa (F)
- Fiona (F)
- Grant (M)

2018:

- Hosub (M)
- Izak (M)
- James (M)
- Kathy (F)
- Lydia (F)
- Moury (F)
- Noopur (F)
- Olga (F)

2019:

- Praneet (M)
- Quincy (M)
- Rohit (M)
- Sonya (F)
- Thomas (M)

Multiple levels

In many classes + multiple assignments

…three-level model

2016:

- Ankur (M) a1…a7
- Kevin (M) a1…a7
- Matias (M) a1…a7
- Paritosh (M) a1…a7
- Yifang (F) a1…a7

2017:

- Adam (M) a1…a7
- Brian (M) a1…a7
- Chen (M) a1…a7
- Daphne (F) a1…a7
- Elisa (F) a1…a7
- Fiona (F) a1…a7
- Grant (M) a1…a7

2018:

- Hosub (M) a1…a7
- Izak (M) a1…a7
- James (M) a1…a7
- Kathy (F) a1…a7
- Lydia (F) a1…a7
- Moury (F) a1…a7
- Noopur (F) a1…a7
- Olga (F) a1…a7

2019:

- Praneet (M) a1…a7
- Quincy (M) a1…a7
- Rohit (M) a1…a7
- Sonya (F) a1…a7
- Thomas (M) a1…a7

Multiple levels

A repeated-measures ANOVA is also a multilevel model:

- Ankur (M): system A, system B, system C, system D
- Kevin (M): system A, system B, system C, system D
- Matias (M): system A, system B, system C, system D
- Paritosh (M): system A, system B, system C, system D
- Yifang (F): system A, system B, system C, system D

Multiple levels

Advantages of multilevel models (over repeated measures ANOVA)

- You can have a different number of within-subjects observations per subject (e.g. # of students per class)
- You can have more than two levels (e.g. assignment within student within class)
- You can have continuous measurements at each level (e.g. assignment difficulty, student gender, class topic)
- You can have random slopes (e.g. effect of assignment difficulty may differ per student or per class)

Multiple levels

Other advantages:

- Heterogeneity and non-sphericity are not a problem
- Independence is not necessary
- Missing data is okay
- Can easily be extended to non-linear models (e.g. logistic, Poisson, ordered logistic)

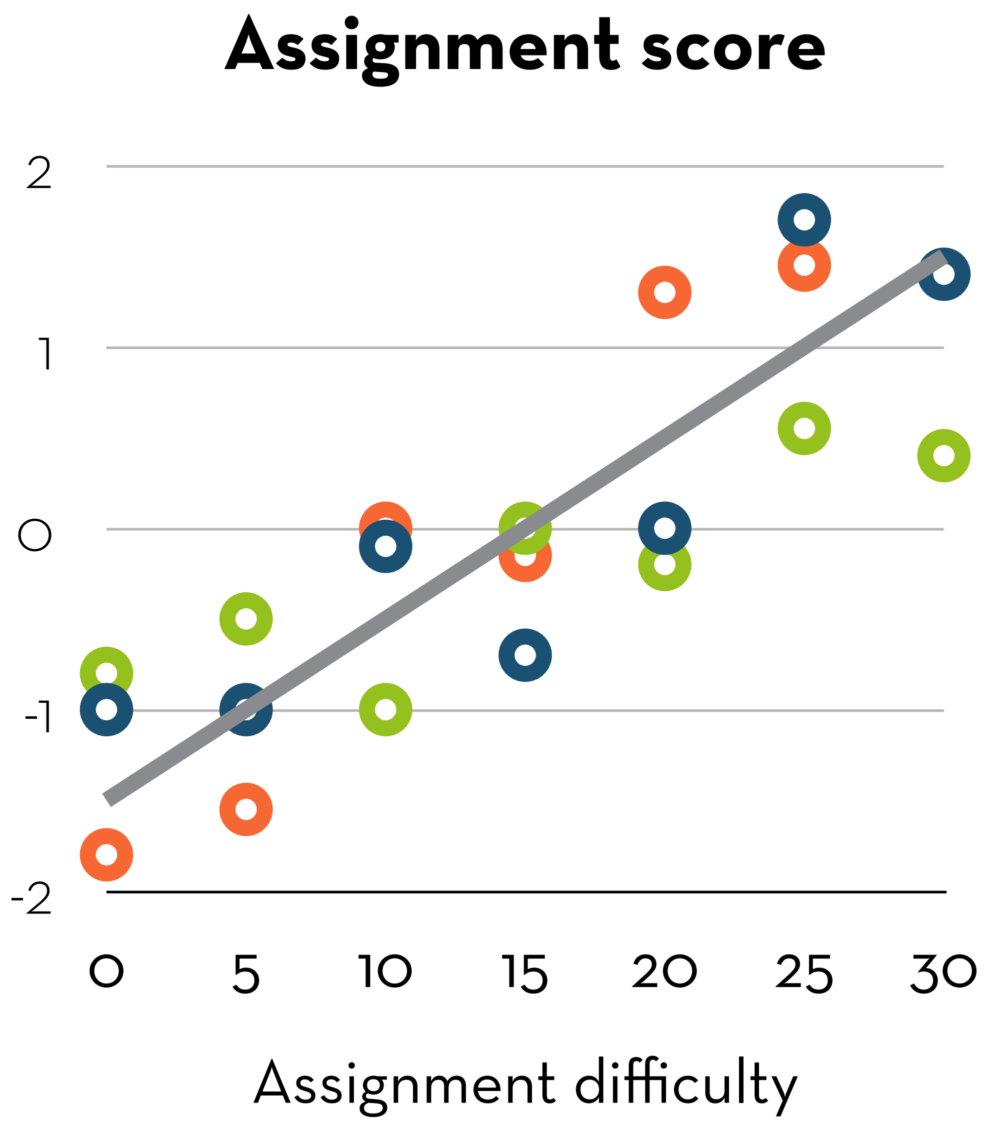
Random effects

Data from three participants:

