# Evaluating IUIs

## with User Experiments


$h_o$
AB
$x \to y$

# Introduction

Welcome everyone!

# Introduction

Bart Knijnenburg

Current: Clemson University

Asst. Prof. in Human-Centered Computing

University of California, Irvine

PhD in Informatics

Carnegie Mellon University

Master in Human-Computer Interaction

Eindhoven University of Technology

Researcher & teacher
MS in Human-Technology interaction
BS in Innovation Sciences

# Introduction

Bart Knijnenburg

## User-centric evaluation

Framework for user-centric evaluation of recommender systems (UMUA 2012)

Chapter in Recommender Systems Handbook

Tutorial at RecSys conference

11 years of experience as a statistics teacher

## Recommender Systems

Research on preference elicitation methods

## Privacy decision-making

Research on adaptive privacy decision support

# Introduction

"A user experiment is a scientific method to investigate factors that influence how people interact with systems"

"A user experiment systematically tests how different system aspects (manipulations) influence the users' experience and behavior (observations)."

# Introduction

My goal:

Teach how to scientifically evaluate intelligent user interfaces using a user-centric approach

My approach:

- I will talk about how to develop a research model
- I will cover every step in conducting a user experiment
- I will teach the "statistics of the 21st century"

# Introduction

Slides and data:

www.usabart.nl/QRMS

Contact info:

E: bartk@clemson.edu

W: www.usabart.nl

T: @usabart

## Introduction
Welcome everyone!

## Hypotheses
Developing a research model

## Participants
Population and sampling

## Testing A vs. B
Experimental manipulations

## Analysis
Statistical evaluation of the results

## Measurement
Measuring subjective valuations

## Evaluating Models
An introduction to Structural Equation Modeling

# Hypotheses

Developing a research model

# Hypotheses

"Can you test if my system is good?"

# Problem…

What does **good** mean?

- Learnability? (e.g. number of errors?)
- Efficiency? (e.g. time to task completion?)
- Usage satisfaction? (e.g. usability scale?)
- Outcome quality? (e.g. survey?)

We need to define **measures**

# Measurement

Measurements: observed or subjective?

Behavior is an "observed" variable

- Relatively easy to quantify
- E.g. time, money spent, click count, yes/no decision

Perceptions, attitudes, and intentions (subjective valuations) are "unobserved" variables

- They happen in the user's mind
- Harder to quantify (more on this later)

# Better...

"Can you test if the user interface of my system scores high on this usability scale?"

# However...

What does **high** mean?

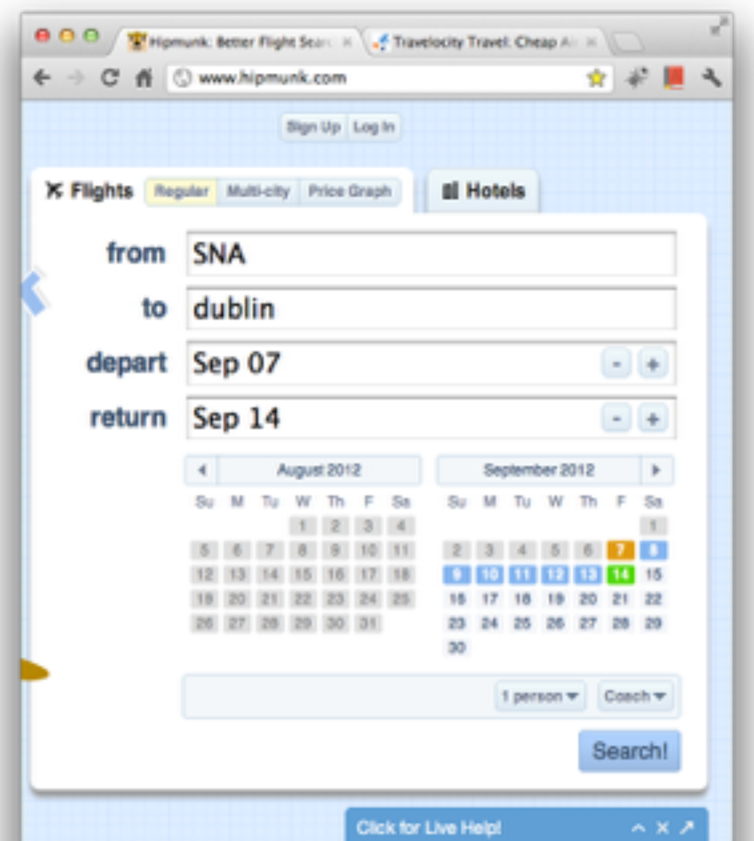
- Is 3.6 out of 5 on a 5-point scale “high”?
- What are 1 and 5?
- What is the difference between 3.6 and 3.7?

We need to **compare** the UI against something

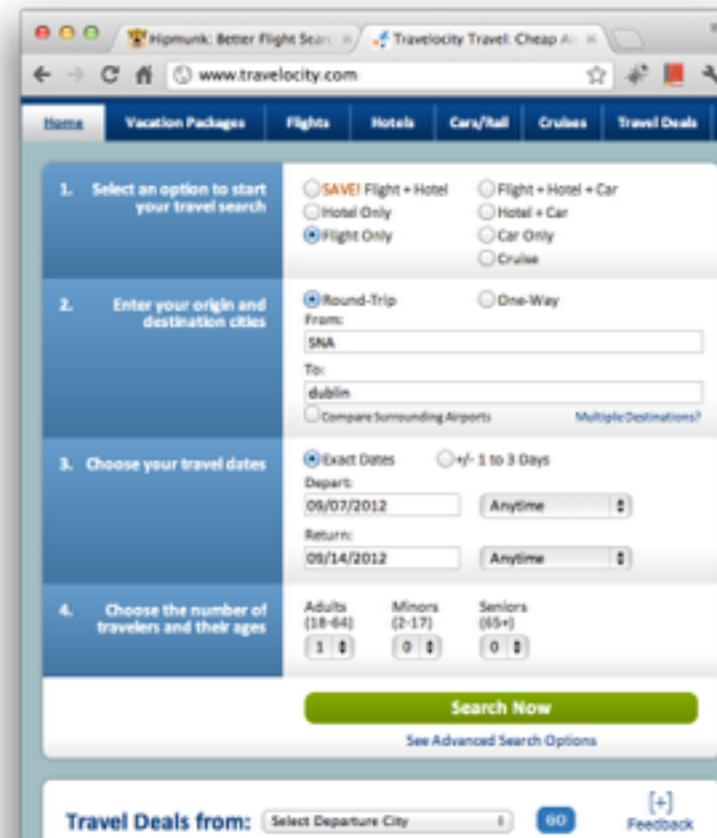
# Even better…

"Can you test if the UI of my system scores high on this usability scale compared to this other system?"

# Testing A vs. B




My new travel system




Travelocity

# However...

Say we find that it scores higher on usability... **why** does it?

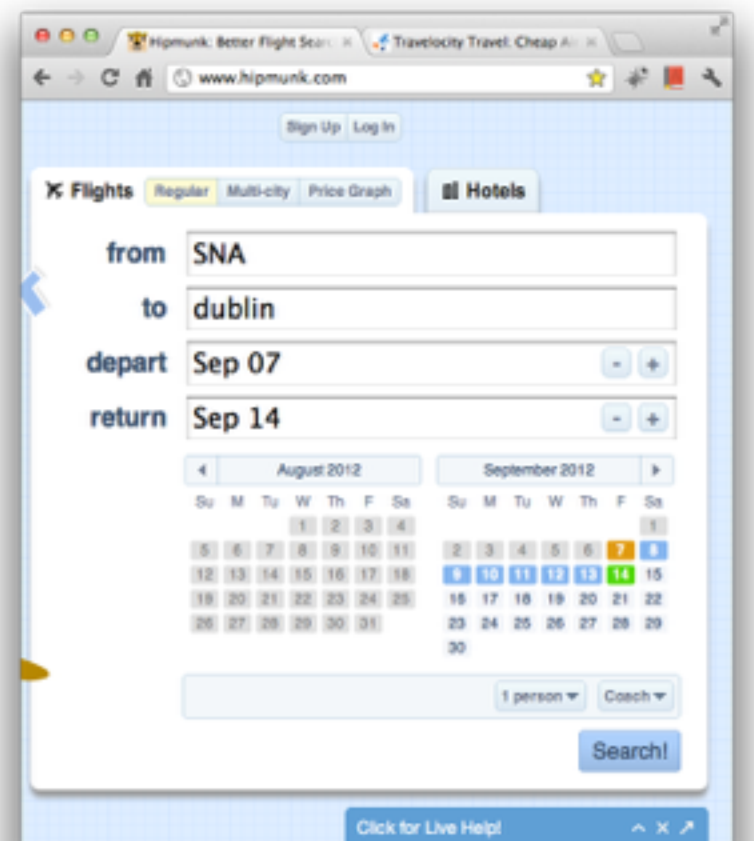
- different date-picker method
- different layout
- different number of options available

Apply the concept of **ceteris paribus** to get rid of confounding variables

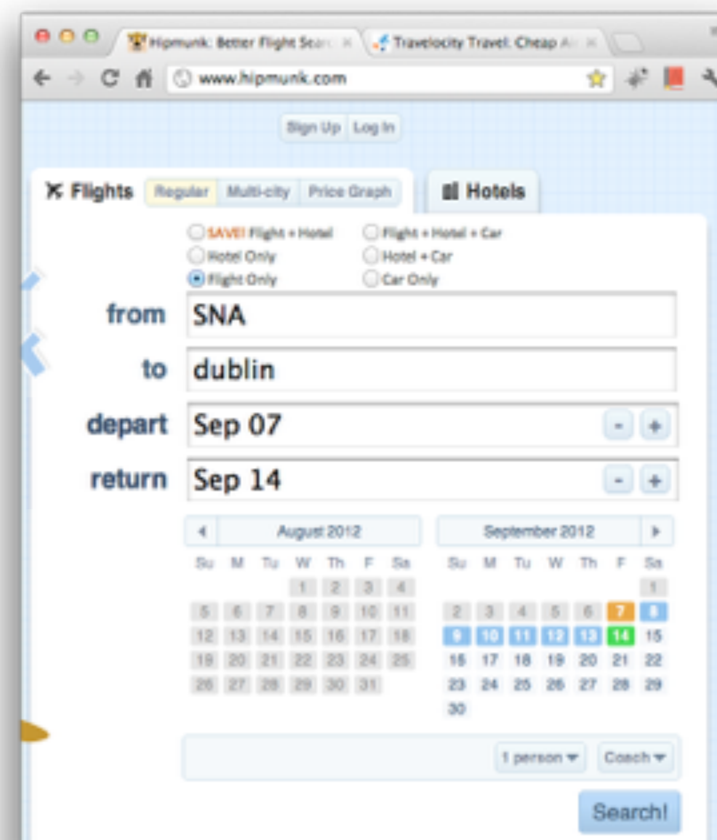
Keep everything the same, except for the thing you want to test (the manipulation)

Any difference can be attributed to the manipulation

# Ceteris Paribus




My new travel system




Previous version

(too many options)

# Theory behind x->y

To learn something from a study, we need a **theory** behind the effect

- This makes the work generalizable
- This may suggest future work

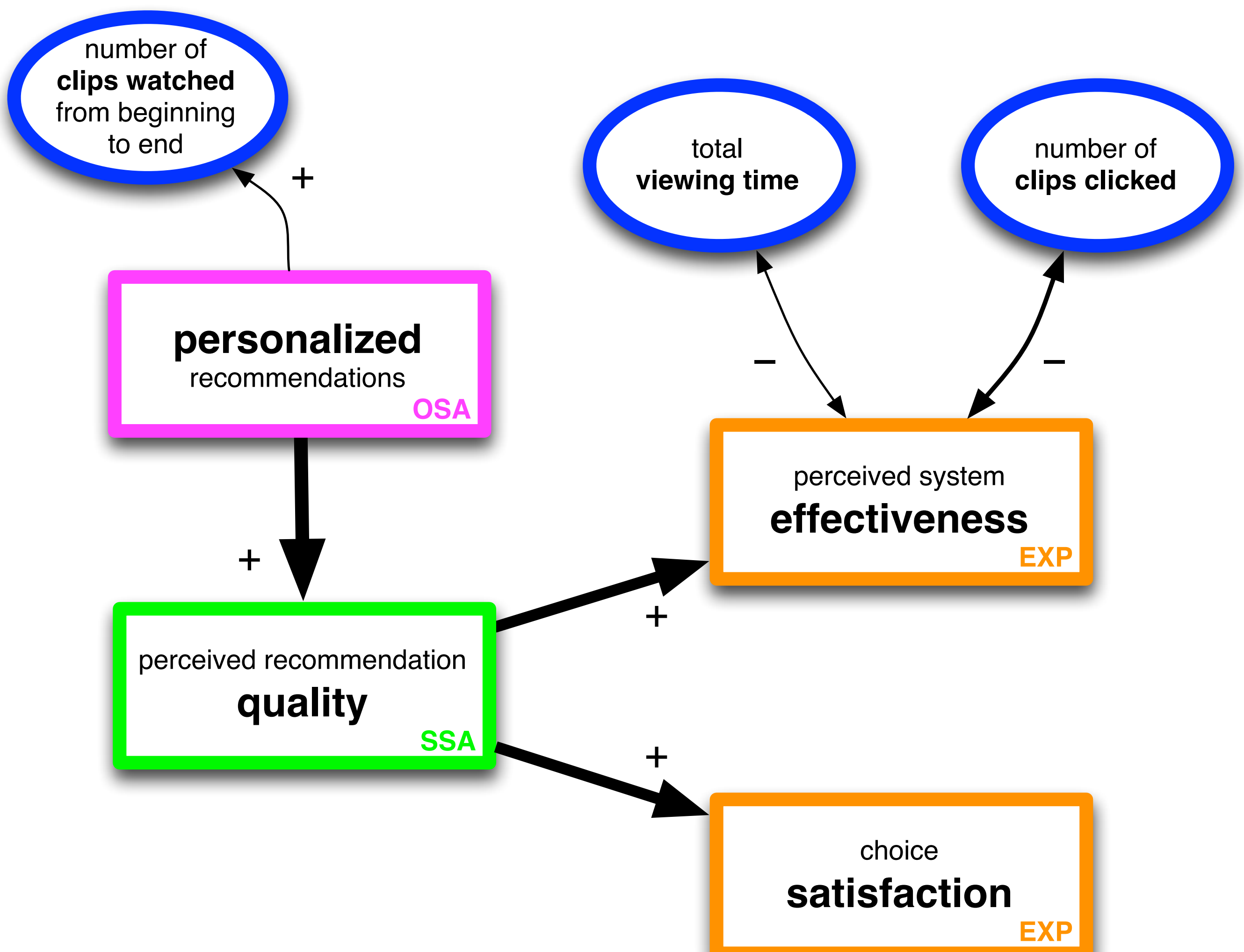
Measure **mediating variables**

- Measure understandability (and a number of other concepts) as well
- Find out how they mediate the effect on usability

# Example

"Testing a recommender against a random videoclip system, the number of clicked clips and total viewing time went down!"

# Example




Knijnenburg et al.: “Receiving Recommendations and Providing Feedback”, EC-Web 2010

# Lessons learned

Behavior is hard to interpret

Relationship between behavior and satisfaction is not always trivial

User experience is a better predictor of long-term retention

With behavior only, you will need to run for a long time

Questionnaire data is more robust

Fewer participants needed

# Hypotheses

Measure **subjective valuations** with questionnaires

- Perception, experience, intention

**Triangulate** these data with behavior

- Ground subjective valuations in observable actions
- Explain observable actions with subjective valuations

Create a **research model**

- System aspect -> perception -> experience -> behavior define **measures**

**compare** system aspects against each other

apply the concept of **ceteris paribus**

$h_0$

measure **subjective valuations**

look for a **theory** behind the found effects

# Hypotheses

What do I want to find out?

measure **mediating variables** to explain the effects

# Participants

Population and sampling

# Participants

Where to get them from?

- An unbiased sample of users of your system
- Not just friends an colleagues!

How many?

- Depends on the size of the effect
- Power analysis

# Where from?

Craigslist:

- Post in various cities under Jobs > Etcetera
- Create a geographically balanced sample

Amazon Mechanical Turk

- Often used for very small tasks, but Turk workers appreciate more elaborate studies
- Anonymous payment facilities.
- Set criteria for workers (e.g. U.S. workers with a high reputation)

# Where from?

Demographics reflect the general Internet population

- Craigslist users: a bit higher educated and more wealthy
- Turk workers: less likely to complain about tedious study procedures, but are also more likely to cheat

Make your study simple and usable

Use quality checks, add an open feedback item to catch unexpected problems

# How many?

Small studies (N << 100) may find medium or large effects that are not significant

Waste of resources! (unless they are pilot studies)

Large studies (N >> 100) may find very small effects that are significant

Also a waste of resources! (could have done with fewer)

How can we prevent wasting resources?

Do a power analysis!

# Power analysis

A calculation involving the following 4 parameters:

- Alpha (cut-off p-value, often .05)
- Power (probability of finding a true effect, often .80 or .85)
- N (sample size, usually the thing we are trying to calculate)
- Effect size (usually the expected effect size)

# Expected effect

An "educated guess" based on:

- Pilot study results
- Findings from similar studies
- Whatever is considered "meaningful"
- Educated guess

# G*Power demo

An existing study found that a new TurboTax interface reduced tax filing time from 3.0 hours (SD: 0.5 hours) to 2.7 hours (SD: 0.5 hours).

You created an adaptive interface that you think is even better. How many participants do you need to find an effect that is at least the same size? (assume 85% power)

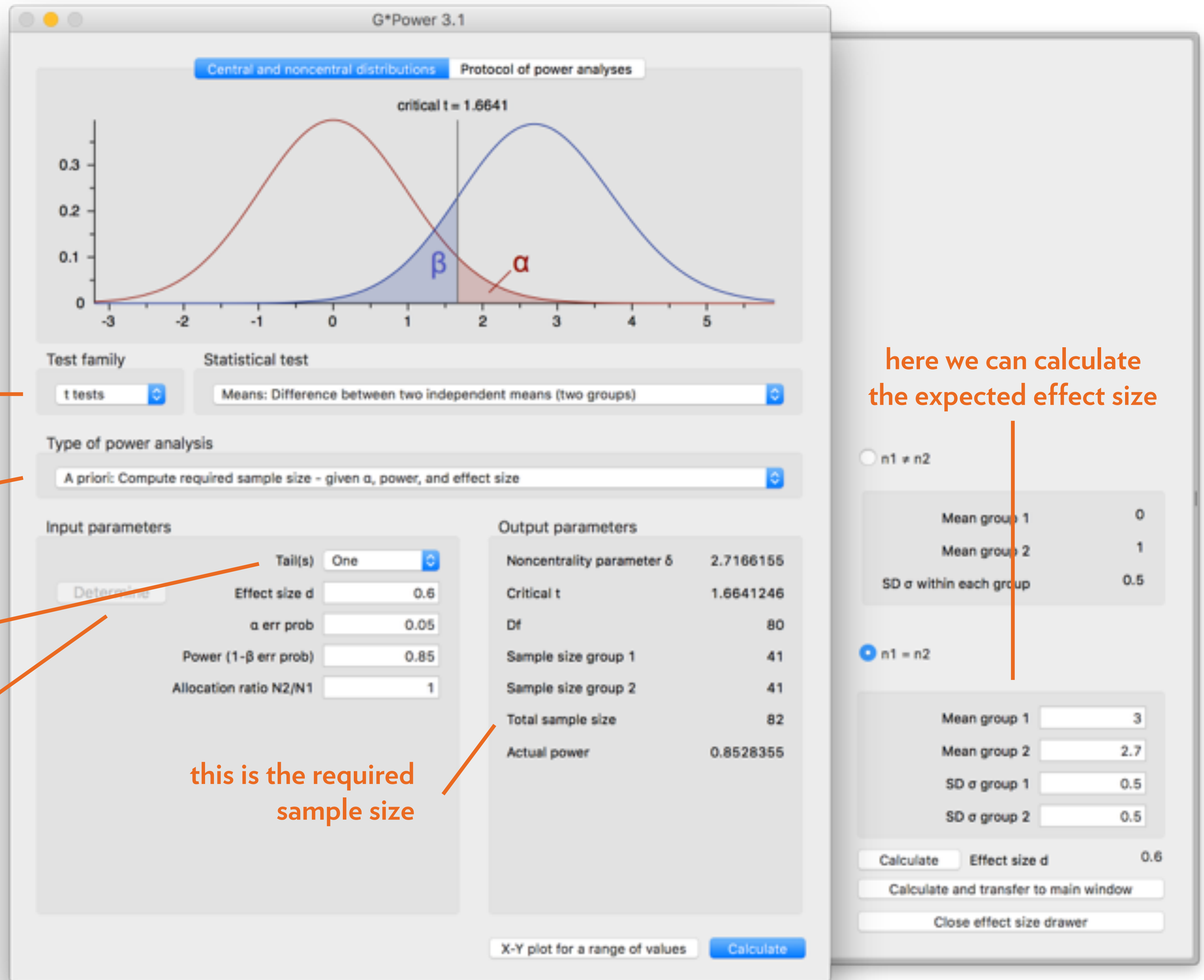
# G*Power demo

this is an independent t-test (see later)

compute the required sample size

we expect our system to be better, so that is a one-tailed test

click here to determine the expected effect size




this is the required sample size

here we can calculate the expected effect size

# G*Power demo

You want to test the combined effect of 6 text sizes and 6 background colors on text readability. You only have money for 150 study participants.

What is the maximum effect size you can find (with 85% power) for a main effects of text size and background color?

What about the interaction effect?

Would it help if you only test 2 sizes and colors?

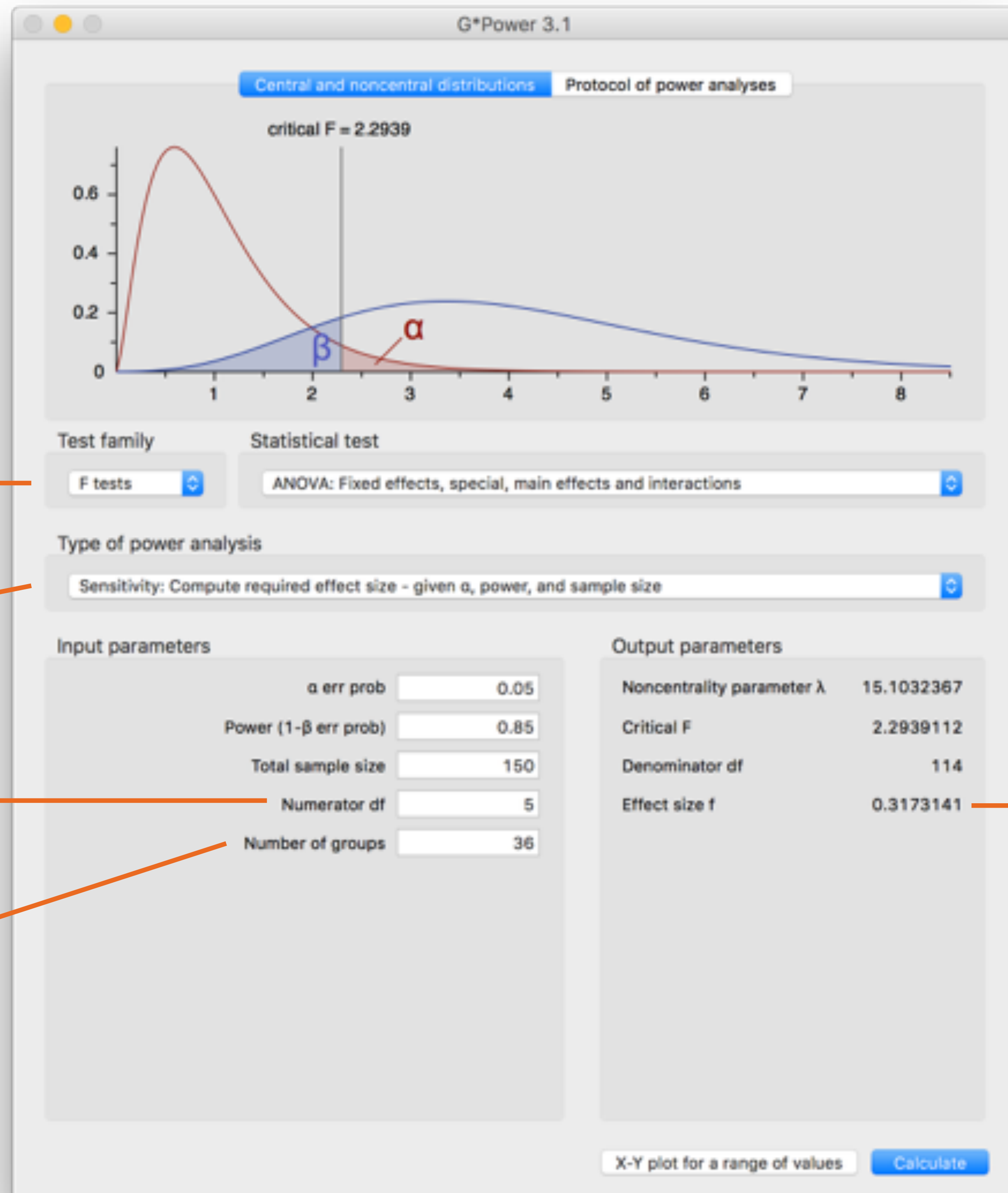
# G*Power demo

this is a factorial ANOVA (see later)

compute the smallest detectable effect, given N

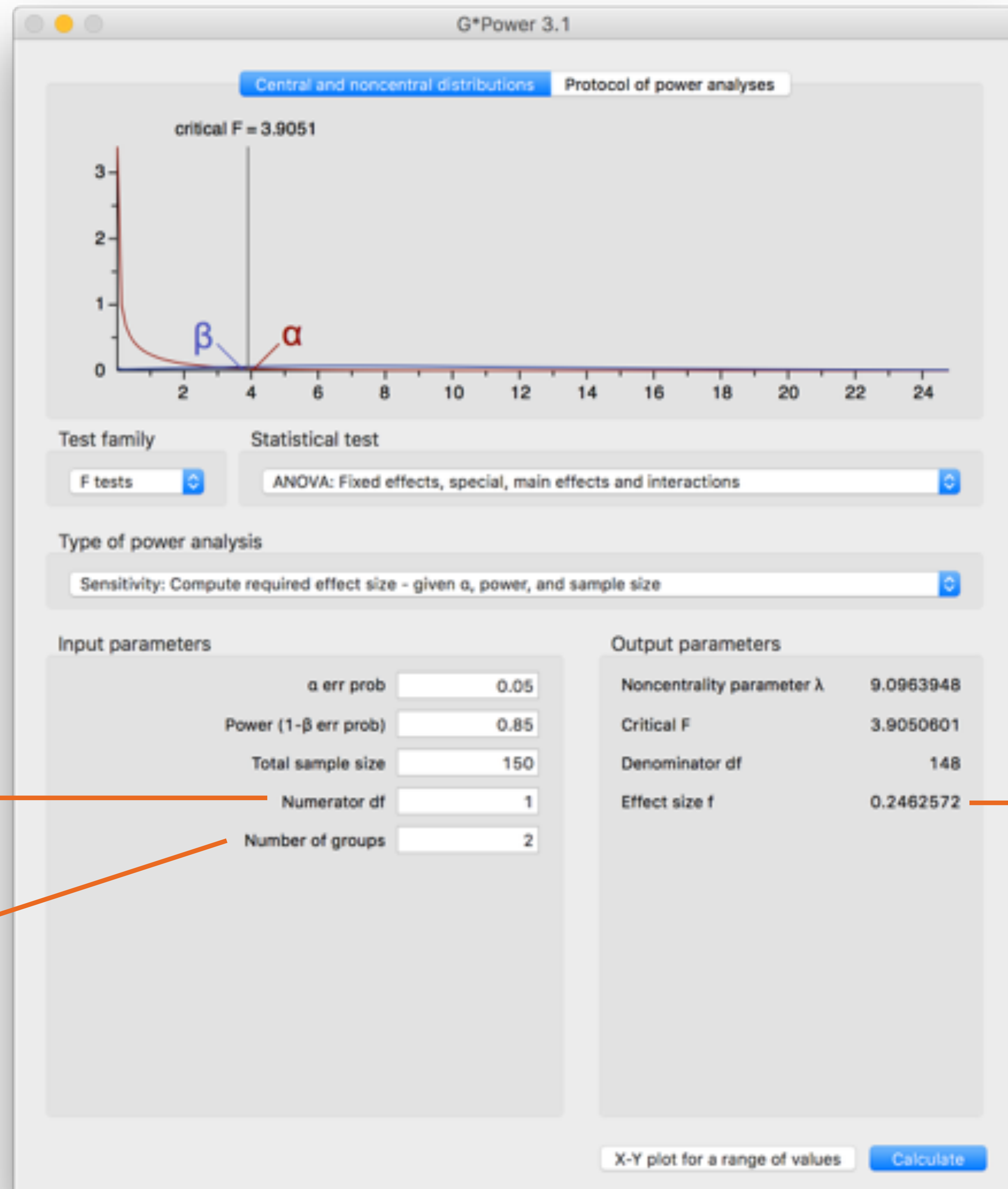
for main effects, we have 5 degrees of freedom (for interactions 25)

we have 6x6 experimental conditions




This is the smallest effect we can find

# G*Power demo




with only 2x2 conditions,
the degrees of freedom is 1

with 2x2 conditions,
this changes to 4

We can now find
a smaller effect!

# Participants

Be aware of **tiny samples** (even when they report significant results)

- Randomization doesn't work well in tiny samples
- Tiny samples fall prey to the "publication bias"
- Due to the "winner's curse", tiny samples overestimate the real effect size

sample from your **target population**

the target population may be **unrestricted**

make sure your sample is **large enough**

# Participants

Population and sampling

conduct a **power analysis** before you run your study

# Testing A vs. B

Experimental manipulations

# Testing A vs. B

What should be the manipulations?

- Choosing interesting versions to test against each other
- Be aware of placebo-effects

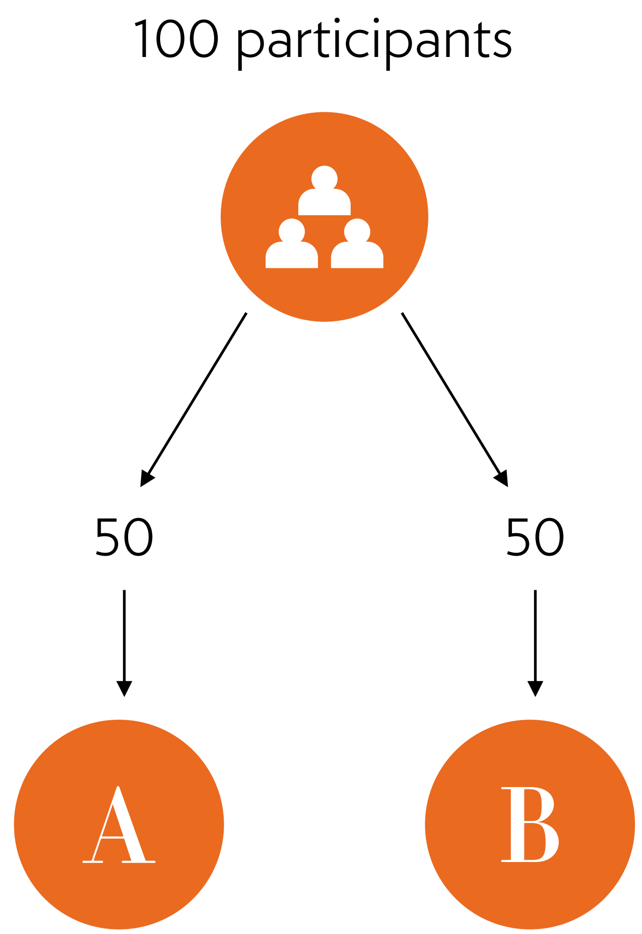
How should participants be assigned to versions?

- Randomization
- Within or between subjects design

# AB Between-subjects

Randomly assign half the participants to A, half to B

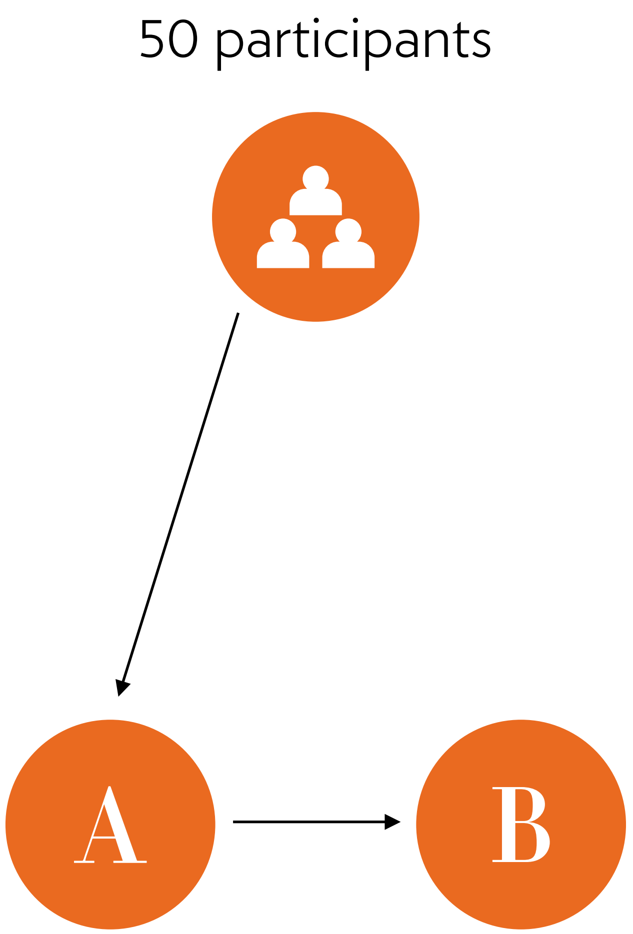
- Realistic interaction
- Manipulation hidden from user
- Many participants needed



# Within-subjects

Give participants A first, then B

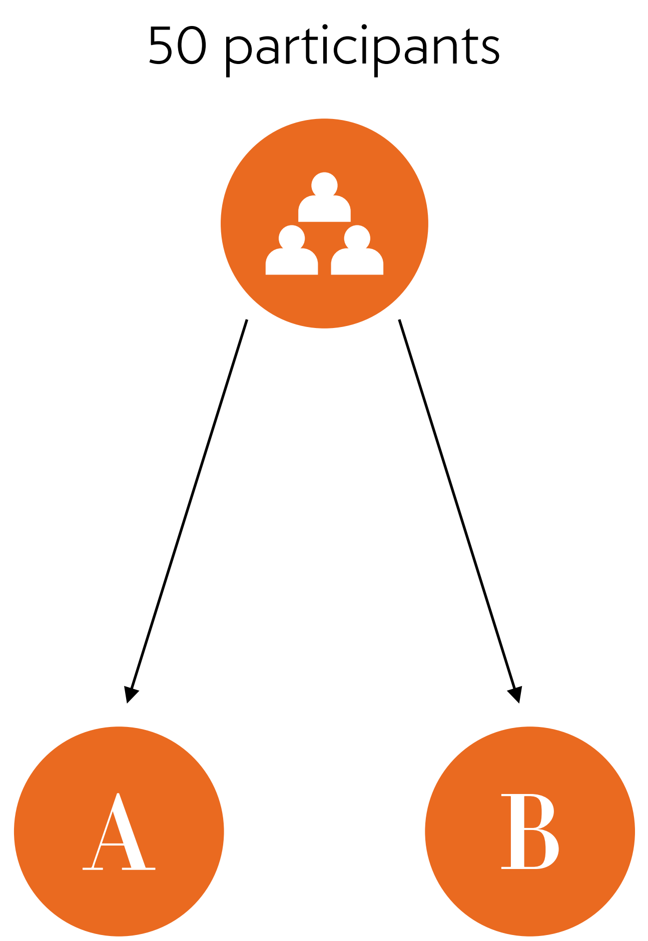
- Remove subject variability
- Participant may see the manipulation (induces demand characteristics)
- Spill-over effect



# Within-subjects

Show participants A and B simultaneously

- Remove subject variability
- Participants can compare conditions
- Not a realistic interaction



# Which one?

Should I do within-subjects or between-subjects?