Adam, Brian, Chen

Fixed intercept + slope

$Y_i = a + b_1 X_{diff} + e_i$



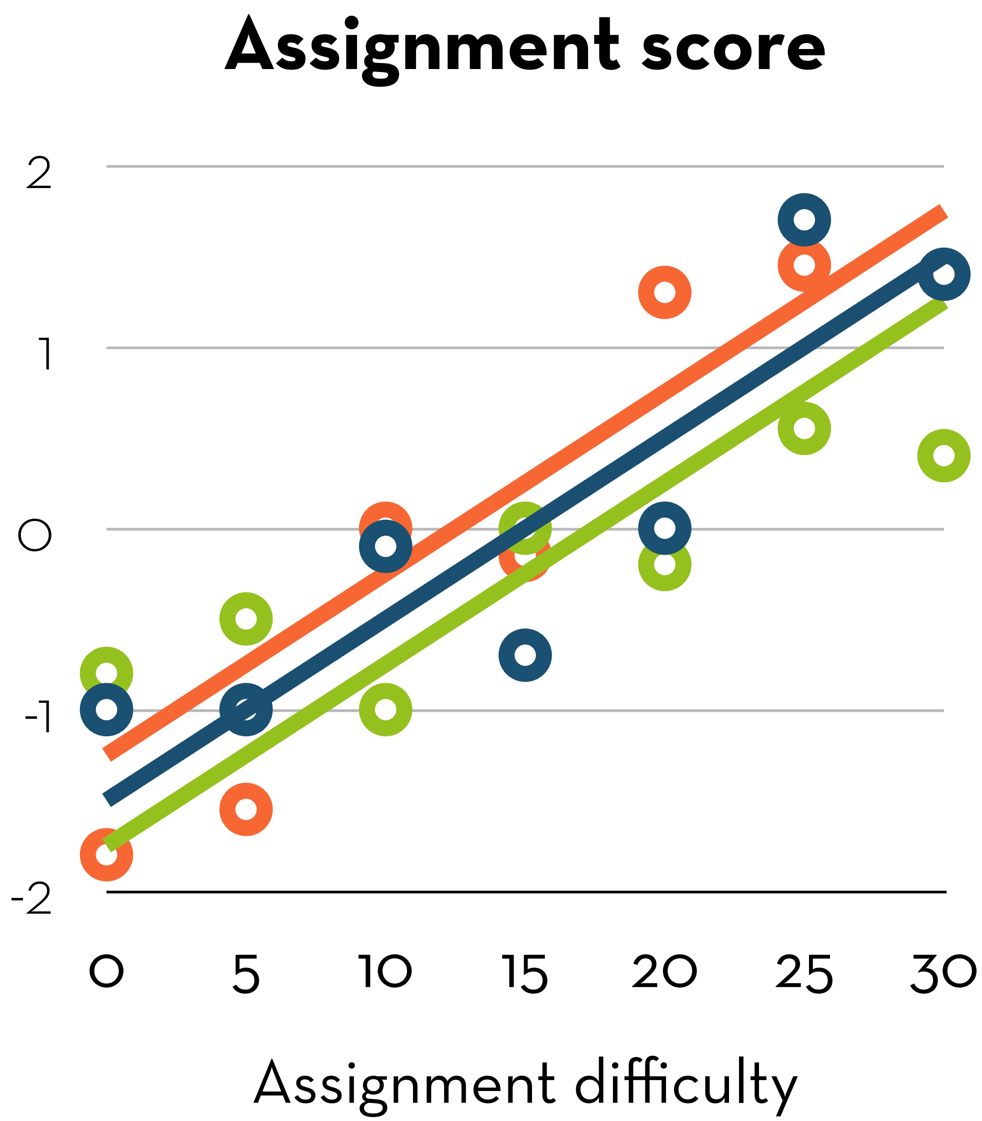
Random effects

Data from three participants:

Adam, Brian, Chen

Different intercept + fixed slope

$Y_i = a + b_1X_{diff} + b_2X_{brian} +$
$b_3X_{chen} + e_i$



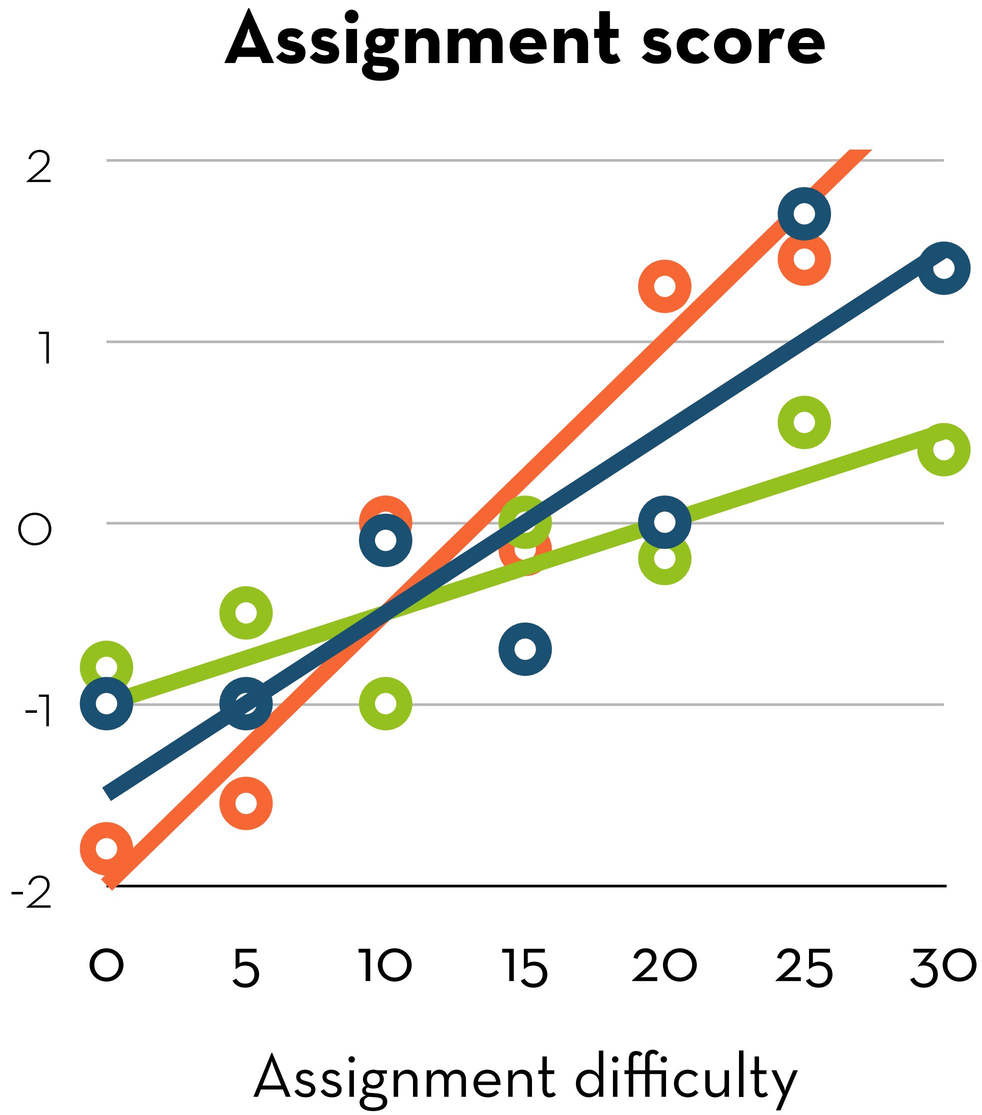
Random effects

Data from three participants:

Adam, Brian, Chen

Different intercept + different slope

$$Y_i = a + b_1X_{diff} + b_2X_{brian} + b_3X_{chen} + b_4X_{diff}X_{brian} + b_5X_{diff}X_{chen} + e_i$$



Random effects

Data from many participants

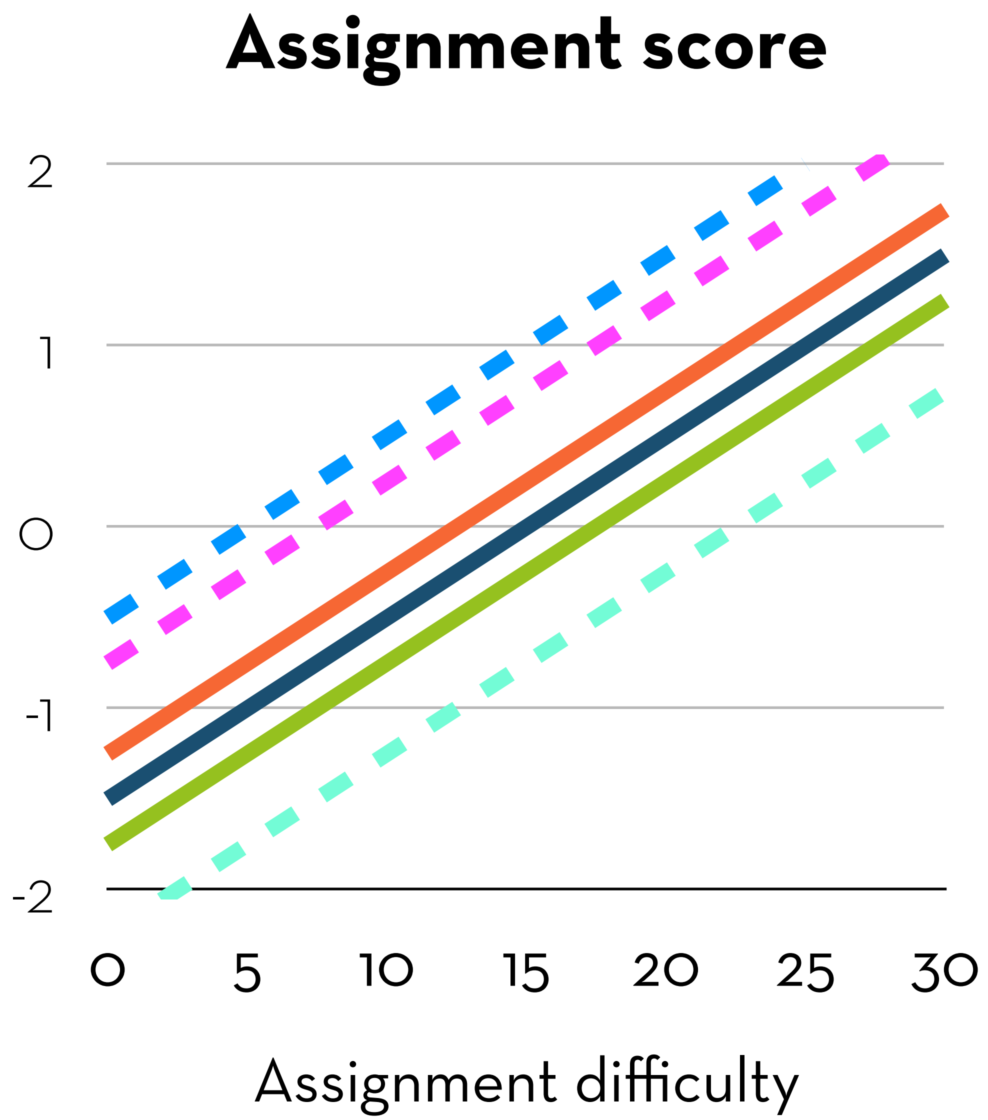
Random intercept + fixed slope

$Y_{ip} = a_p + b_1X_{diff} + e_{ip}$

where $a_p = a + u_p$

u_p differs per participant!

we fit a single parameter for it (variance)