Use **between-subjects** designs for user experience

- Closer to a real-world usage situation
- No unwanted spill-over effects

Use **within-subjects** designs for psychological research

- Effects are typically smaller
- Nice to control between-subjects variability

# Factorial designs

You can test multiple manipulations in a **factorial design**

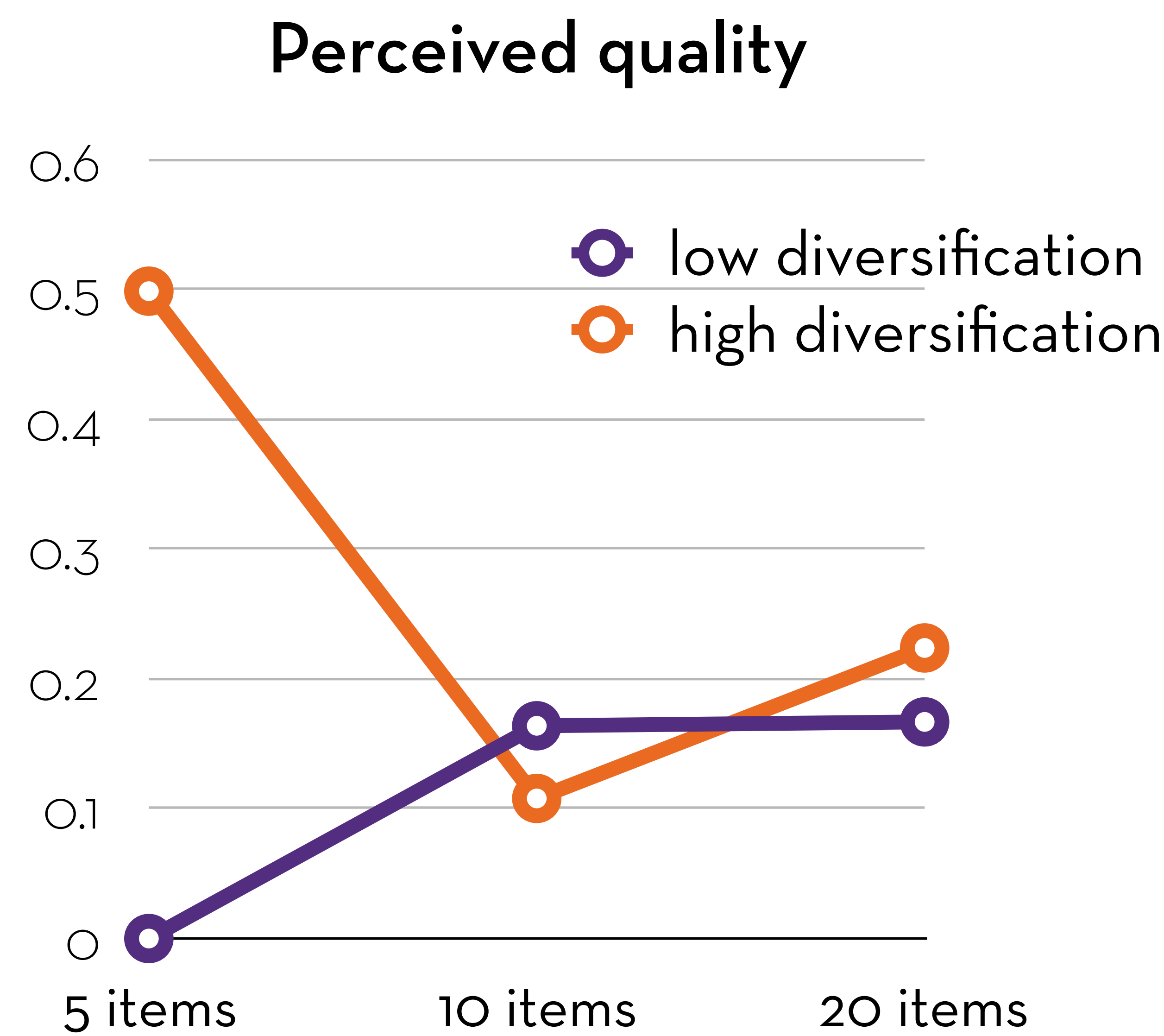
The more conditions, the **more participants** you will need!

| | Low diversity | High diversity |
|---|---|---|
| **5 items** | 5+low | 5+high |
| **10 items** | 10+low | 10+high |
| **20 items** | 20+low | 20+high |

# AB Factorial designs

Allows you to test **interaction effects**

- Is the effect of diversification different per list length?
- Is the effect of list length different for high and low diversification?




Willemsen et al.: "Understanding the Role of Latent Feature Diversification on Choice Difficulty and Satisfaction", submitted to UMUAI

# AB Testing A vs. B

“We were demonstrating our new recommender to a client. They were amazed by how well it predicted their preferences!”

“Later we found out that we forgot to activate the algorithm: the system was giving completely random recommendations.”

(anonymized)

# Testing A vs. B

Experimental manipulations


test against a **reasonable alternative**

**randomize** assignment of conditions

use **between-subjects** for user experience

use **within-subjects** for psychological research

you can test **more than two** conditions

you can test multiple manipulations in a **factorial design**

# Analysis

Statistical evaluation of the results

# Analysis

This section gives a lightning-speed overview of statistical analysis in R:

- regression
- t-test (as a regression)
- ANOVA (as a regression)
- factorial ANOVA (as a regression)
- generalized linear models*
- multi-level generalized linear models*

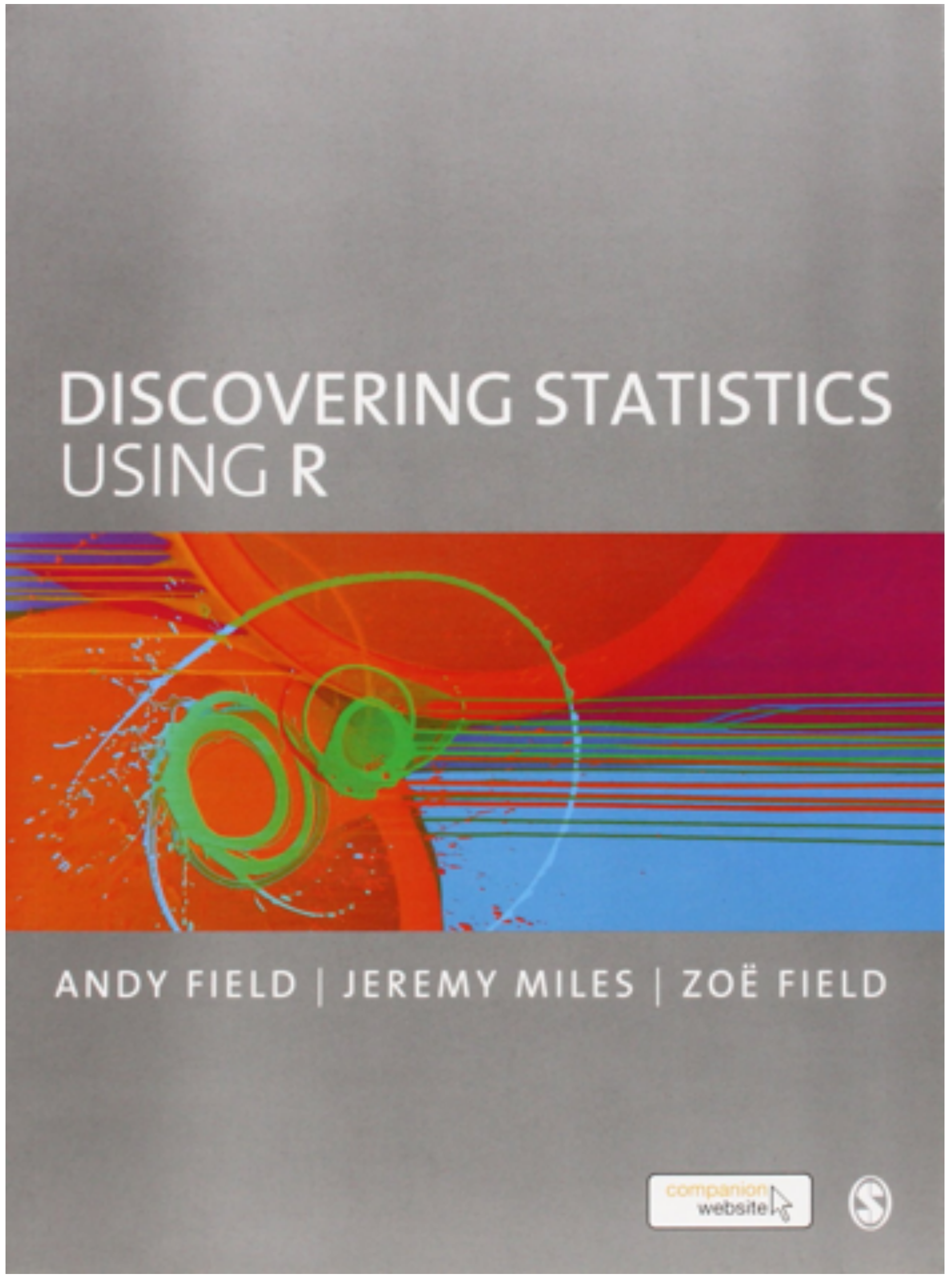
# Analysis

Want to learn more?

- Check out this great book!

Materials and assignments:

- www.usabart.nl/eval
- (free to use, with attribution)



# Example

Knijnenburg et al. (2012): "Inspectability and Control in Social Recommenders", *RecSys'12*

The TasteWeights system uses the overlap between you and your friends' Facebook "likes" to give you music recommendations.

- Friends "weights" based on the overlap in likes w/ user
- Friends' other music likes—the ones that are not among the user's likes—are tallied by weight
- Top 10 is displayed to the user

# Example

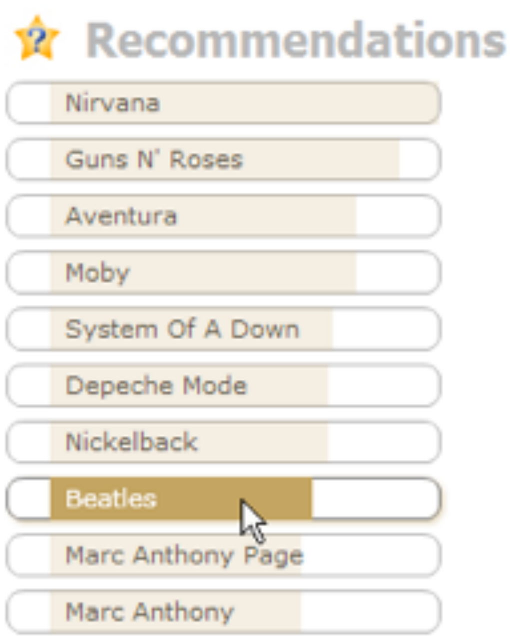
3 control conditions:

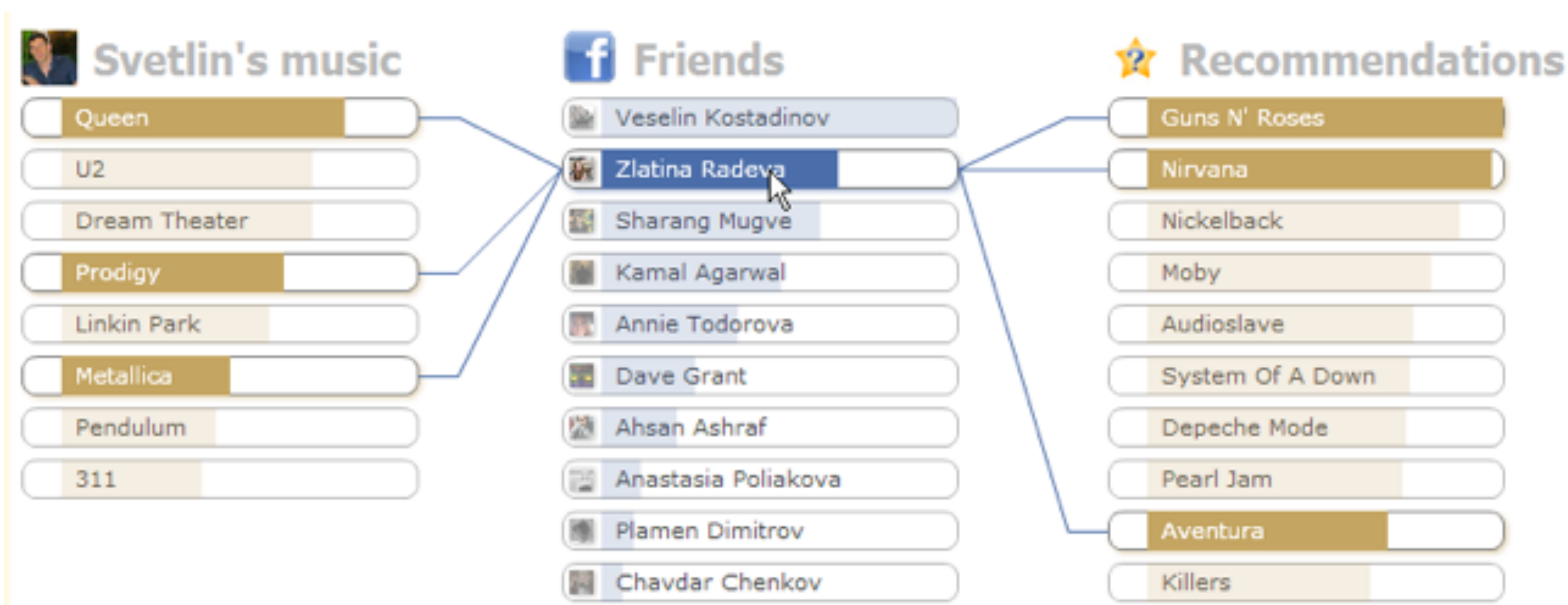
- No control (just use likes)
- Item control (weigh likes)
- Friend control (weigh friends)

# Example

2 inspectability conditions:

- List of recommendations vs. recommendation graph



# Example

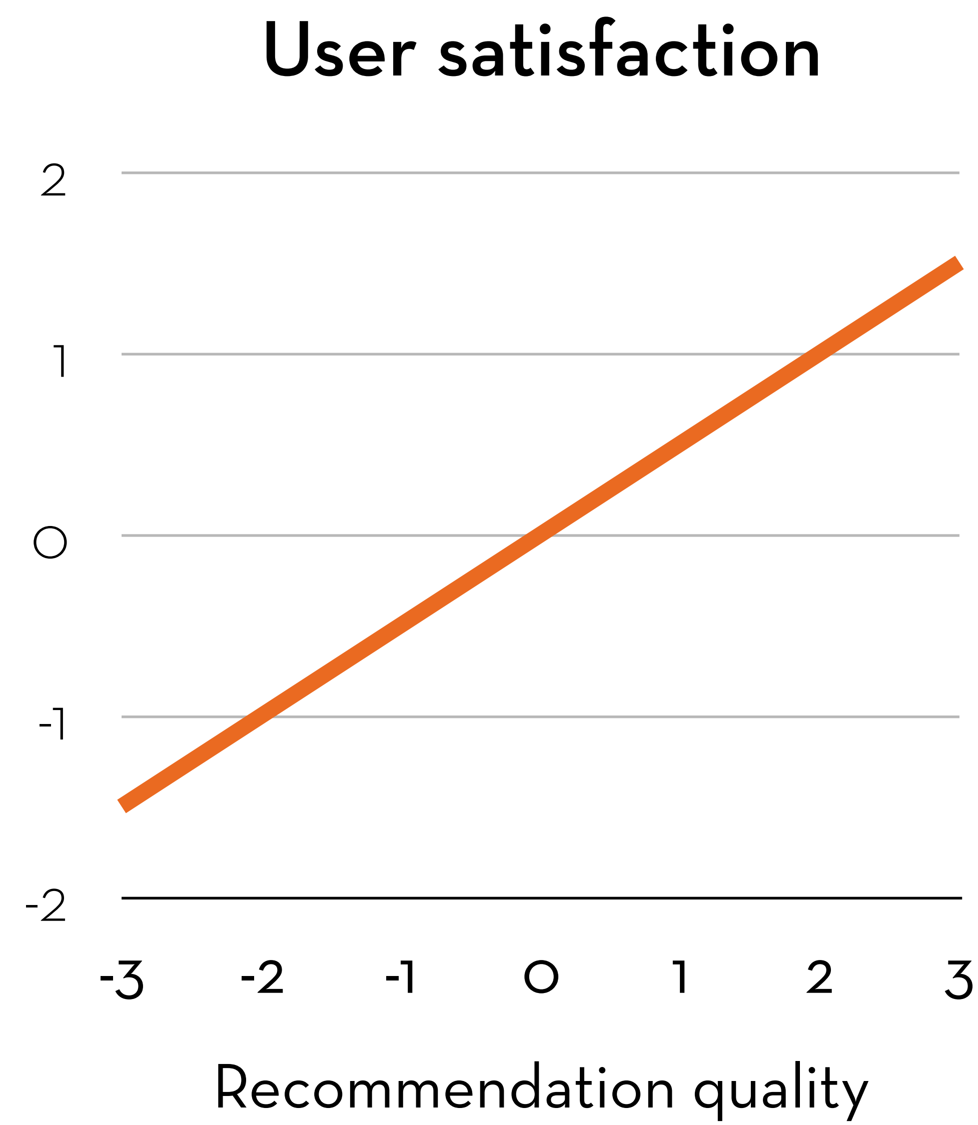
tw.dat, variables:

- **inspectability** and **control** manipulations
- **satisfaction** with the system (sum of seven 5-point scale items)
- **quality** of the recommendations (sum of six items)
- **perceived_control** over the system (four)
- **understandability** of the system (three)
- user music **expertise** (four), propensity to **trust** (three), and **familiarity** (two) with recommenders
- average **rating** of, and number of **known** items in, the top 10
- **time** taken to inspect the recommendations

# Regression

More of X -> more of Y:

Does user satisfaction (Y) increase with perceived recommendation quality (X)?