Random effects

Data from many participants

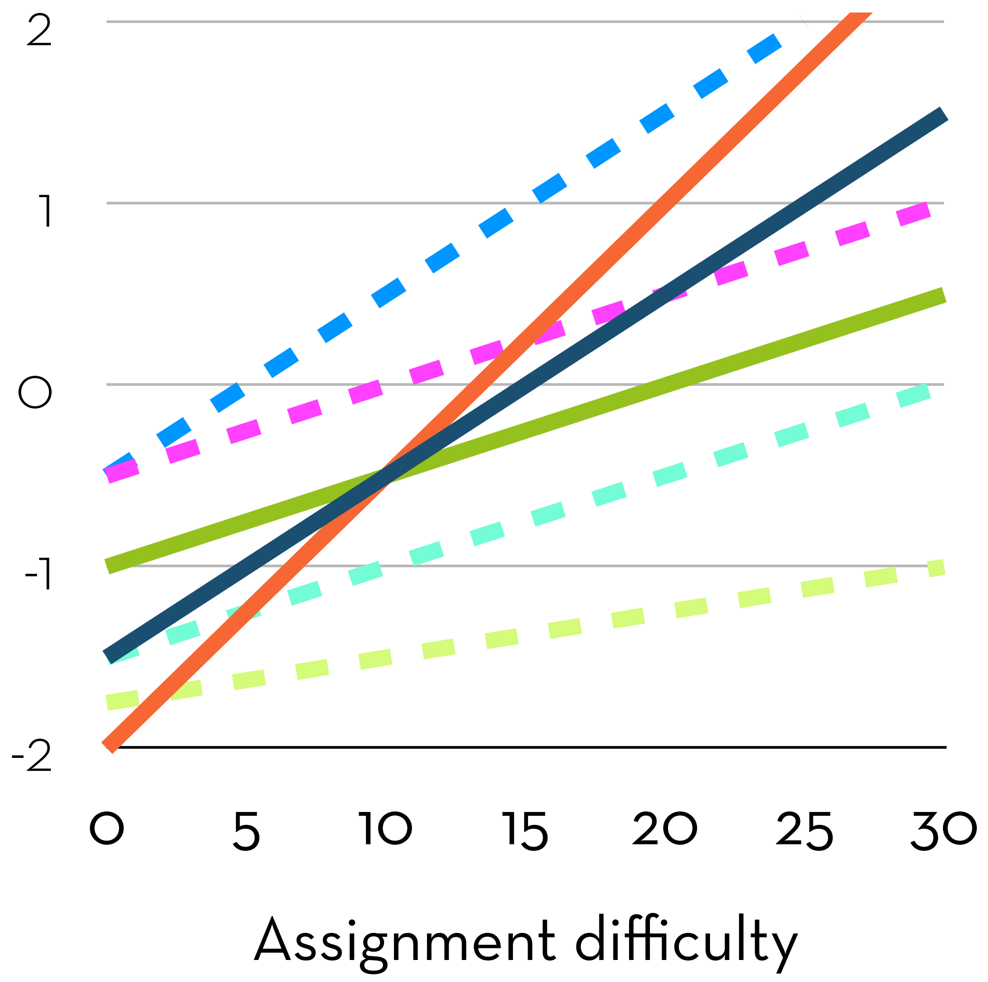
Random intercept + random slope

$$Y_{ip} = a_p + b_{1p}X_{diff} + e_{ip}$$

where $a_p = a + u_p$

and $b_{1p} = b_1 + v_p$

Both u_p and v_p differ per participant!



Assignment score

Error covariance

Normally, e_i is uncorrelated

This means the correlation of errors between assignments per participant looks like this:

p1	a1	a2	a4	a5	a5	a6	a7
a1	1	0	0	0	0	0	0
a2	0	1	0	0	0	0	0
a3	0	0	1	0	0	0	0
a4	0	0	0	1	0	0	0
a5	0	0	0	0	1	0	0
a6	0	0	0	0	0	1	0
a7	0	0	0	0	0	0	1

Error covariance

The effect of a random intercept is that errors become correlated.

u_p plus a simple e_i results in:

p1	a1	a2	a4	a5	a5	a6	a7
a1	1	s_u	s_u	s_u	s_u	s_u	s_u
a2	s_u	1	s_u	s_u	s_u	s_u	s_u
a3	s_u	s_u	1	s_u	s_u	s_u	s_u
a4	s_u	s_u	s_u	1	s_u	s_u	s_u
a5	s_u	s_u	s_u	s_u	1	s_u	s_u
a6	s_u	s_u	s_u	s_u	s_u	1	s_u
a7	s_u	s_u	s_u	s_u	s_u	s_u	1

Note: between subjects the errors are still independent!!

Error covariance

Sometimes, your within-group level consists of time-steps.

In that case, you want the correlation between adjacent time steps to be higher

You can use the following e_{ip} instead — this is called AR(1):

p1	a1	a2	a4	a5	a5	a6	a7
a1	1	r	r^2	r^3	r^4	r^5	r^6
a2	r	1	r	r^2	r^3	r^4	r^5
a3	r^2	r	1	r	r^2	r^3	r^4
a4	r^3	r^2	r	1	r	r^2	r^3
a5	r^4	r^3	r^2	r	1	r	r^2
a6	r^5	r^4	r^3	r^2	r	1	r
a7	r^6	r^5	r^4	r^3	r^2	r	1

Error covariance

When you have both random intercepts and random slopes, the error covariance becomes super confusing

In that case, you are better off using the following e_{ip} — this is called an “unstructured” error covariance matrix:

p1	a1	a2	a4	a5	a5	a6	a7
a1	d_1^2	d_{12}	d_{13}	d_{14}	d_{15}	d_{16}	d_{17}
a2	d_{12}	d_2^2	d_{23}	d_{24}	d_{25}	d_{26}	d_{27}
a3	d_{13}	d_{23}	d_3^2	d_{34}	d_{35}	d_{36}	d_{37}
a4	d_{14}	d_{24}	d_{34}	d_4^2	d_{45}	d_{46}	d_{47}
a5	d_{15}	d_{25}	d_{35}	d_{45}	d_5^2	d_{56}	d_{57}
a6	d_{16}	d_{26}	d_{36}	d_{46}	d_{56}	d_6^2	d_{67}
a7	d_{17}	d_{27}	d_{37}	d_{47}	d_{57}	d_{67}	d_7^2

Comparing models

Nested models: -2LL test

- e.g. test fixed model vs. random intercept vs. random intercept + slope

Non-nested models: AIC or BIC (based on -2LL)

- Lower is better
- BIC prefers simpler models than AIC

Only works when method="ML"

Assumptions

Outcome should be quantitative, continuous, unbounded

Predictors should not be too highly correlated (centering helps!)

No variables correlated with both X and Y should be left out

Homoscedasticity and ~~**independence**~~

Linearity (although we can test for some non-linear effects)

Random effects should be normally distributed

Sample size?

Very hard to determine!

- At least N=20 at each level with X variables

Centering

We already considered **grand mean centering** as a way to reduce multicollinearity when doing interactions with linear X variables

You subtract the grand mean from each value

We already considered **group mean centering** as a way to remove between-subjects error from plotted error bars

You subtract the (between-subjects) group mean from each (within-subjects) value

Centering

In LME, you can use either of these to center your X variables. Which one should you use?