# Scatterplot

Scatterplot of sales and adverts, with regression line and mean:

```
ggplot(tw, aes(quality, satisfaction))+geom_point()
+geom_smooth(method="lm", color="red", se=F)
+geom_line(aes(y=mean(tw$satisfaction)), color="blue")
```

Result:

- A positive relationship
- Regression line is noticeably different from the mean

# A linear model

Write the regression model:

```
satModel <- lm(satisfaction ~ quality, data = tw)
```

Get the results:

```
summary(satModel)
```

# Output

```
Call:
lm(formula = satisfaction ~ quality, data = tw)

Residuals:
    Min      1Q  Median      3Q     Max 
-15.845  -2.425   1.316   3.477  14.254 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 0.005348   0.473405   0.011    0.991    
quality     0.709846   0.058705  12.092   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.869 on 265 degrees of freedom
Multiple R-squared:  0.3556,	Adjusted R-squared:  0.3531 
F-statistic: 146.2 on 1 and 265 DF,  p-value: < 2.2e-16
```

# Overall fit

The “Multiple R-squared” tells us the percentage of variance in **satisfaction** explained by **quality**

- Seems to be 35.56%

“F-statistic” gives us the improvement of this model

- $F(1, 265) = 146.2$, $p < .001$
- The model makes significantly a better prediction than the mean

# $x \to y$ Model parameters

$Y_i = a + bX_i + e_i$

a: the estimate for "(Intercept)"

The average satisfaction with zero quality (X=0) is 0.005

b: the estimate for "quality"

For a 1-point increase in quality, the model predicts a 0.710-point increase in satisfaction

This effect is significant: $t(265) = 12.092$, $p < .001$

effect size: $\sqrt{(t^2/(t^2+df))} = 0.596$

# Add predictors

Add perceived control and understandability:

satModel2 <- update(satModel, .~. + perceived_control + understandability)

summary(satModel2)

```
                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)        2.35401    0.50722   4.641 5.48e-06 ***
quality            0.40151    0.06054   6.632 1.87e-10 ***
perceived_control  0.74217    0.08400   8.836  < 2e-16 ***
understandability  0.11932    0.08136   1.467    0.144    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.225 on 263 degrees of freedom
Multiple R-squared:  0.5185,	Adjusted R-squared:  0.513 
F-statistic: 94.42 on 3 and 263 DF,  p-value: < 2.2e-16
```

# Add predictors

Compare against the original model:

anova(satModel, satModel2)

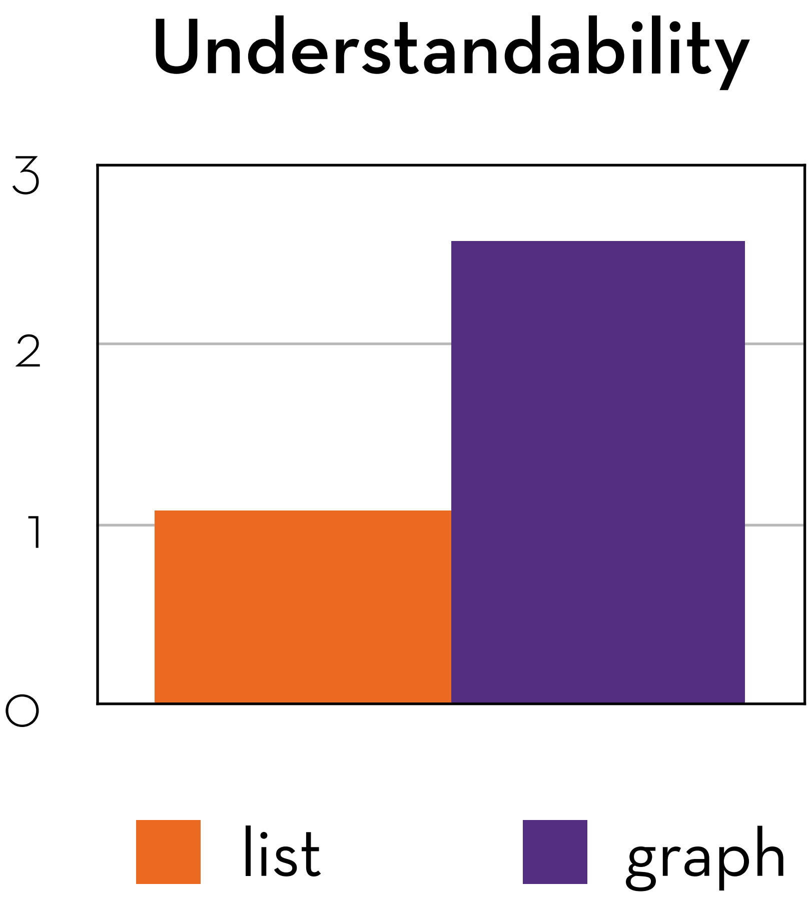
difference in R-squared: .5185 – .3556 = .1629

```
Analysis of Variance Table

Model 1: satisfaction ~ quality
Model 2: satisfaction ~ quality + perceived_control + 
understandability
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)    
1    265 6283.4                                  
2    263 4694.3  2    1589.1 44.514 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

# T-test

Difference between two conditions:

Does inspectability (list vs graph) lead to a different level of understandability?



# T-test = regression!

Regression: $Y = a + bX + e$

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

Your X is a dummy variable:

- $X = 0$ for system A, and 1 for system B
- For system A: $Y = a + b*0 = a$
- For system B: $Y = a + b*1 = a + b$

Parameter b tests the **difference** between system A and B!

# Bar chart

Bar chart with error bars:

> ggplot(tw ,aes(inspectability, understandability)) +stat_summary(fun.y=mean, geom="bar", fill="white", color="black") + stat_summary(fun.data=mean_cl_normal, geom="errorbar", width=0.2)

Result:

- Graph view has higher understandability
- Confidence intervals do not overlap -> probably significant

# x→y Run model

tw$inspectability = relevel(tw$inspectability, ref="listview")

undModel <- lm(understandability ~ inspectability, data = tw)

summary(undModel)

```
                            Estimate Std. Error t value Pr(>|t|)    
(Intercept)                   1.0840     0.2863   3.786 0.000189 ***
inspectabilitygraphview       1.4896     0.4011   3.713 0.000249 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.277 on 265 degrees of freedom
Multiple R-squared:  0.04946,	Adjusted R-squared:  0.04587 
F-statistic: 13.79 on 1 and 265 DF,  p-value: 0.0002494
```

# Model parameters

$Y_i = a + bX_i + e_i$

a: the estimate for "(Intercept)"

The average understandability with list view (X=0) is 1.08

b: the estimate for "inspectabilitygraphview"

The model predicts the understandability of graph view to be 1.49 points higher than list view

This effect is significant: $t(265) = 3.713$, $p < .001$

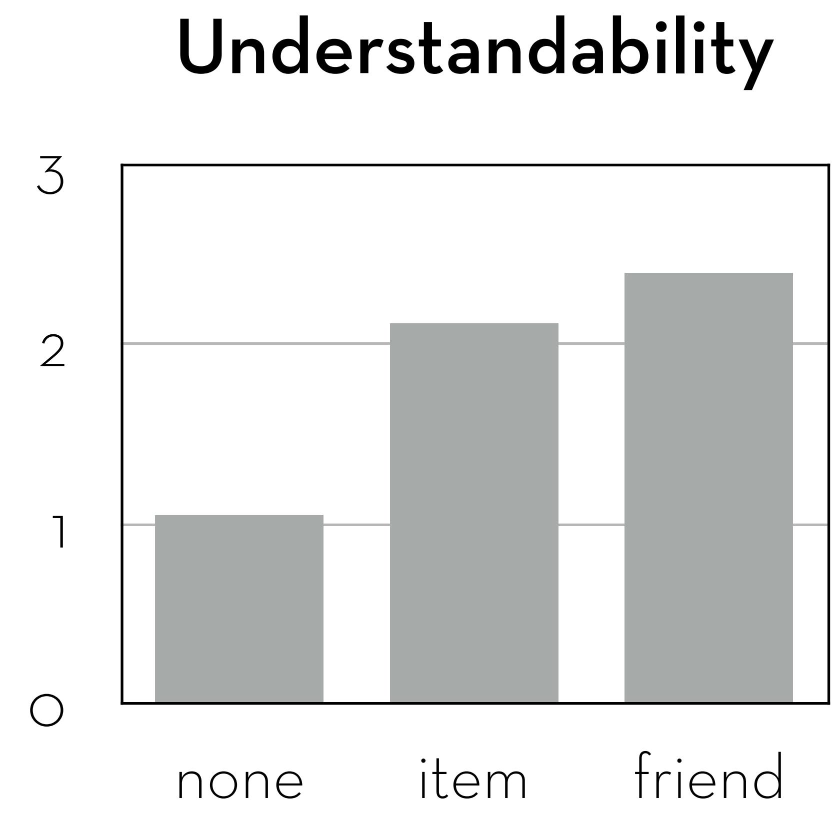
effect size: $\sqrt{(t^2/(t^2+df))} = 0.222$

# ANOVA

Differences between more than two conditions:

> Are there differences in understandability between the three control conditions?

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



# Contrasts

We test if there is **any** effect using an **omnibus test**

- If this test is significant, we know that there is an effect but not where... None and item? None and friend? Item and friend? All of them?

If you have specific hypotheses, test **planned contrasts**

- Otherwise, do post-hoc tests (test all of them)

We are going to run **dummy contrasts**

- These are not optimal (see Andy Field's book for more details), but they are the default method in R

# ANOVA = regression!

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

# ANOVA = regression!

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

# Plotting

Line plot with error bars:

```
ggplot(tw, aes(control,understandability)) +
stat_summary(fun.y=mean, geom="line", aes(group=1)) +
stat_summary(fun.data=mean_cl_normal,
geom="errorbar", width = 0.2)
```

Result:

– item and friend seem to have higher somewhat understandability

# Run the ANOVA

Run the ANOVA:

- undModel2 <- lm(understandability~control, data=tw)
- summary.aov(undModel2)
- this is the omnibus test (there is "some" difference between control conditions)

```
             Df Sum Sq Mean Sq F value Pr(>F)  
control       2   93.3   46.65   4.246 0.0153 *
Residuals   264 2900.1   10.99                 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

# Run the ANOVA

Get the regression results:

summary(undModel2)

tests item vs. none, and friend vs. none

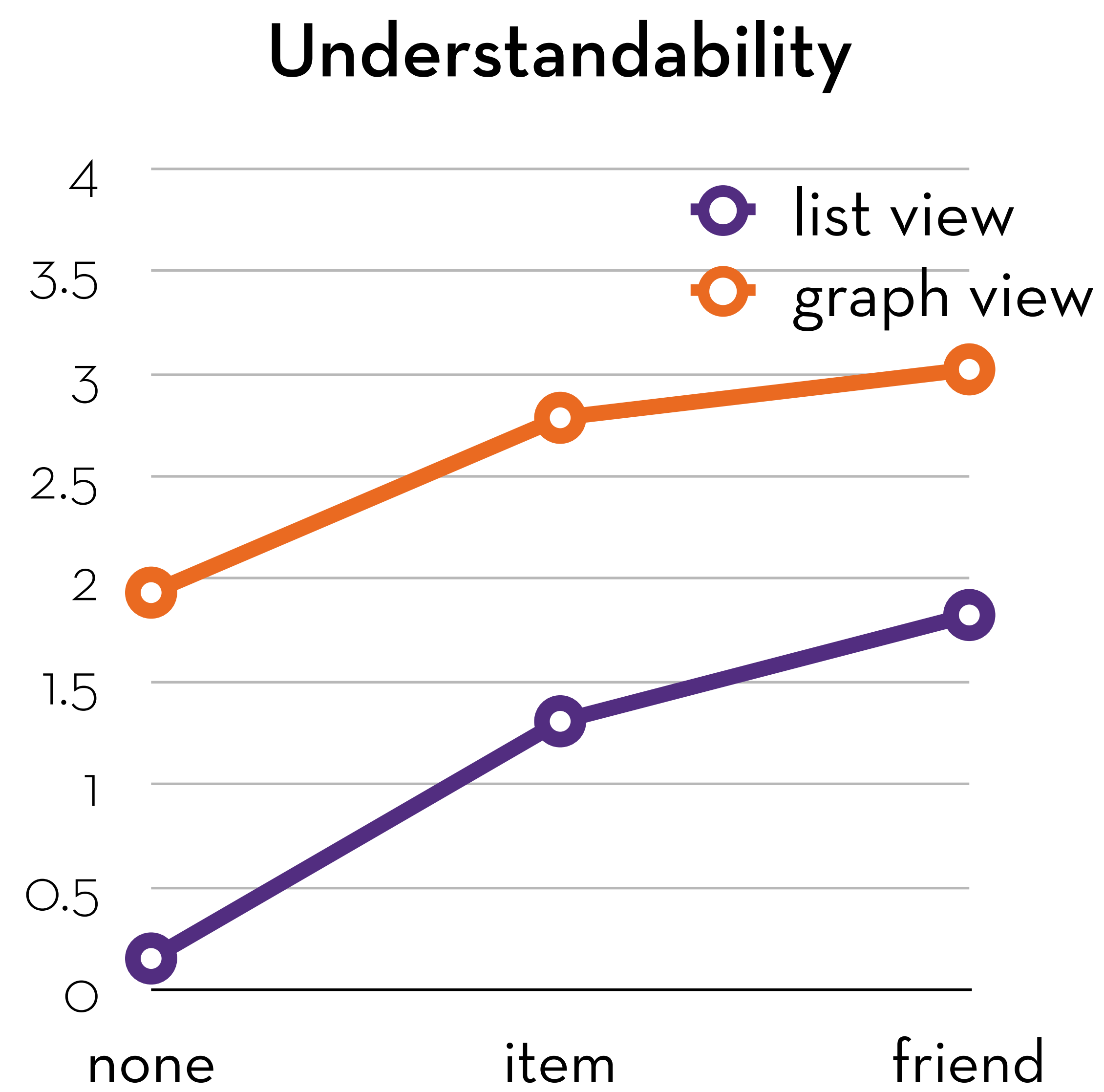
```
              Estimate Std. Error t value Pr(>|t|)
(Intercept)     1.0435     0.3455   3.020  0.00278 **
controlitem     1.0728     0.4971   2.158  0.03183 *
controlfriend   1.3610     0.4928   2.762  0.00615 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.314 on 264 degrees of freedom
Multiple R-squared:  0.03117,Adjusted R-squared:  0.02383
F-statistic: 4.246 on 2 and 264 DF,  p-value: 0.01531
```

# Factorial ANOVA

Two manipulations at the same time:

> What is the combined effect of control and inspectability on understandability?

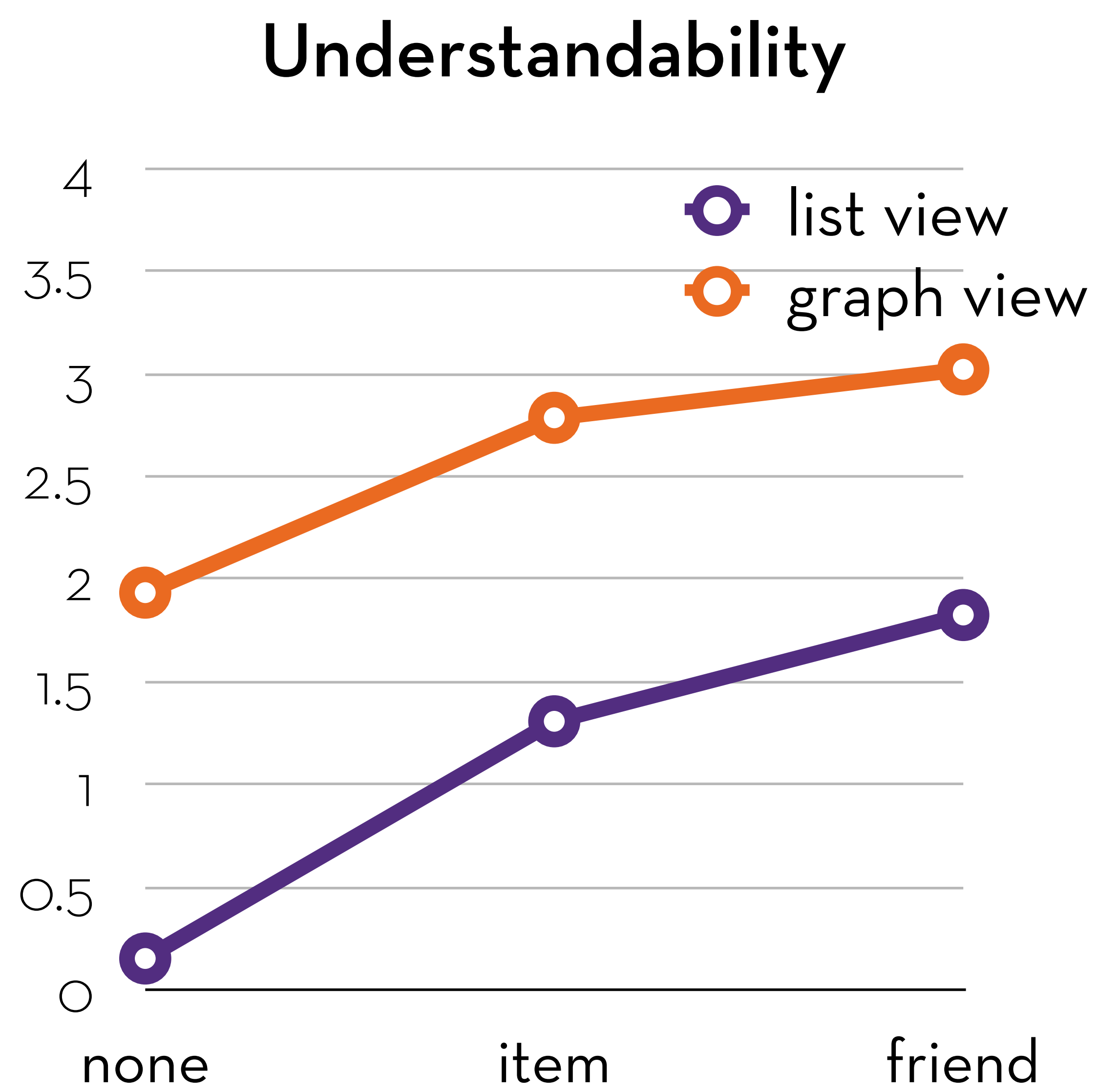
Test for the interaction effect!



# Factorial ANOVA

Parallel lines = no interaction effect

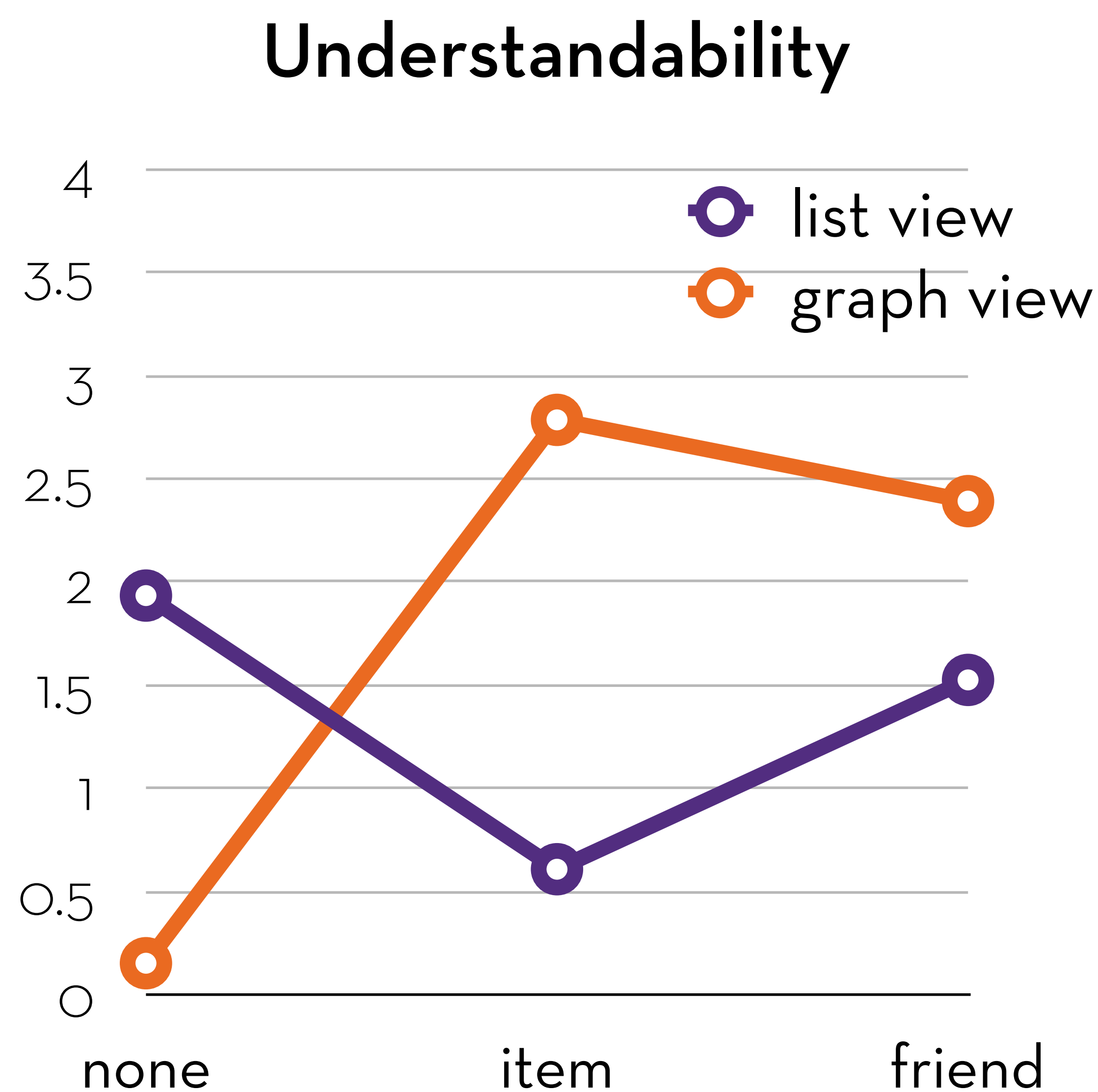
- Effect of control is the same for list and graph view
- Effect of inspectability is the same for none, item, and friend control



# Factorial ANOVA

Non-parallel lines = interaction effect

- Effect of control **differs** for list and graph view
- Effect of inspectability **differs** for none, item, and friend control



# ...as a regression

$Y_i = a + b_1X_{1i} + b_2X_{2i} + b_3X_{3i} + b_4X_{1i}X_{2i} + b_5X_{1i}X_{3i} + e_i$

- View (div.): $X_1 = 1$ for graph, $X_1 = 0$ for list
- Control: $X_2 = 1$ for item control, $X_3 = 1$ for friend control (both are 0 for no control)
- $b_1$: difference between graph and list (for no control only)
- $b_2$: difference between none and item (for list view only)
- $b_3$: difference between none and friend (for list view only)

# ...as a regression

$Y_i = a + b_1X_{1i} + b_2X_{2i} + b_3X_{3i} + b_4X_{1i}X_{2i} + b_5X_{1i}X_{3i} + e_i$

- $b_4$: extra difference between list and graph for item, or extra difference between none and item for graph view
- $b_5$: extra difference between list and graph for friend, or extra difference between none and friend for list graph
- $b_4$ and $b_5$ measure the interaction effect
- $b_1$, $b_2$ and $b_3$ are uninterpretable without $b_4$ and $b_5$

# Double line plot

Double line plot with error bars:

```
ggplot(tw, aes(control, understandability, color = inspectability)) + stat_summary(fun.y = mean, geom = “line”, aes(group = inspectability)) + stat_summary(fun.data = mean_cl_normal, geom = “errorbar”, width = 0.2)
```

Result:

- Lines are parallel; probably no interaction effect

# Run the ANOVA

Run the ANOVA:

undModel3 <- lm(understandability~control*inspectability, data=tw)

summary.aov(undModel3)

```
                        Df Sum Sq Mean Sq F value   Pr(>F)    
control                  2   93.3   46.65   4.430 0.012829 *  
inspectability           1  147.7  147.72  14.028 0.000222 ***
control:inspectability   2    3.9    1.94   0.184 0.831962    
Residuals              261 2748.5   10.53                     
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

# Run the ANOVA

Get the regression results:

summary(undModel3)

```
                                      Estimate Std. Error t value Pr(>|t|)
(Intercept)                             0.1522     0.4785   0.318  0.75070
controlitem                             1.1555     0.7064   1.636  0.10307
controlfriend                           1.6739     0.6766   2.474  0.01400 *
inspectabilitygraphview                 1.7826     0.6766   2.634  0.00893 **
controlitem:inspectabilitygraphview    -0.3031     0.9757  -0.311  0.75633
controlfriend:inspectabilitygraphview  -0.5854     0.9652  -0.607  0.54469
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.245 on 261 degrees of freedom
Multiple R-squared:  0.08181,   Adjusted R-squared:  0.06422
F-statistic: 4.651 on 5 and 261 DF,  p-value: 0.0004388
```

# Analysis

Good job!

You now have the stats knowledge of about 80% of the people in this field!

Coincidentally, we worked ourselves through 50% of Andy Field's book

Disco (my bunny rabbit) is impressed!

# If Y is not normal…

Standard tests assume that the dependent variable (Y) is an continuous, unbounded, normally distributed interval variable

- Continuous: variable can take on any value, e.g. 4.5 or 3.23 (not just whole numbers)
- Unbounded: range of values is unlimited (or at least does not stop abruptly)
- Interval: differences between values are comparable; is the difference between 1 and 2 the same as the difference between 3 and 4?

# If Y is not normal…

Most behavioral measures are not normal!

- Number of clicks (discrete, zero-bounded)
- Time, money (zero-bounded)
- Ratings (1-5)
- Decisions (yes no)

# Logistic regression

Linear regression:

$Y_i = a + b_1X_{1i} + b_2X_{2i} + ... + b_kX_{ki} + e_i$

What if Y is **binary** (0 or 1)?

We can try to predict the **probability** of Y=1 — P(Y)

However, this probability is a number between 0 and 1

For linear regression, we want an unbounded linear Y!

Can we find some transformation that allows us to do this?

Yes: $P(Y) = 1 / (1+e^{-U})$

# Logistic regression

$P(Y) = 1 / (1+e^{-U})$

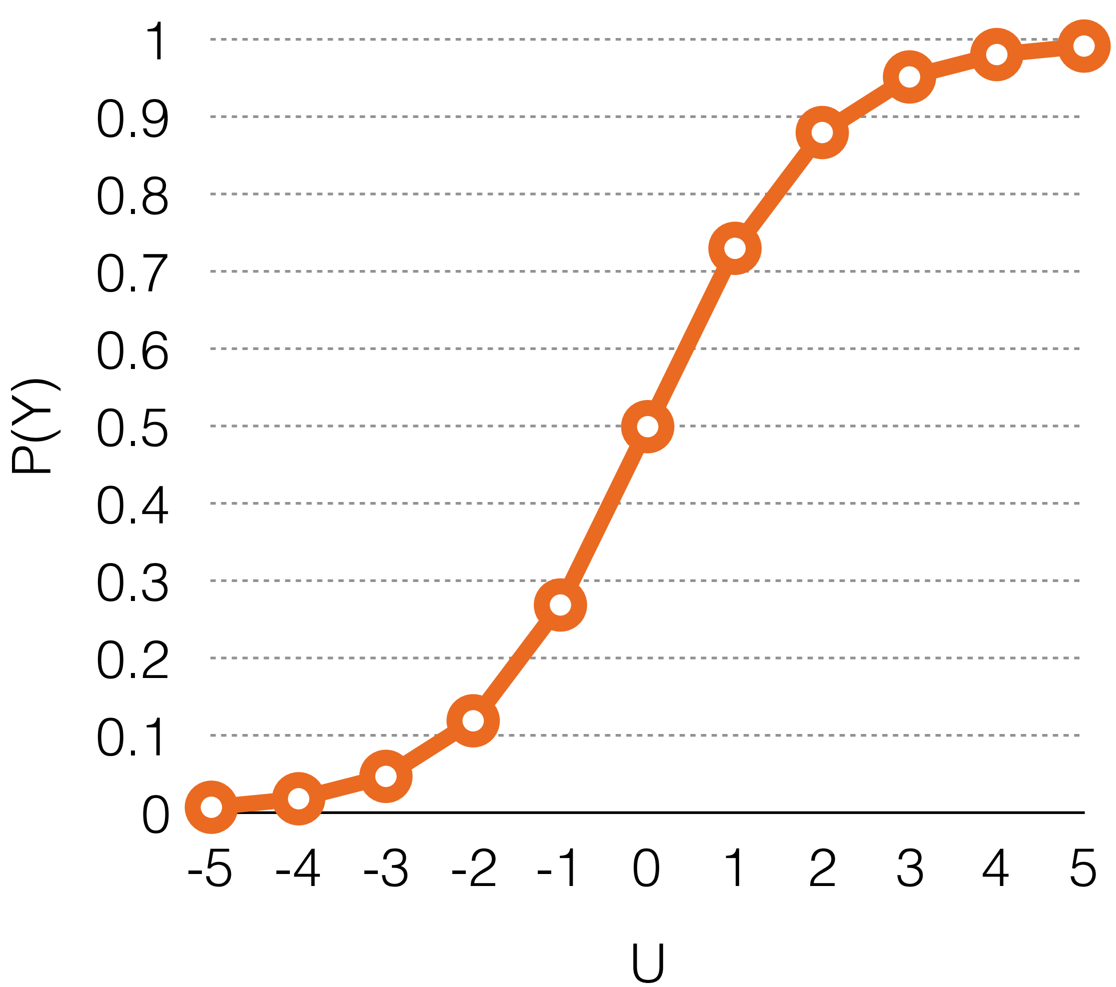
Conversely:

$U = \ln(P(Y)/(1-P(Y)))$

Interpretation:

$P(Y)/(1-P(Y))$ is the **odds** of Y

Therefore, U is the log odds, or **logit** of Y



# Logistic regression

Since U is unbounded, we can treat it as our regression outcome:

$$U_i = \ln(P(Y_i)/(1-P(Y_i))) = Y_i = a + b_1X_{1i} + b_2X_{2i} + \ldots + b_kX_{ki} + e_i$$

We can always transform it back to $P(Y_i)$ if we want to:

$$P(Y_i) = 1 \,/\, \left(1+e^{-(a + b1X1i + b2X2i + \ldots + bkXki + ei)}\right)$$

# Coefficients

How to interpret the b coefficients?

- b is the increase in U for each increase of X
- b is the increase in $\ln(P(Y)/(1-P(Y)))$ for each increase in X
- $e^b$ is the ratio of $P(Y)/(1-P(Y))$ for each increase in X
- $e^b$ is the **odds ratio**

# Create a variable

Objective: Our recommender system is obviously less useful if the participant already knew all ten recommendations.

New variable: “allknown”

```
tw$allknown <- tw$known == 10
```

# Run the regression

Run the logistic regression:

allknownModel <- glm(allknown~expertise, family=binomial, data=tw)

summary(allknownModel)

```
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  -1.0529     0.2801  -3.759 0.000171 ***
expertise     0.1254     0.0506   2.479 0.013184 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

# Run the regression

```
              Estimate Std. Error z value Pr(>|z|)    
(Intercept)   -1.0529     0.2801  -3.759 0.000171 ***
expertise      0.1254     0.0506   2.479 0.013184 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Interpretation:

Probability that a user with expertise = 0 already knows all recommendations: $1/(1+e^{-(-1.0529)}) = 0.259$

Probability that a user with expertise = 4 already knows all recommendations: $1/(1+e^{-(-1.0529+4*0.1254)}) = 0.366$

# Run the regression

```
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  -1.0529     0.2801  -3.759 0.000171 ***
expertise     0.1254     0.0506   2.479 0.013184 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Interpretation:

Odds ratio: $e^{0.1254} = 1.134$.

"The odds of already knowing all the recommendations are predicted to be 13.4% higher for participants with a 1-point higher level of music expertise."

# Poisson regression

What if Y is a (non-normal) **count variable**?

Example: number of recommendations not yet known:

```
tw$notknown <- 10 - tw$known
ggplot(tw, aes(notknown)) + geom_histogram()
```

Doesn't look normal!

This is because notknown is a count variable!

Can we find some transformation that makes this work?

Yes: $Y = e^U$

# Coefficients

How to interpret the b coefficients?

- b is the increase in U for each increase of X
- b is the increase in the **log rate** of Y for each increase in X
- $e^b$ is the ratio of rate Y for each increase in X
- $e^b$ is the **rate ratio**

# Run the regression

Run the Poisson regression:

notknownModel <- glm(notknown~expertise +inspectability, family=quasipoisson, data=tw)

summary(notknownModel)

```
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)               0.79324    0.13642   5.815 1.75e-08 ***
expertise                -0.04967    0.02456  -2.023  0.04412 *
inspectabilitygraphview -0.37482    0.13942  -2.688  0.00763 **
```

# $x \to y$ Run the regression

```
                          Estimate Std. Error t value Pr(>|t|)    
(Intercept)                0.79324    0.13642   5.815 1.75e-08 ***
expertise                 -0.04967    0.02456  -2.023  0.04412 *  
inspectabilitygraphview   -0.37482    0.13942  -2.688  0.00763 ** 
```

Interpretation:

Predicted # of recs not known by a user with expertise = 0 in the list view condition: $e^{0.793} = 2.21$

Predicted # of recs not known by a user with expertise = 4 in the graph view condition: $e^{0.793+4^{*}-0.050-0.375} = 1.24$

# $x \to y$ Run the regression

```
                          Estimate Std. Error t value Pr(>|t|)    
(Intercept)                0.79324    0.13642   5.815 1.75e-08 ***
expertise                 -0.04967    0.02456  -2.023  0.04412 *  
inspectabilitygraphview -0.37482    0.13942  -2.688  0.00763 ** 
```

Interpretation:

Rate ratio: $e^{-0.050} = 0.952$

"Controlling for the effect of inspectability condition, participants with a 1-point higher level of music expertise are predicted to have 4.8% fewer unknown recommendations."

# Correlated errors

Standard regression requires **uncorrelated errors**

This is not the case when...

...you have repeated measurements of the same participant (e.g. you measured 5 task performance times per participant, for 60 participants)

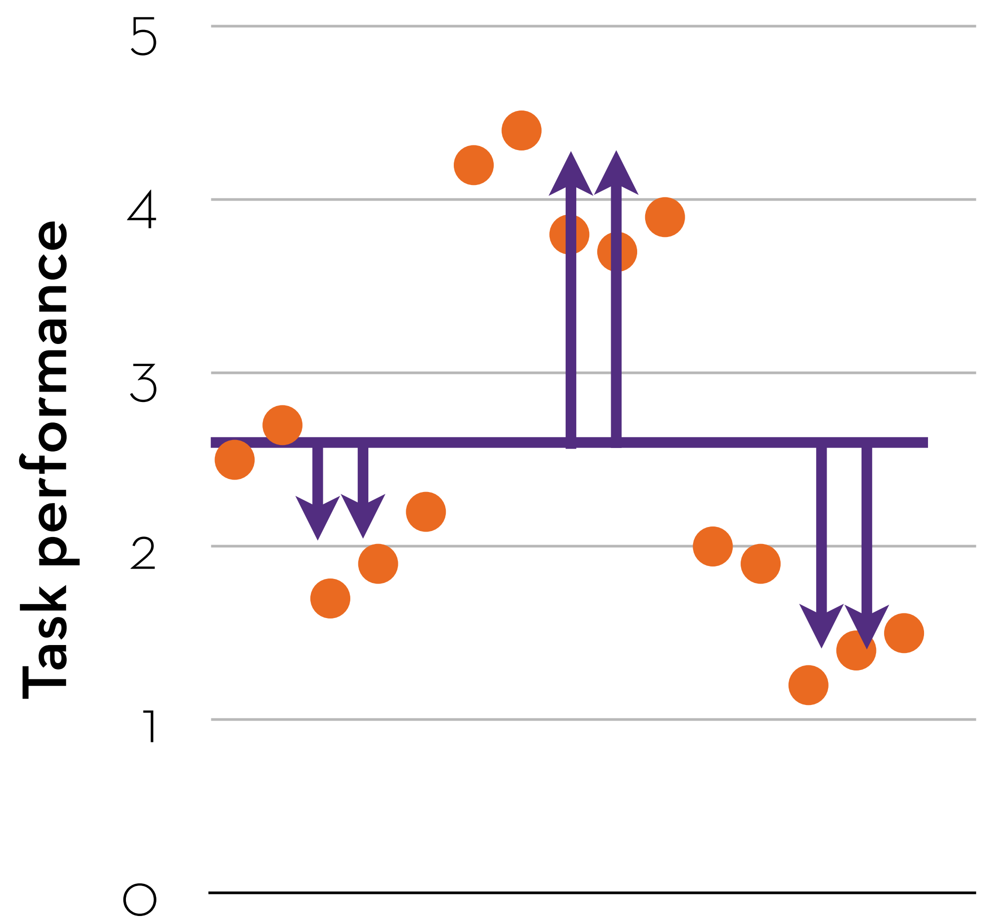
...participants are somehow related (e.g. you measured the performance of 5 group members, for 60 groups)