- If you care mainly about within-subjects effects or cross-level interactions: group mean centering
- If you care mainly about between-subjects effects: grand mean centering
- If you want to compare effects at different levels: group mean centering + add grand-centered group means to the model

Centering example

Group mean centering:

Easier to see within-subjects effects

	a1	a2	a3	a4	a5	a6	a7	M
p1	83	74	83	89	96	85	85	**85**
p2	72	68	78	77	74	73	76	**74**
p3	92	91	95	94	95	95	96	**94**
p4	44	43	48	47	51	42	40	**45**
p5	59	49	54	68	69	46	54	**57**
								71

	a1	a2	a3	a4	a5	a6	a7
p1	-2	-11	-2	4	11	0	0
p2	-2	-6	4	3	0	-1	2
p3	-2	-3	-1	0	1	1	2
p4	-1	-2	3	2	6	-3	-5
p5	2	-8	-3	11	12	-9	-3

Centering example

Grand mean centering:

Easier to see between-subjects effects

	a1	a2	a3	a4	a5	a6	a7	M
p1	83	74	83	89	96	85	85	**85**
p2	72	68	78	77	74	73	76	**74**
p3	92	91	95	94	95	95	96	**94**
p4	44	43	48	47	51	42	40	**45**
p5	59	49	54	68	69	46	54	**57**
								71

	a1	a2	a3	a4	a5	a6	a7	M
p1	12	3	12	18	25	14	14	**14**
p2	1	-3	7	6	3	2	5	**3**
p3	21	20	24	23	24	24	25	**23**
p4	-27	-28	-23	-24	-20	-29	-31	**-26**
p5	-12	-22	-17	-3	-2	-25	-17	**-14**

Centering example

Group mean centering + add grand-centered group means:

Easier to compare effects at different levels

	a1	a2	a3	a4	a5	a6	a7	M
p1	83	74	83	89	96	85	85	**85**
p2	72	68	78	77	74	73	76	**74**
p3	92	91	95	94	95	95	96	**94**
p4	44	43	48	47	51	42	40	**45**
p5	59	49	54	68	69	46	54	**57**
								71

	a1	a2	a3	a4	a5	a6	a7	M
p1	-2	-11	-2	4	11	0	0	**14**
p2	-2	-6	4	3	0	-1	2	**3**
p3	-2	-3	-1	0	1	1	2	**23**
p4	-1	-2	3	2	6	-3	-5	**-26**
p5	2	-8	-3	11	12	-9	-3	**-14**

$x \rightarrow y$

Multilevel in R

using random intercepts and random slopes

Multilevel in R

Dataset: Cosmetic Surgery.dat -> rename to surgery

Effect of cosmetic surgery on quality of life

Variables:

Post_QoL: quality of life after surgery

Base_QoL: quality of life before surgery

Surgery: whether they had surgery (1) or on waiting list (0)

Clinic: which clinic they went to (1-10)

Age: age in years

BDI: depression index

Reason: whether it was to improve appearance (0) or for physical reason (1)

Gender: male (1) or female (0)

First a simple lm...

Model with Surgery as X:

```
surgeryLM <- lm(Post_QoL ~ Surgery, data=surgery)
```

Model controlling for Base_QoL:

```
surgerycontrolLM <- lm(Post_QoL ~ Surgery +
Base_QoL, data=surgery)
```

Multilevel?

Test whether we need to control for “clinic” in this model

How? Compare a baseline model without multilevel to a baseline model with multilevel:

```
baseline <- gls(Post_QoL ~ 1, data=surgery, method=“ML”)
random <- lme(Post_QoL ~ 1, data=surgery, random =
~1|Clinic, method=“ML”)
anova(baseline, random)
```

Run lme...

Add surgery, Base_QoL, compare, inspect

```
randomSurgery <- update(random, .~. + Surgery)
randomSurgeryControl <- update(randomSurgery, .~. + Base_QoL)
anova(random, randomSurgery, randomSurgeryControl)
summary(randomSurgeryControl)
```

Random slope

Add a random slope for Surgery (interpretation: does the effect of surgery change per clinic?):

```
addRandomSlope <- lme(Post_QoL ~ Surgery +
Base_QoL, data=surgery, random = ~Surgery|Clinic,
method=“ML”)

summary(addRandomSlope)

anova(randomSurgeryControl, addRandomSlope)
```

Random slope

```
               StdDev   Corr
(Intercept) 6.132655 (Intr)
Surgery     6.197489 -0.965
Residual    5.912335

Fixed effects: Post_QoL ~ Surgery + Base_QoL
                Value Std.Error  DF   t-value p-value
(Intercept) 40.10253  3.892945 264 10.301334  0.0000
Surgery     -0.65453  2.110917 264 -0.310069  0.7568
Base_QoL     0.31022  0.053506 264  5.797812  0.0000
```

Note the standard deviation of the intercept, the slope, and their correlation

The correlation is the reason why we have 2 additional df!

Add an interaction

Add the Reason for the surgery, and its interaction with Surgery

```
addReason <- update(addRandomSlope, .~. + Reason)
full <- update(addReason, .~. + Reason:Surgery)
anova(addRandomSlope, addReason, full)
```

Get the final model:

```
summary(full)
intervals(full)
```

Add an interaction

```
                   Value Std.Error  DF   t-value p-value
(Intercept)     42.51782  3.875318 262 10.971440  0.0000
Surgery         -3.18768  2.185369 262 -1.458645  0.1459
Base_QoL         0.30536  0.053125 262  5.747833  0.0000
Reason          -3.51515  1.140934 262 -3.080938  0.0023
Surgery:Reason   4.22129  1.700269 262  2.482717  0.0137
```

Interpretation:

With appearance reason, Surgery reduces QoL (but n.s.)

Base_QoL has a strong effect (as expected)

Having a physical reason for the surgery reduces QoL when you get no surgery

With physical reason, the effect of Surgery is significantly more positive than with appearance reason

Add an interaction

Can we look at the effect within each group (appearance vs. physical reason)?

Yes, using subset!

```
physicalModel <- lme(Post_QoL ~ Surgery + Base_QoL,
data=surgery, random = ~Surgery|Clinic, subset = Reason
== 1, method = “ML”)
```

```
appearanceModel <- lme(Post_QoL ~ Surgery +
Base_QoL, data=surgery, random = ~Surgery|Clinic,
subset = Reason == 0, method = “ML”)
```

Reporting

See 19.8 for guidelines to report the model

Create a table for the effects

- Like with lm and glm, you can display multiple models in the same table
- compare models with likelihood ratio test

$x \rightarrow y$

Growth Models

when your within-subjects variable is “time”

Growth Models

Testing trends in time-series data

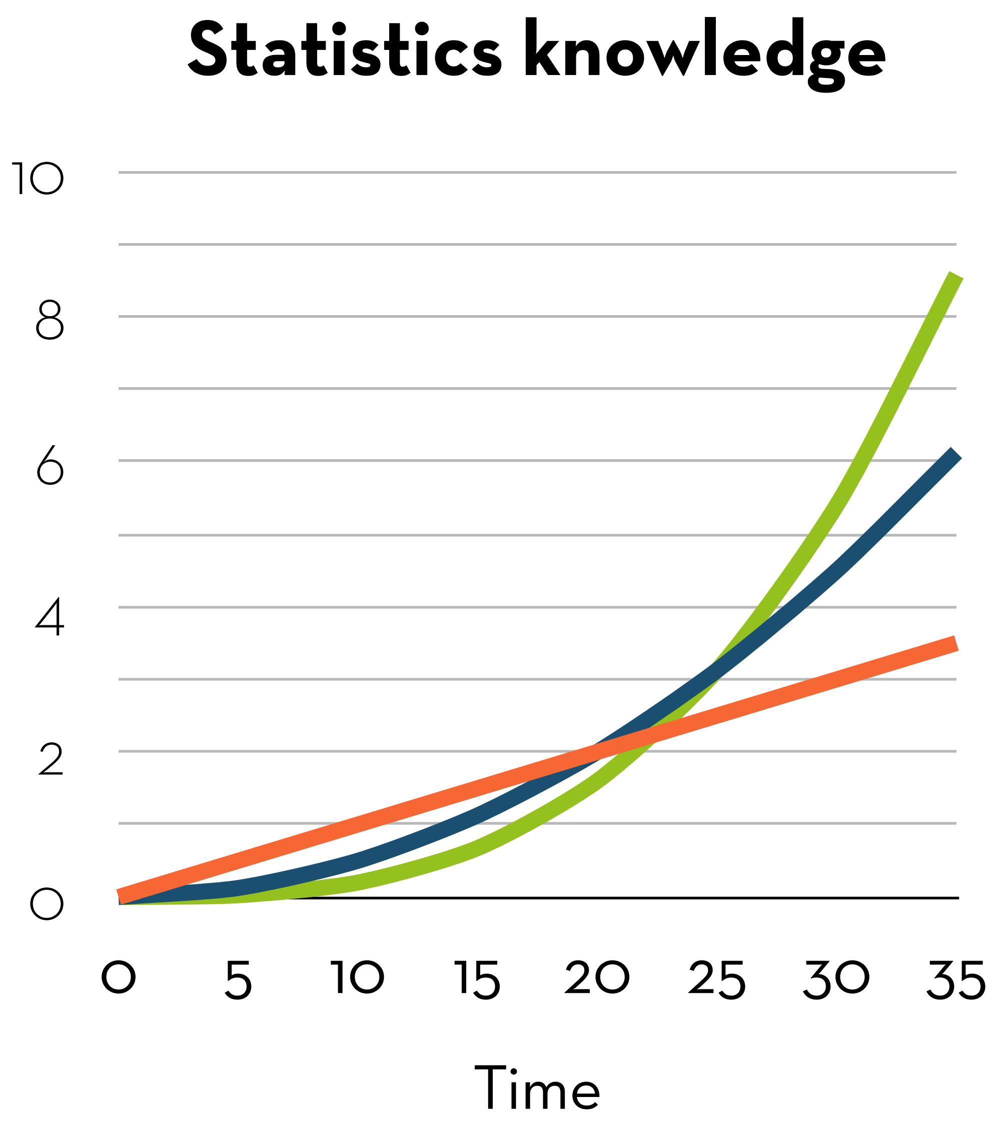
linear (time)

quadratic (add time^2)

cubic (add time^3)

...anything larger is probably not useful

In most cases, AR(1) is used


Statistics knowledge

Growth Models in R

Dataset: Honeymoon Period.dat -> rename to honeymoon

Effect of of marriage on life satisfaction over time

Variables:

Person: participant ID

Satsifaction_Baseline, Satsifaction_6_Months, Satsifaction_12_Months, Satsifaction_18_Months: life satisfaction after 0, 6, 12 and 18 months of marriage

Gender: participant gender

Long format

Reshape to long format, with time points 0, 1, 2, 3 (number of 6-month periods since marriage):

```
hlong <- reshape(honeymoon, direction=“long”,
idvar=c(“Person”), varying=list(2:5), times=c(0,1,2,3),
v.names = “satisfaction”)
```

Build models

Build a baseline and random intercept model:

```
baseline <- gls(satisfaction ~ 1, data=hlong, method=“ML”,
na.action = na.exclude)

random <- lme(satisfaction ~ 1, data=hlong, random =
~1|Person, method=“ML”, na.action = na.exclude, control =
list(opt=“optim”))
```

Build models

Add time, add random slope for time:

```
time <- update(random, .~. + time)
timeRS <- update(time, random = ~time|Person)
```

Change error covariance to AR(1):

```
ARmodel <- update(timeRS, correlation = corAR1(0, form
= ~time|Person))
```

Compare all models:

```
anova(baseline, random, time, timeRS, ARmodel)
```

Build models

Add quadratic time, add random slope for quadratic time, compare:

```
timeq <- update(ARmodel, .~. + I(time^2))
timeqRS <- update(timeq, random = ~time+I(time^2)|Person, correlation = corAR1(0, form = ~time+I(time^2)|Person))
anova(ARmodel, timeq, timeqRS)
```

Build models

Add cubic time, compare:

```
timec <- update(timeq, .~. + I(time^3))
anova(timeq, timec)
```

Best model: timeq!

Plot:

```
ggplot(hlong, aes(time,satisfaction)) +
geom_point(position=position_jitter(width=0.1)) +
stat_summary(fun.y=mean, geom=”line”) +
geom_smooth(method=”lm”, formula = y ~ x + I(x^2))
```

Good job!

You finished >80% of Andy Field's book!