# $x \to y$ Correlated errors

Consequence: errors are correlated

There will be a user-bias (and maybe an task-bias)

Solution: use **linear models effects models** to introduce **random effects**



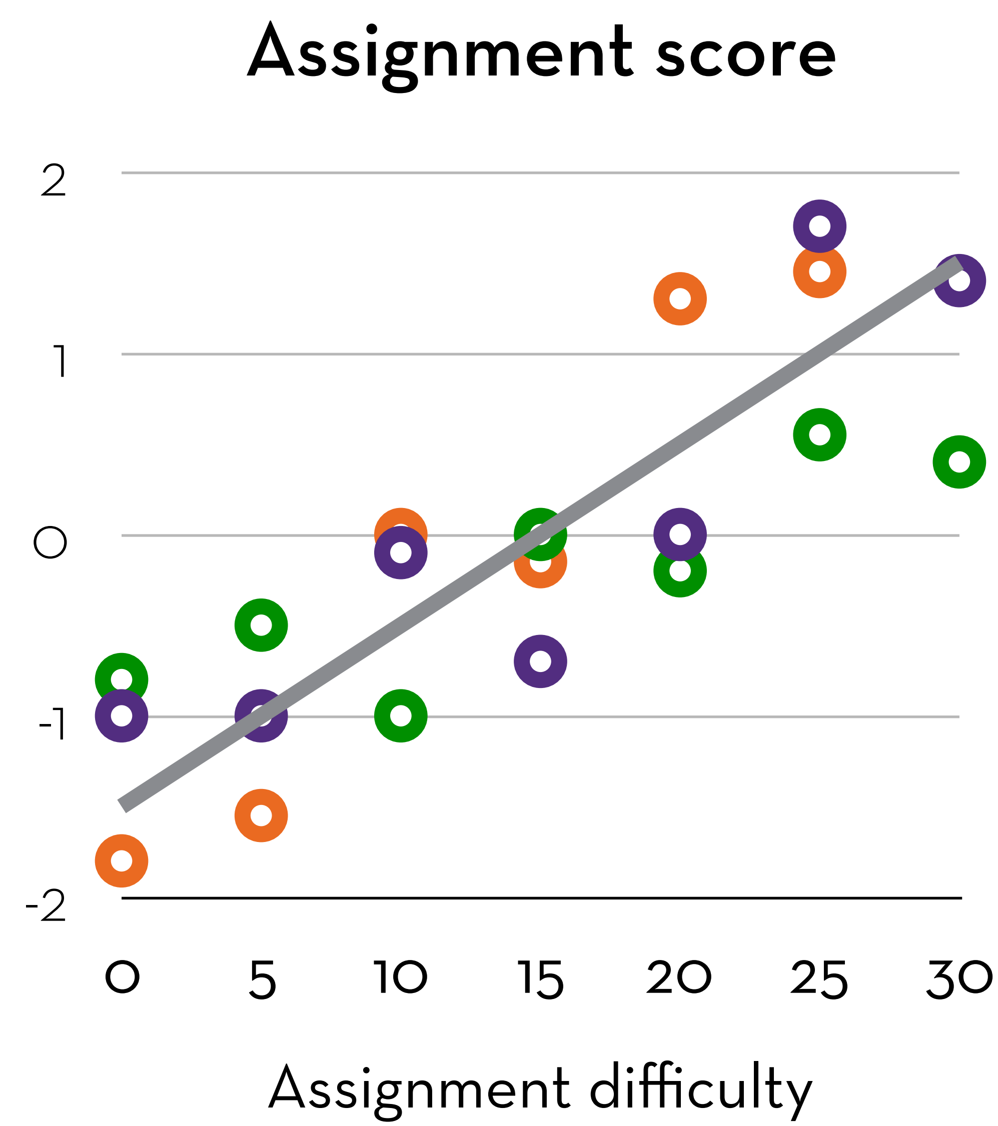
# Random effects

Data from three participants:

Adam, Brian, Chen

Fixed intercept + slope

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



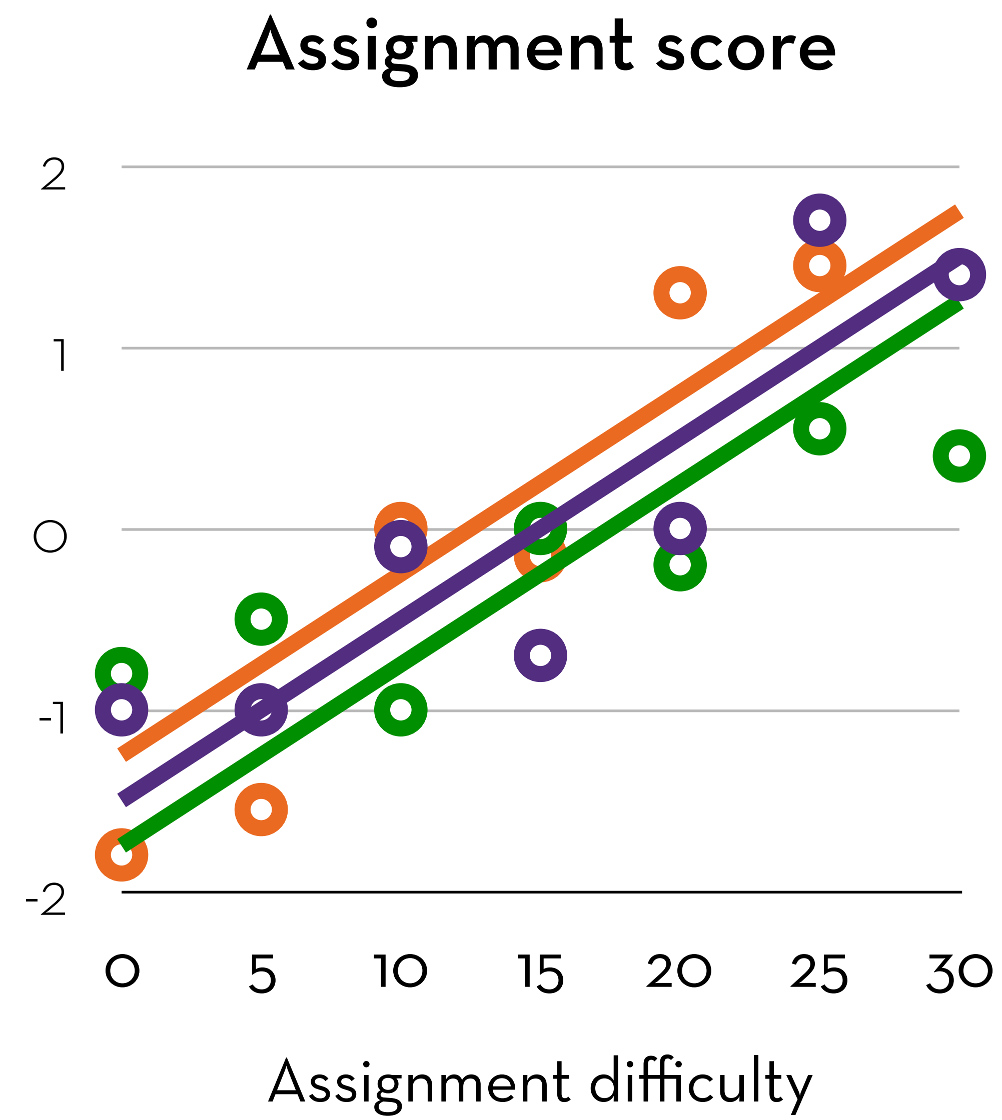
# Random effects

Data from three participants:

Adam, Brian, Chen

Different intercept + fixed slope

$Y_i = a + b_1 X_{diff} + b_2 X_{brian} + b_3 X_{chen} + e_i$



# Random effects

Data from **many** participants

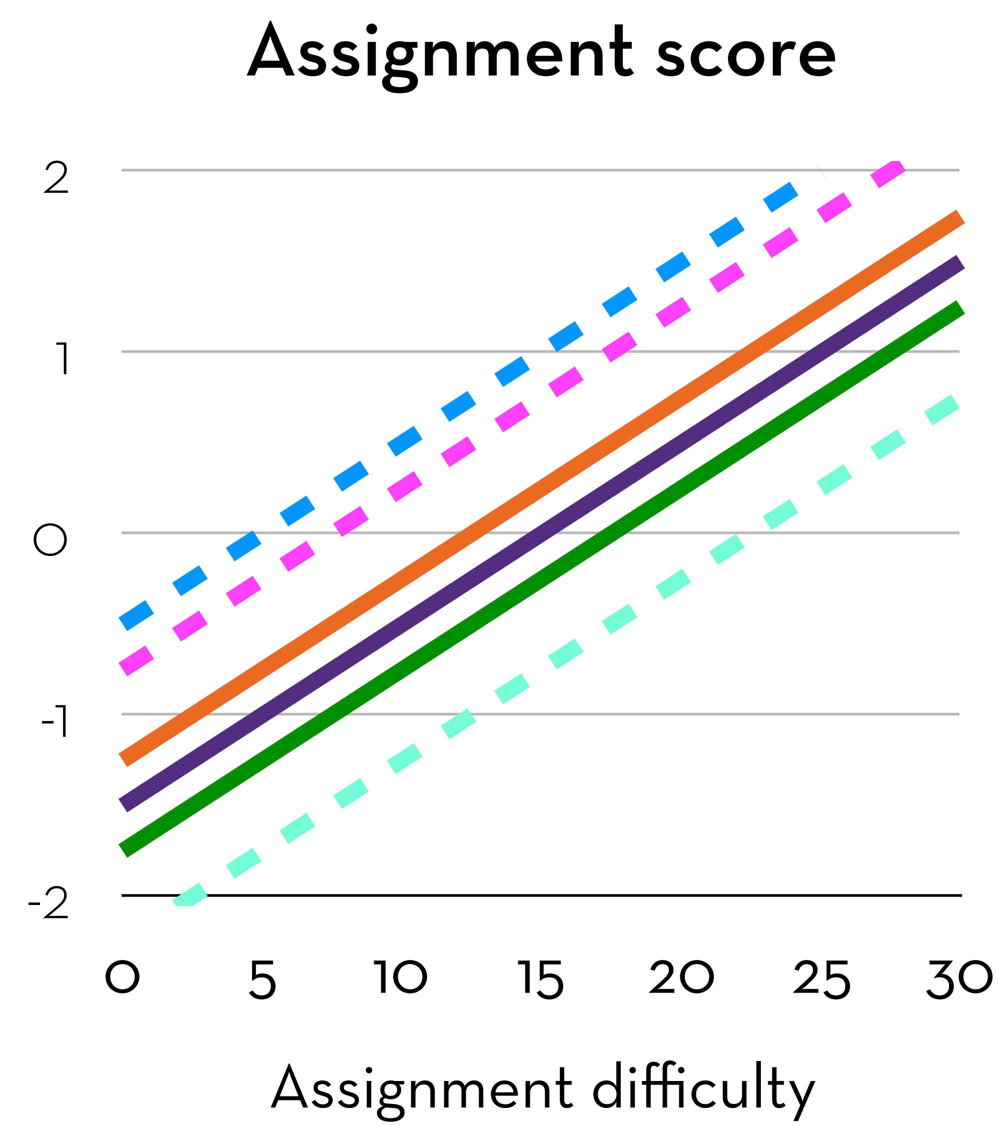
**Random** intercept + fixed slope

$$Y_{ip} = a_p + b_1 X_{diff} + e_{ip}$$

where $a_p = a + u_p$

$u_p$ differs per participant!

we fit a single parameter for it (variance)



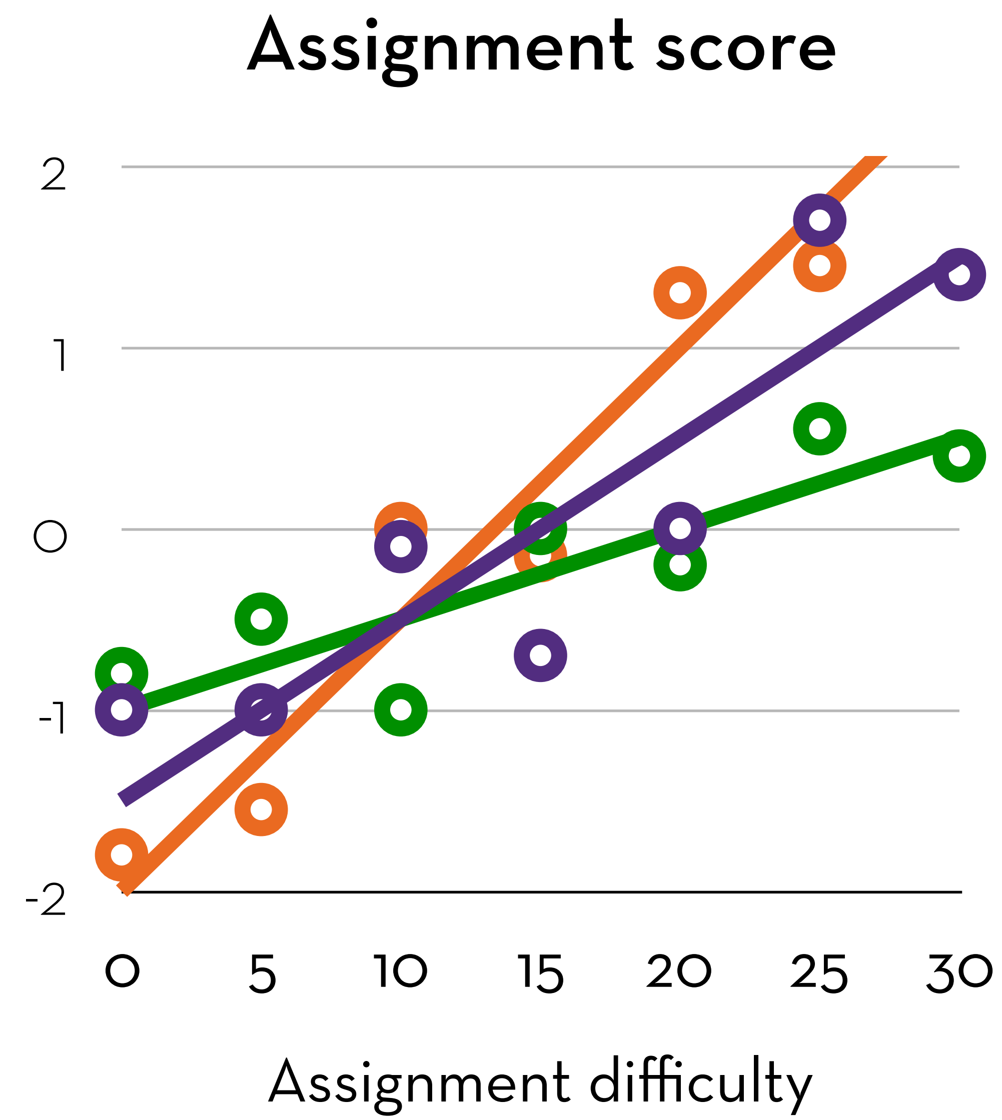
# Random effects

Data from 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

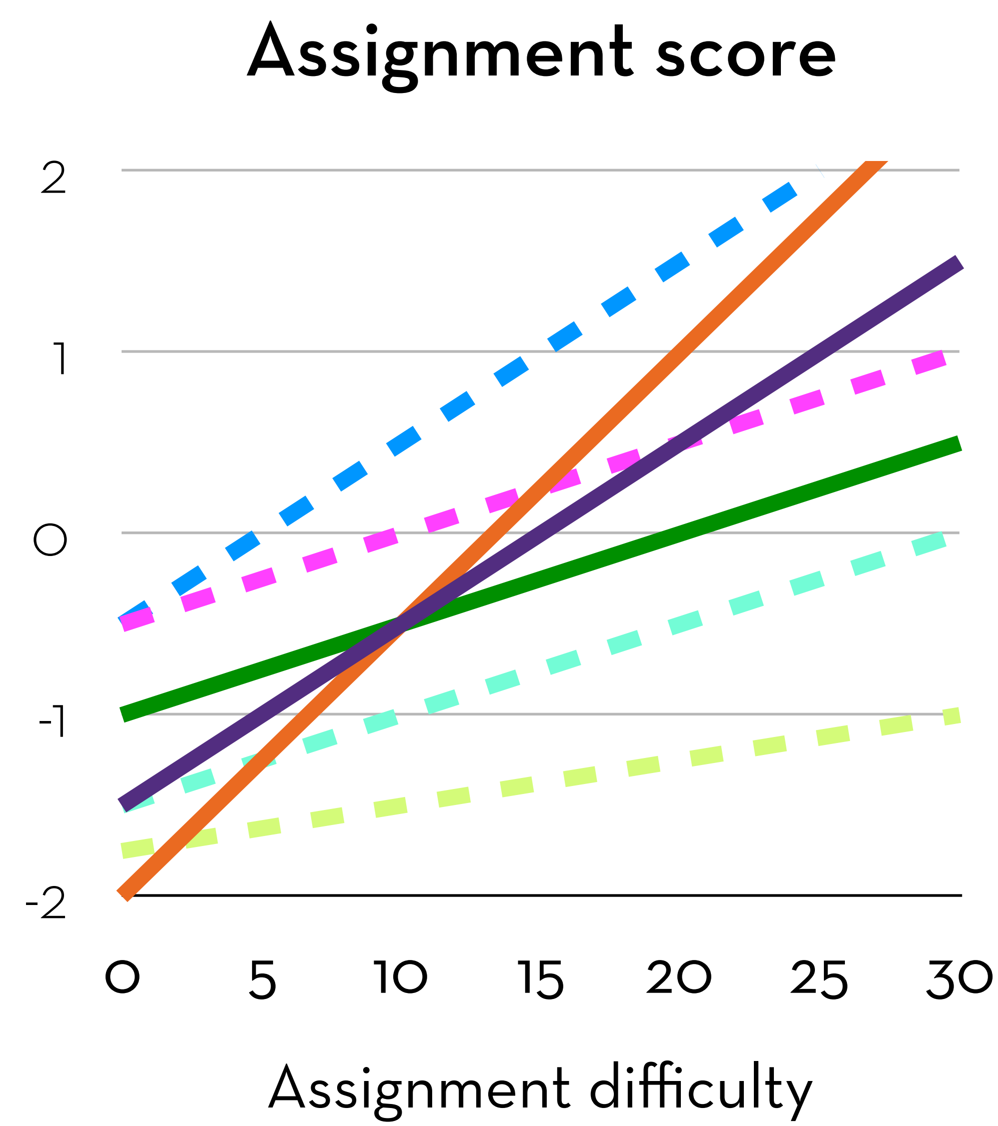
**Random** intercept +
**random** slope

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

where $a_p = a + u_p$

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

Both $u_p$ and $v_p$ differ per participant!