$x \rightarrow y$

GLME

Generalized Linear Mixed Effects models in R

GLME

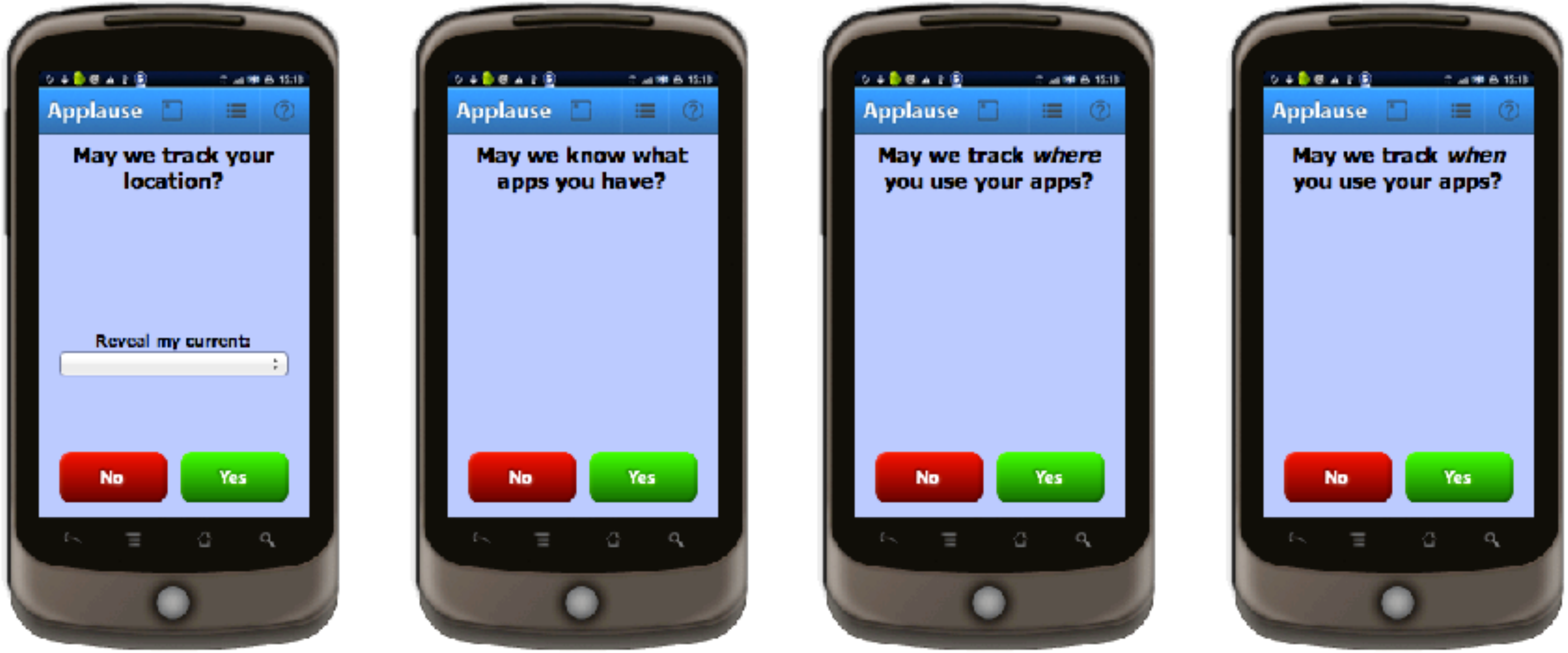
Dataset: disclosure.dat

- 396 participants (level 2) each make disclosure decisions (binary) about 31 items (level 1)

Justifications (between subjects):

- None
- Useful-for-you
- % of others
- Useful for others
- Explanation

GLME



May we track your location?

Reveal my current:

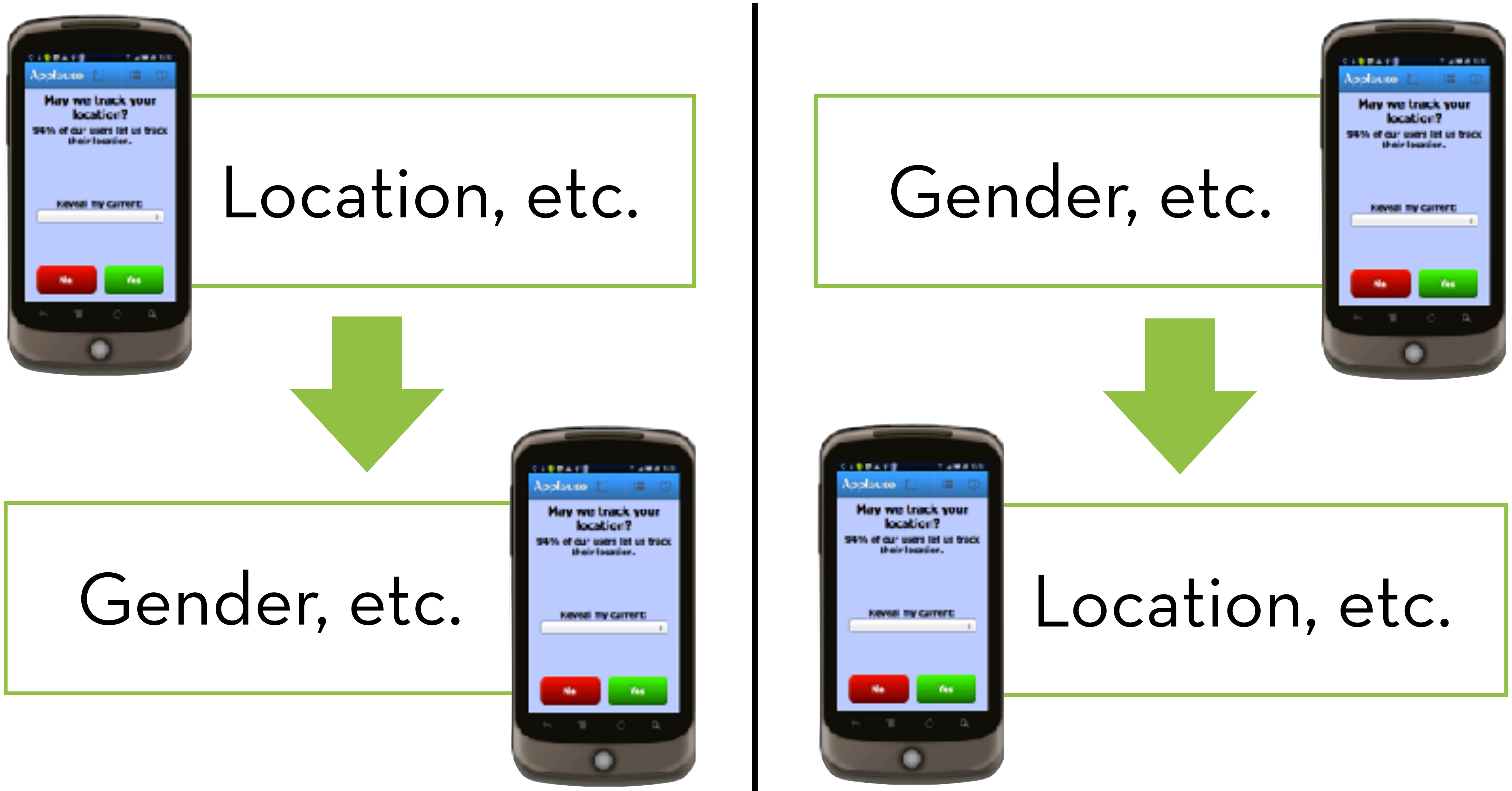
May we know what apps you have?