# Example

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

# Example

Applause
May we track your location?
Reveal my current:
No
Yes
Applause
May we know what apps you have?
No
Yes
Applause
May we track *where* you use your apps?
No
Yes
Applause
May we track *when* you use your apps?
No
Yes

# Example


Context data first

Demographic data first

# Example

5 justification types

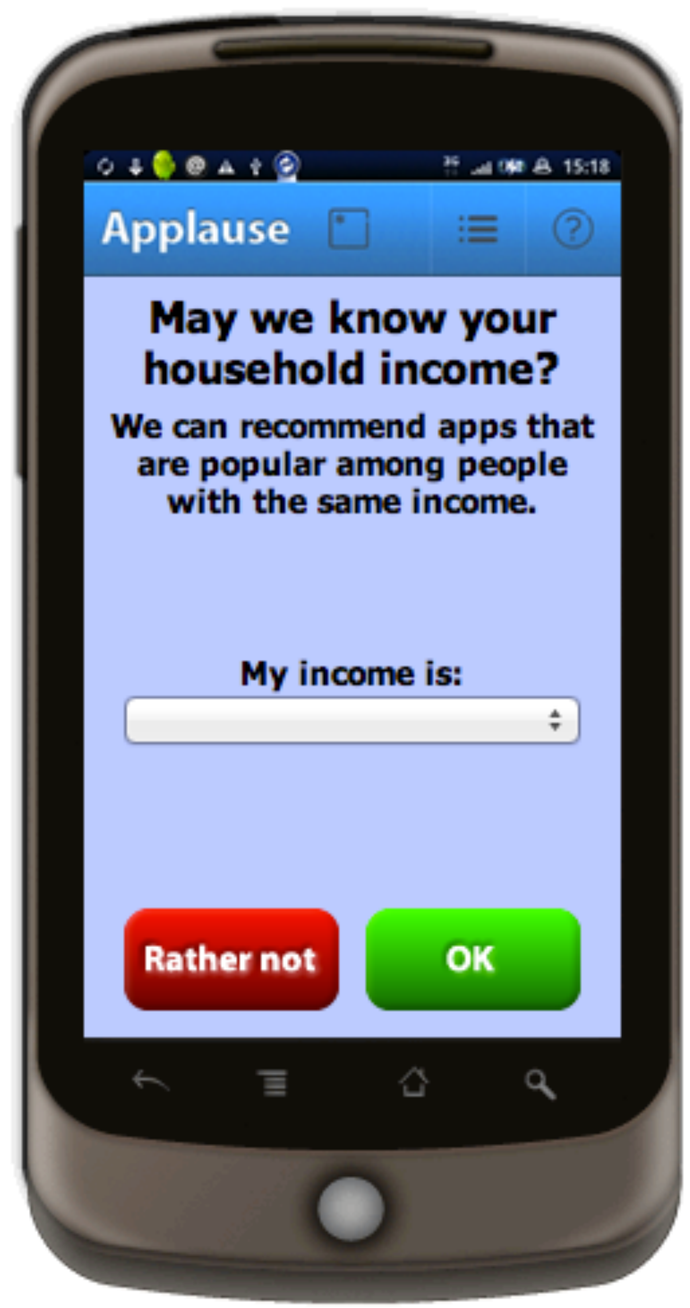
None

Useful for you

Number of others

Useful for others

Explanation



# Example

Variables at level 1:

- decision: whether the participant disclosed the item (1) or not (1)
- 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)

# Example

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

Load package “lme4”

Build a random intercept model:

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

# Build models

Add message and percentage:

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

```
summary(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)
```

# Analysis

Good job!

You now have the stats knowledge of about 95% of the people in this field!

Disco is super impressed!

**Now for the final 5%...**

use an **omnibus test** when testing multiple conditions

use correct methods for **non-normal data**

$x \rightarrow y$

use correct methods for **repeated measures**

# Analysis

Statistical evaluation of the results

check out **Andy Field's book** for more details

# Measurement

Measuring subjective valuations

# Measurement

"To measure satisfaction, we asked users whether they liked the system (on a 5-point rating scale)."

# Why is this bad?

Does the question mean the **same** to everyone?

- John likes the system because it is convenient
- Mary likes the system because it is easy to use
- Dave likes it because the outcomes are useful

A single question is not enough to establish **content validity**

We need a multi-item measurement scale

# Why use a scale?

Objective traits can usually be measured with a single question

- (e.g. age, income)

For subjective traits, single-item measurements lack **content validity**

- Each participant may interpret the item differently
- This reduces precision and conceptual clarity

Accurate measurement requires a **shared conceptual understanding** between all participants and researcher

# Use existing scales

Why?

- Constructing your own scale is a lot of work
- "Famous" scales have undergone extensive validity tests
- Ascertains that two related papers measure exactly the same thing

Finding existing scales:

- In related work (especially if they tested them)
- The Inter-Nomological Network (INN) at inn.theorizeit.org

# Create new scales

When?

- Existing scales do not hold up
- Nobody has measured what you want to measure before
- Scale relates to the specific context of measurement

How:

- Adapt existing scales to your purpose
- Develop a brand new scale (see next slides!)

# Adapting scales

| Information collection concerns: | System-specific concerns: |
|---|---|
| It usually bothers me when websites ask me for personal information. | It bothered me that [system] asked me for my personal information. |
| When websites ask me for personal information, I sometimes think twice before providing it. | I had to think twice before providing my personal information to [system]. |
| It bothers me to give personal information to so many websites. | n/a |
| I am concerned that websites are collecting too much personal information about me. | I am concerned that [system] is collecting too much personal information about me. |

# Concept definition

Start by writing a good concept definition!

A concept definition is a careful explanation of what you want to measure

Examples: leadership

"Leadership is power, influence, and control" (objectivish)

"Leadership is status, respect, and authority" (subjectivish)

"Leadership is woolliness, foldability, and grayness" (nonsensical, but valid!)

# Concept definition

Note: They need to be more detailed than this!

- A good definition makes it unambiguously clear what the concept is supposed to mean
- The foundation for a shared conceptual understanding

Note 2: A concept definition is an equality relation, not a causal relation

- Power, influence, control == leadership
- Not: power, influence, control —> leadership

# Concept definition

If a concept becomes "too broad", split it up!

- e.g. you could create separate concept definitions for power, influence, and control

If two concepts are too similar, try to differentiate them, but otherwise integrate them!

- e.g. "attitude towards the system" and "satisfaction with the system" are often very similar

# Good items...

Use both positively and negatively phrased items

- They make the questionnaire less "leading"
- They help filtering out bad participants
- They explore the "flip-side" of the scale

The word "not" is easily overlooked

Bad: "The results were not very novel."

Good: "The results felt outdated."

# Good items...

Choose simple over specialized words

Bad: "Do you find the illumination of your work environment sufficient to work in?"

Avoid double-barreled questions

Bad: "The recommendations were relevant and fun."

Avoid loaded or leading questions

Bad: "Is it important to treat people fairly?"

# Good items...

"Undecided" and "neutral" are not the same thing

Bad: disagree - somewhat disagree - undecided - somewhat agree - agree

Good: disagree - somewhat disagree - neutral (or: neither agree nor disagree) - somewhat agree - agree

Soften the impact of objectionable questions

Bad: "I do not care about the environment."

Good: "There are more important things than caring about the environment."

# Answer categories

Most common types of items: binary, 5- or 7-point scale

Why? We want to measure the **extent** of the concept:

- Agreement (completely disagree - - - completely agree) or (no - yes)
- Frequency (never - - - very frequently)
- Importance (unimportant - - - very important)
- Quality (very poor - - - very good)
- Likelihood (almost never true - - - almost always true) or (false - true)

# Answer categories

Sometimes, the answer categories represent the item

Based on what I have seen, FormFiller makes it ________ to fill out online forms.

- easy - - neutral - - difficult
- simple - - neutral - - complicated
- convenient - - neutral - - inconvenient
- effortless - - neutral - - daunting
- straightforward - - neutral - - burdensome

# How many items?

One scale for each concept

At least 3 (but preferably 5 or more) items per scale

Developing items involves multiple iterations of testing and revising

- First develop 10–15 items
- Then reduce it to 5–7 through discussions with domain experts and comprehension pre-tests with test subjects
- You may remove 1-2 more items in the final analysis

# Testing items

Experts discussion:

- Card-sorting into concepts (with or without definition)
- Let experts write the definition based on your items, then show them your definition and discuss difference

Comprehension pre-tests:

- Also card-sorting
- Think-aloud testing: ask users to 1) give an answer, 2) explain the question in their own words, and 3) explain their answer

# Examples

Satisfaction:

- In most ways FormFiller is close to ideal.
- I would not change anything about FormFiller.
- I got the important things I wanted from FormFiller.
- FormFiller provides the precise functionality I need.
- FormFiller meets my exact needs.

(completely disagree - disagree - somewhat disagree - neutral - somewhat agree - agree - completely agree)

# Examples

Satisfaction (alternative):

- Check-it-Out is useful.
- Using Check-it-Out makes me happy.
- Using Check-it-Out is annoying.
- Overall, I am satisfied with Check-it-Out.
- I would recommend Check-it-Out to others.

(completely disagree - disagree - somewhat disagree - neutral - somewhat agree - agree - completely agree)

# Examples

Satisfaction (another alternative):

*I am __________ with FormFiller.*

- very dissatisfied - - neutral - - very satisfied
- very displeased - - neutral - - very pleased
- very frustrated - - neutral - - very contended

# Attention checks

Always begin with clear directions

- Ask comprehension questions about the directions

Make sure your participants are paying attention!

- "To make sure you are paying attention, please answer somewhat agree to this question."
- "To make sure you are paying attention, please do not answer agree to this question."
- Repeat certain questions
- Test for non-reversals of reverse-coded questions

# OK solution...

"We asked users ten 5-point scale questions and summed the answers."

# What is missing?

Is the scale really measuring a **single** thing?

– 5 items measure satisfaction, the other 5 convenience

– The items are not related enough to make a reliable scale

Are two scales really measuring **different** things?

– They are so closely related that they actually measure the same thing

We need to establish **construct validity**

This makes sure the scales are unidimensional

# Construct validity

Discriminant validity

Are two scales really measuring different things? (e.g. attitude and satisfaction may be too highly correlated)

Convergent validity

Is the scale really measuring a single thing? (e.g. a usability scale may actually consist of several sub-scales: learnability, effectiveness, efficiency, satisfaction, etc.)

**Factor analysis** (CFA) helps you with construct validity

# Why CFA?

Establish convergent and discriminant validity

CFA can suggest ways to remedy problems with the scale

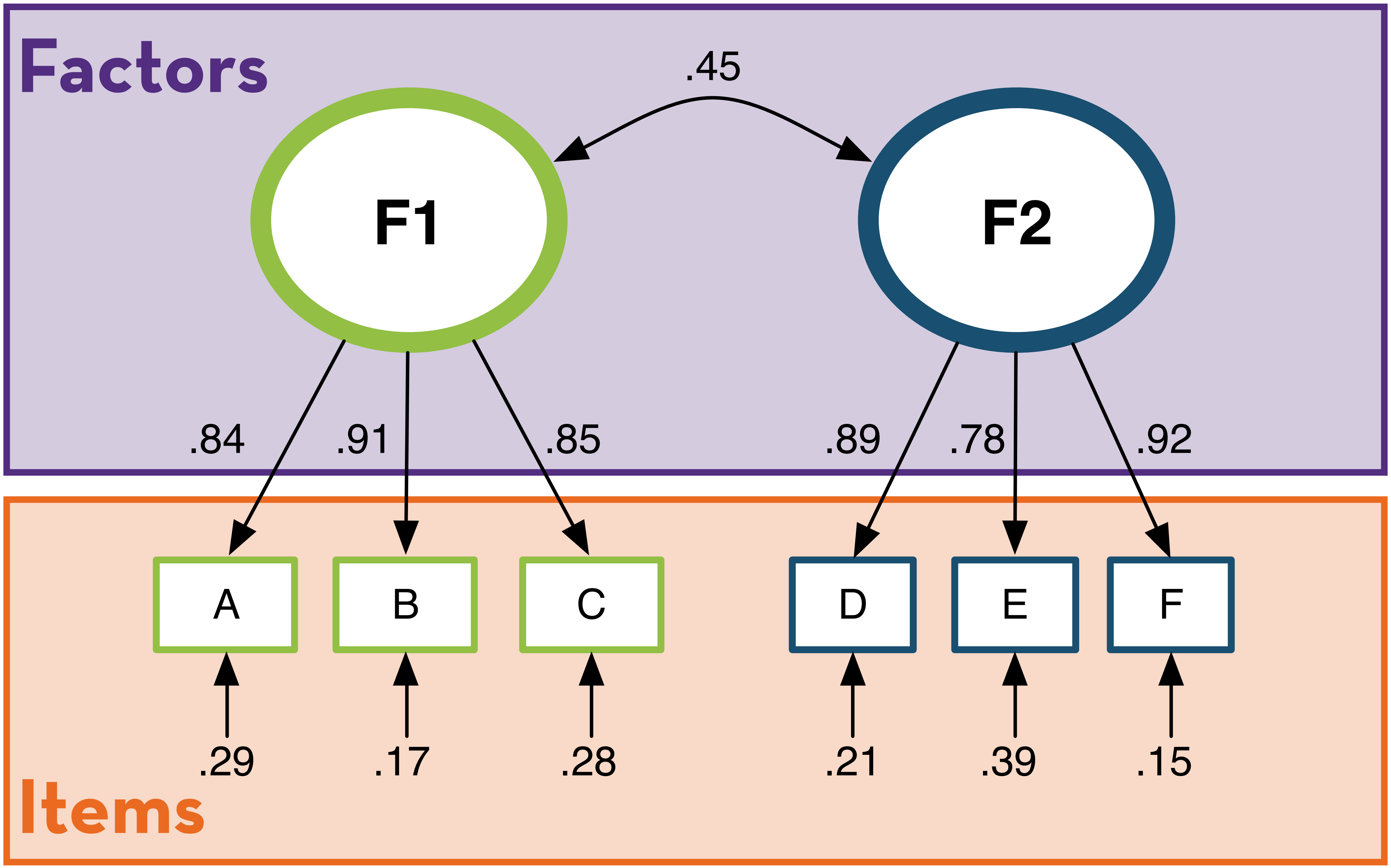
Outcome is a normally distributed measurement scale

Even when the items are yes/no, 5- or 7-point scales!

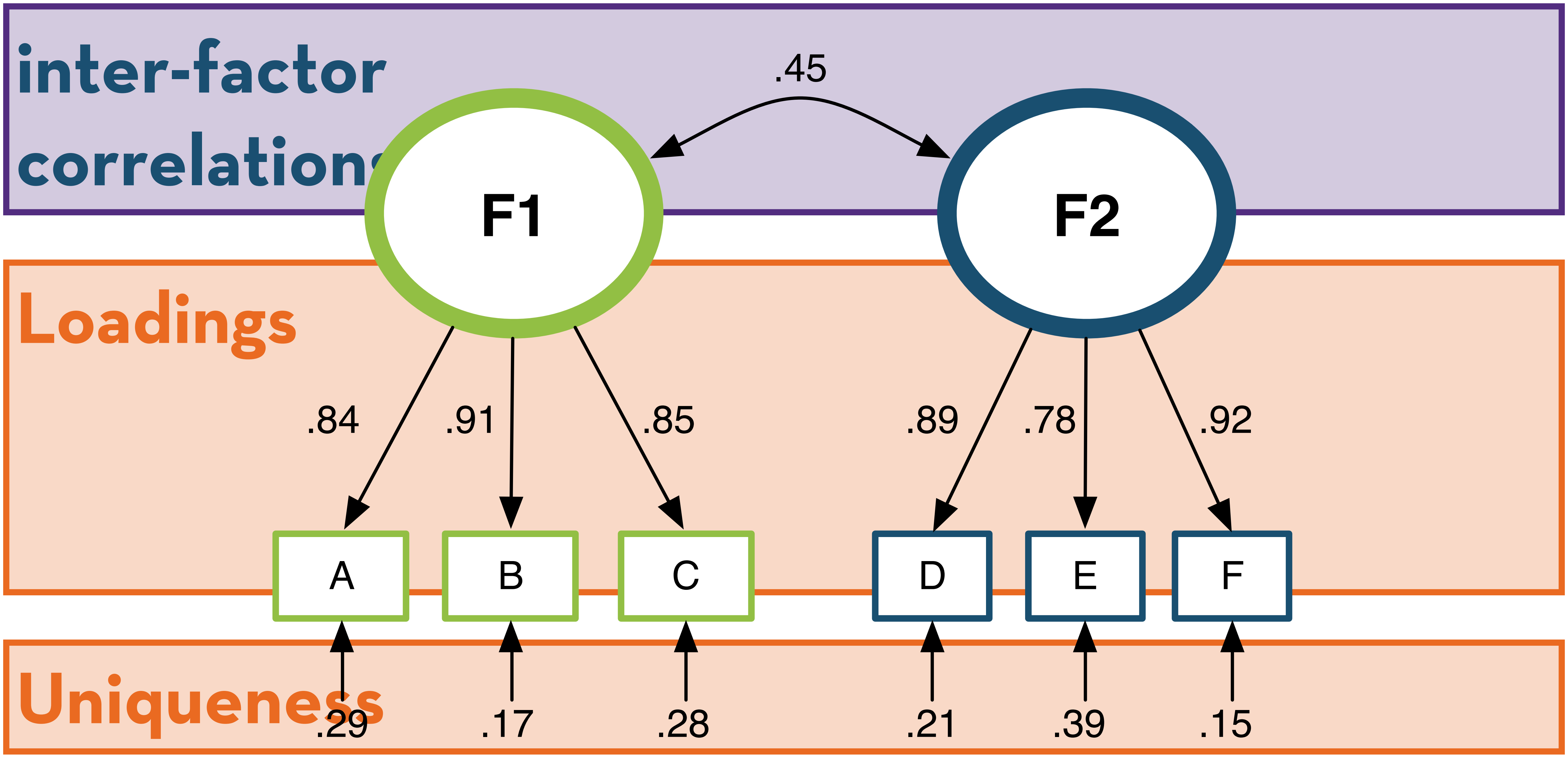
The scale captures the “shared essence” of the items

You can remove the influence of measurement error in your statistical tests!