May we track *where* you use your apps?

May we track *when* you use your apps?

● ● ●

GLME


May we track your location?
94% of our users let us track their location.
Reveal my current:
No
Yes

Location, etc.

Gender, etc.

Context data first

Gender, etc.

Location, etc.

Demographical data first

GLME

5 justification types

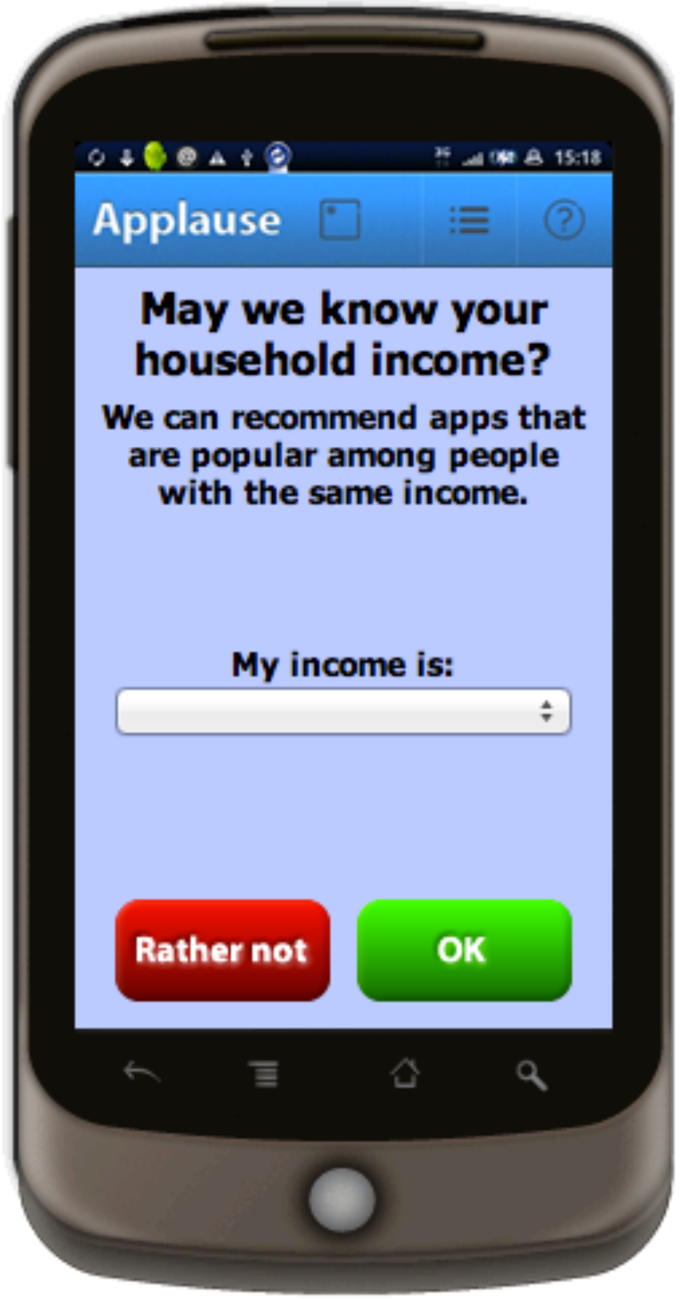
None

Useful for you

Number of others

Useful for others

Explanation



Applause

May we know your household income?

We can recommend apps that are popular among people with the same income.

My income is:

GLME

Variables at level 1:

- decision: whether the participant disclosed the item (1) or not (0)
- qid: question ID
- qcat: type of question (context or demographic)
- pos: position of the question (semi-randomized)
- perc: percentage used in the justification, centered around 50% (manipulated, only for types 2, 3 and 4)

GLME

Variables at level 2:

- id: participant id
- message: the justification (manipulated)
- gord: order in in which questions are asked (manipulated)
- satisfaction: expected satisfaction with the system
- concern: privacy concern
- age
- gender

Build models

Turn message into a factor with “none” as the baseline:

```
disclosure$message <- factor(disclosure$message,
levels=c(“none”, “num_other”, “useful_you”, “userful_other”,
”explanation”))
```

Build models

Load package “lme4”

Build a baseline and random intercept model:

```
baseline <- glm(decision ~ 1, data=disclosure,
family=binomial)
```

```
randompart <- glmer(decision ~ 1 + (1|id), data=disclosure,
family=binomial)
```

ANOVA doesn’t work, so we compare deviances by hand:

```
pchisq(deviance(baseline)-deviance(randompart), 1,
lower=F)
```

Build models

Add message and percentage:

```
msg <- update(randompart, .~. + message)
perc <- update(msg, .~. + perc)
msgperc <- update(perc, .~. message:perc)
anova(randompart, msg, perc, msgperc)
```

Inspect msgperc:

```
summary(msgperc)
exp(fixef(msgperc))
```

Build models

Add gord and qcat:

```
order <- update(msgperc, .~. + gord)
type <- update(order, .~. + qcat)
ordertype <- update(type, .~. + gord:qcat)
anova(msgperc, order, type, ordertype)
```

Build models

Add satisfaction and concern:

```
sat <- update(ordertype, .~. + satisfaction)
concern <- update(sat, .~. + concern)
anova(ordertype, sat, concern)
```

Final model output and odds ratio confidence intervals:

```
summary(concern)
exp(fixef(concern))
exp(confint(concern))
```

Advanced...

Add a random intercept for item:

```
randitem <- update(concern, .~. + (1|qid)
anova(concern, randitem)
```

We now have “crossed” random intercepts!

Advanced...

Add a random slope for position within participant:

```
randpos <- update(concern, .~. + (pos|id)
anova(concern, randpos)
```

Advanced...

Install “geepack”

Run a “geeglm” to get an AR(1) covariance structure:

```
ar1 <- geeglm(decision ~  message*perc + gord*qcat +
satisfaction + concern, id=id, data=disclosure, waves=pos,
family=binomial, corstr=“ar1”)

summary(ar1)
```

Advanced...

Run a “geeglm” to get an unstructured covariance structure:

```
unstr <- geeglm(decision ~  message*perc + gord*qcat +
satisfaction + concern, id=id, data=disclosure,
family=binomial, corstr=“unstructured”)

anova(ar1,unstr)

summary(unstr)
```

Prepare to wait...

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

THANKS!

George Bernard Shaw