# CFA: the concept



# CFA: the concept



# CFA: the concept

Factors are **latent constructs** that represent the trait or concept to be measured

- The latent construct cannot be measured directly

The latent construct "**causes**" users' answers to items

- Items are therefore also called **indicators**

Like any measurement, indicators are not perfect measurements

- They depend on the true score (loading) as well as some measurement error (uniqueness)

# How it works

By looking at the **overlap** (covariance) between items, we can separate the measurement error from the true score!

The scale captures the "shared essence" of the items

The basis for Factor Analysis is thus the item correlation matrix

How do we determine the loadings etc?

By **modeling** the correlation matrix as closely as possible!

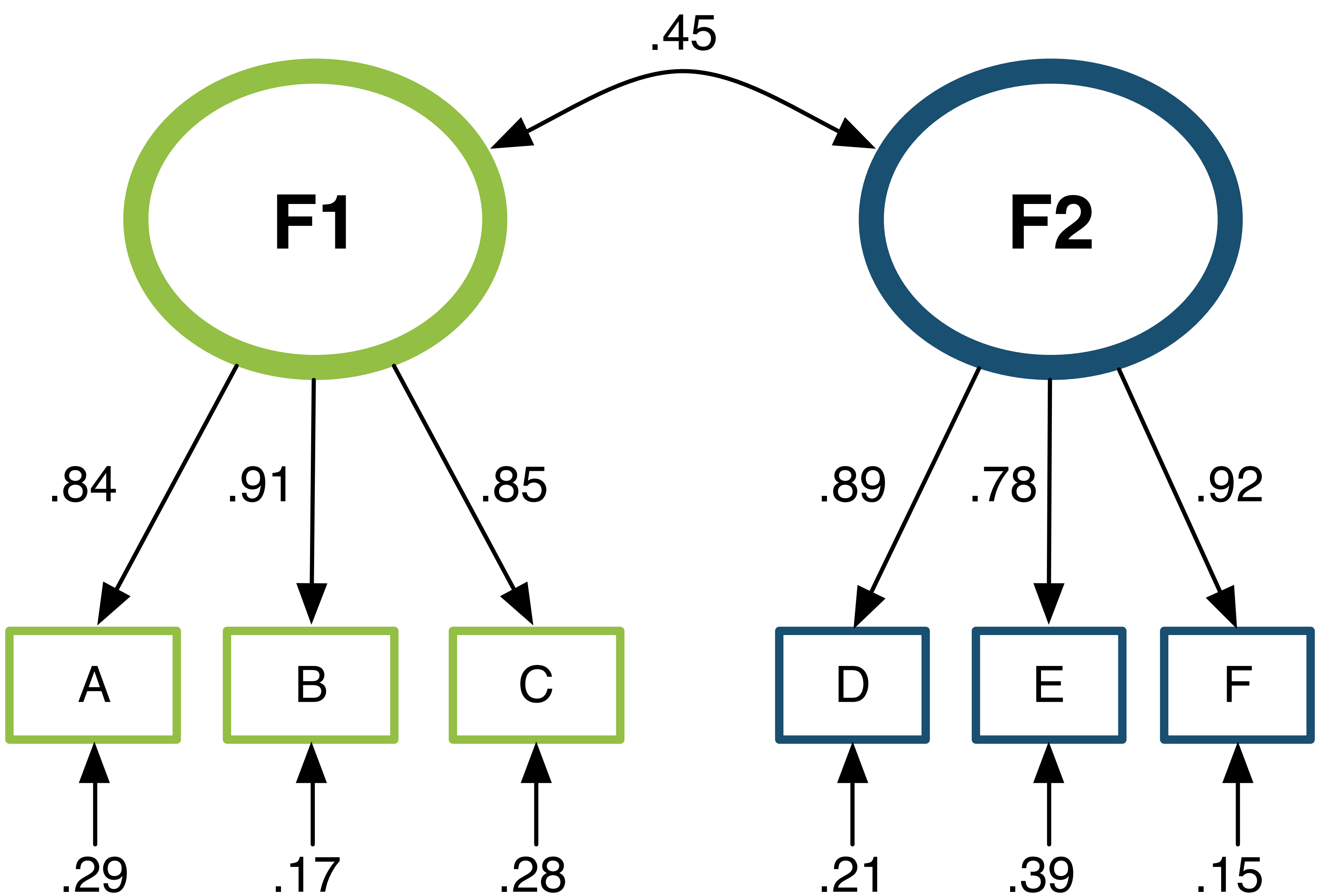
# Observed

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| A | 1.00 | 0.73 | 0.71 | 0.34 | 0.49 | 0.34 |
| B | 0.73 | 1.00 | 0.79 | 0.35 | 0.32 | 0.32 |
| C | 0.71 | 0.79 | 1.00 | 0.29 | 0.33 | 0.35 |
| D | 0.34 | 0.35 | 0.29 | 1.00 | 0.74 | 0.81 |
| E | 0.49 | 0.32 | 0.33 | 0.74 | 1.00 | 0.75 |
| F | 0.34 | 0.32 | 0.35 | 0.81 | 0.75 | 1.00 |

# Observed

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| A | 1.00 | 0.73 | 0.71 | 0.34 | 0.49 | 0.34 |
| B | 0.73 | 1.00 | 0.79 | 0.35 | 0.32 | 0.32 |
| C | 0.71 | 0.79 | 1.00 | 0.29 | 0.33 | 0.35 |
| D | 0.34 | 0.35 | 0.29 | 1.00 | 0.74 | 0.81 |
| E | 0.49 | 0.32 | 0.33 | 0.74 | 1.00 | 0.75 |
| F | 0.34 | 0.32 | 0.35 | 0.81 | 0.75 | 1.00 |

# Model



# Estimated

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| A | 0.71 | 0.76 | 0.71 | 0.34 | 0.29 | 0.35 |
| B | 0.76 | 0.83 | 0.77 | 0.36 | 0.32 | 0.38 |
| C | 0.71 | 0.77 | 0.72 | 0.34 | 0.30 | 0.35 |
| D | 0.34 | 0.36 | 0.34 | 0.79 | 0.69 | 0.82 |
| E | 0.29 | 0.32 | 0.30 | 0.69 | 0.61 | 0.72 |
| F | 0.35 | 0.38 | 0.35 | 0.82 | 0.72 | 0.85 |

# Residual

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| A | 0.29 | –0.03 | 0.00 | 0.00 | 0.20 | –0.01 |
| B | –0.03 | 0.17 | 0.02 | –0.01 | 0.00 | –0.06 |
| C | 0.00 | 0.02 | 0.28 | –0.05 | 0.03 | 0.00 |
| D | 0.00 | –0.01 | –0.05 | 0.21 | 0.05 | –0.01 |
| E | 0.20 | 0.00 | 0.03 | 0.05 | 0.39 | 0.03 |
| F | –0.01 | –0.06 | 0.00 | –0.01 | 0.03 | 0.15 |

# Example

twq.dat, variables:

- cgraph: inspectability (0: list, 1: graph)
- citem-cfriend: control (baseline: no control)
- cig (citem * cgraph) and cfg (cfriend * cgraph)
- s1-s7: satisfaction with the system
- q1-q6: perceived recommendation quality
- c1-c5: perceived control
- u1-u5: understandability

# Example

twq.dat, variables:

- e1-e4: user music expertise
- t1-t6: propensity to trust
- f1-f6: familiarity with recommenders
- average rating of, and number of known items in, the top 10
- time taken to inspect the recommendations

# Run the CFA

Write model definition:

```
model <- ‘satisf =~ s1+s2+s3+s4+s5+s6+s7
quality =~ q1+q2+q3+q4+q5+q6
control =~ c1+c2+c3+c4+c5
underst =~ u1+u2+u3+u4+u5’
```

Run cfa (load package lavaan):

```
fit <- cfa(model, data=twq, ordered=names(twq), std.lv=TRUE)
```

Inspect model output:

```
summary(fit, rsquare=TRUE, fit.measures=TRUE)
```

# Run the CFA

Output (model fit):

```
lavaan (0.5-17) converged normally after  39 iterations

  Number of observations                           267

  Estimator                                       DWLS      Robust
  Minimum Function Test Statistic              251.716     365.719
  Degrees of freedom                               224         224
  P-value (Chi-square)                           0.098       0.000
  Scaling correction factor                                  1.012
  Shift parameter                                          117.109
    for simple second-order correction (Mplus variant)

Model test baseline model:

  Minimum Function Test Statistic            48940.029   14801.250
  Degrees of freedom                               253         253
  P-value                                        0.000       0.000
```

# Run the CFA

Output (model fit, continued):

```
User model versus baseline model:

  Comparative Fit Index (CFI)                    0.999       0.990
  Tucker-Lewis Index (TLI)                       0.999       0.989

Root Mean Square Error of Approximation:

  RMSEA                                          0.022       0.049
  90 Percent Confidence Interval          0.000  0.034       0.040  0.058
  P-value RMSEA <= 0.05                          1.000       0.579

Weighted Root Mean Square Residual:

  WRMR                                           0.855       0.855

Parameter estimates:

  Information                                 Expected
  Standard Errors                           Robust.sem
```

# Run the CFA

Output (loadings):

```
                   Estimate  Std.err  Z-value  P(>|z|)
  Latent variables:
    satisf =~
      s1              0.888    0.018   49.590    0.000
      s2             -0.885    0.018  -48.737    0.000
      s3              0.771    0.029   26.954    0.000
      s4              0.821    0.025   32.363    0.000
      s5              0.889    0.018   50.566    0.000
      s6              0.788    0.031   25.358    0.000
      s7             -0.845    0.022  -38.245    0.000
    quality =~
      q1              0.950    0.013   72.421    0.000
      q2              0.949    0.013   72.948    0.000
      q3              0.942    0.012   77.547    0.000
      q4              0.805    0.033   24.257    0.000
      q5             -0.699    0.042  -16.684    0.000
      q6             -0.774    0.040  -19.373    0.000
```

# Run the CFA

Output (loadings, continued):

```
  control =~
    c1                0.712    0.038   18.684    0.000
    c2                0.855    0.024   35.624    0.000
    c3                0.905    0.022   41.698    0.000
    c4                0.723    0.037   19.314    0.000
    c5               -0.424    0.056   -7.571    0.000
   underst =~
    u1               -0.557    0.047  -11.785    0.000
    u2                0.899    0.016   57.857    0.000
    u3                0.737    0.030   24.753    0.000
    u4               -0.918    0.016  -58.229    0.000
    u5                0.984    0.010   97.787    0.000
```

# Run the CFA

Output (factor correlations):

```
Covariances:
  satisf ~~
    quality           0.686    0.033   20.503    0.000
    control          -0.760    0.028  -26.913    0.000
    underst           0.353    0.048    7.320    0.000
  quality ~~
    control          -0.648    0.040  -16.041    0.000
    underst           0.278    0.058    4.752    0.000
  control ~~
    underst          -0.382    0.051   -7.486    0.000
```

# Run the CFA

Output (factor correlations):

```
Covariances:
  satisf ~~
    quality           0.686    0.033   20.503    0.000
    control          -0.760    0.028  -26.913    0.000
    underst           0.353    0.048    7.320    0.000
  quality ~~
    control          -0.648    0.040  -16.041    0.000
    underst           0.278    0.058    4.752    0.000
  control ~~
    underst          -0.382    0.051   -7.486    0.000
```

# Run the CFA

## Output (variance extracted):

```
R-Square:

    s1              0.788
    s2              0.782
    s3              0.594
    s4              0.674
    s5              0.790
    s6              0.621
    s7              0.714
    q1              0.903
    q2              0.901
    q3              0.888
    q4              0.648
    q5              0.489
    q6              0.599
    c1              0.506
    c2              0.731
    c3              0.820
    c4              0.522
    c5              0.179
    u1              0.310
    u2              0.808
    u3              0.544
    u4              0.843
    u5              0.968
```

# Things to inspect

Item-fit: Loadings, communality, residuals

Remove items that do not fit

Factor-fit: Average Variance Extracted

Respecify or remove factors that do not fit

Model-fit: Chi-square test, CFI, TLI, RMSEA

Make sure the model meets criteria

# Item-fit metrics

Variance extracted (squared loading):

- The amount of variance explained by the factor (1-uniqueness)
- Should be > 0.50 (although some argue 0.40 is okay)

In lavaan output: r-squared

Based on r-squared, iteratively remove items:

c5 (r-squared = 0.180)

u1 (r-squared = 0.324)

# Item-fit metrics

Residual correlations:

- The observed correlation between two items is significantly higher (or lower) than predicted
- Might mean that factors should be split up

Cross-loadings:

- When the model suggest that the model fits significantly better if an item also loads on an additional factor
- Could mean that an item actually measures two things

# Item-fit metrics

In R: modification indices

- We only look the ones that are significant and large enough to be interesting (decision == "epc")

```
mods <- modindices(fit,power=TRUE)
mods[mods$decision == "epc",]
```

Based on modification indices, remove item:

- u3 loads on control (modification index = 24.667)
- Some residual correlations within Satisfaction (might mean two factors?), but we ignore those because AVE is good (see next couple of slides)

# Item-fit metrics

For all these metrics:

- Remove items that do not meet the criteria, but be careful to keep at least 3 items per factor
- One may remove an item that has values much lower than other items, even if it meets the criteria

# Factor-fit

Average Variance Extracted (AVE) in lavaan output:

average of R-squared per factor

Convergent validity:

AVE > 0.5

Discriminant validity

$\sqrt{(\text{AVE})}$ > largest correlation with other factors

# Factor-fit

Satisfaction:

AVE = 0.709, √(AVE) = 0.842, largest correlation = 0.762

Quality:

AVE = 0.737, √(AVE) = 0.859, largest correlation = 0.687

Control:

AVE = 0.643, √(AVE) = 0.802, largest correlation = 0.762

Understandability:

AVE = 0.874, √(AVE) = 0.935, largest correlation = 0.341

# Model-fit metrics

Chi-square test of model fit:

- Tests whether there any significant misfit between estimated and observed correlation matrix
- Often this is true ($p < .05$)... models are rarely perfect!
- Alternative metric: chi-squared / df < 3 (good fit) or < 2 (great fit)

# Model-fit metrics

CFI and TLI:

- Relative improvement over baseline model; ranging from 0.00 to 1.00
- CFI should be > 0.96 and TLI should be > 0.95

RMSEA:

- Root mean square error of approximation
- Overall measure of misfit
- Should be < 0.05, and its confidence intervall should not exceed 0.10.

# Model-fit

Use the "robust" column in R:

- Chi-Square value: 288.517, df: 164 (value/df = 1.76, good)
- CFI: 0.990, TLI: 0.989 (both good)
- RMSEA: 0.053 (slightly high), 90% CI: [0.043, 0.063] (ok)

# Summary

Specify and run your CFA

Alter the model until all remaining items fit

Make sure you have at least 3 items per factor!

Report final loadings, factor fit, and model fit

# Summary

We conducted a CFA and examined the validity and reliability scores of the constructs measured in our study.

Upon inspection of the CFA model, we removed items c5 (communality: 0.180) and u1 (communality: 0.324), as well as item u3 (high cross-loadings with several other factors). The remaining items shared at least 48% of their variance with their designated construct.

# Summary

To ensure the convergent validity of constructs, we examined the average variance extracted (AVE) of each construct. The AVEs were all higher than the recommended value of 0.50, indicating adequate convergent validity.

To ensure discriminant validity, we ascertained that the square root of the AVE for each construct was higher than the correlations of the construct with other constructs.

# Summary

| Construct | Item | Loading |
|---|---|---|
| System satisfaction | I would recommend TasteWeights to others. | 0.888 |
| | TasteWeights is useless. | -0.885 |
| | TasteWeights makes me more aware of my choice options. | 0.768 |
| Alpha: 0.92 | I can make better music choices with TasteWeights. | 0.822 |
| AVE: 0.709 | I can find better music using TasteWeights. | 0.889 |
| | Using TasteWeights is a pleasant experience. | 0.786 |
| | TasteWeights has no real benefit for me. | -0.845 |
| Perceived Recommendation Quality | I liked the artists/bands recommended by the TasteWeights system. | 0.950 |
| | The recommended artists/bands fitted my preference. | 0.950 |
| | The recommended artists/bands were well chosen. | 0.942 |
| Alpha: 0.90 | The recommended artists/bands were relevant. | 0.804 |
| AVE: 0.737 | TasteWeights recommended too many bad artists/bands. | -0.697 |
| | I didn't like any of the recommended artists/bands. | -0.775 |
| Perceived Control | I had limited control over the way TasteWeights made recommendations. | 0.700 |
| | TasteWeights restricted me in my choice of music. | 0.859 |
| Alpha: 0.84<br>AVE: 0.643 | Compared to how I normally get recommendations, TasteWeights was very limited. | 0.911 |
| | I would like to have more control over the recommendations. | 0.716 |
| | I decided which information was used for recommendations. | |
| Understandability | The recommendation process is not transparent. | |
| | I understand how TasteWeights came up with the recommendations. | 0.893 |
| Alpha: 0.92<br>AVE: 0.874 | TasteWeights explained the reasoning behind the recommendations. | |
| | I am unsure how the recommendations were generated. | -0.923 |
| | The recommendation process is clear to me. | 0.987 |

# Summary

| | Alpha | AVE | Satisfaction | Quality | Control | Underst. |
|---|---|---|---|---|---|---|
| **Satisfaction** | 0.92 | 0.709 | **0.842** | 0.687 | –0.762 | 0.336 |
| **Quality** | 0.90 | 0.737 | 0.687 | **0.859** | –0.646 | 0.282 |
| **Control** | 0.84 | 0.643 | –0.762 | –0.646 | **0.802** | –0.341 |
| **Underst.** | 0.92 | 0.874 | 0.336 | 0.282 | –0.341 | **0.935** |

diagonal: $\sqrt{(AVE)}$

off-diagonal: correlations

establish content validity with **multi-item scales**

follow the general principles for **good questionnaire items**

establish **convergent** and **discriminant** validity

# Measurement

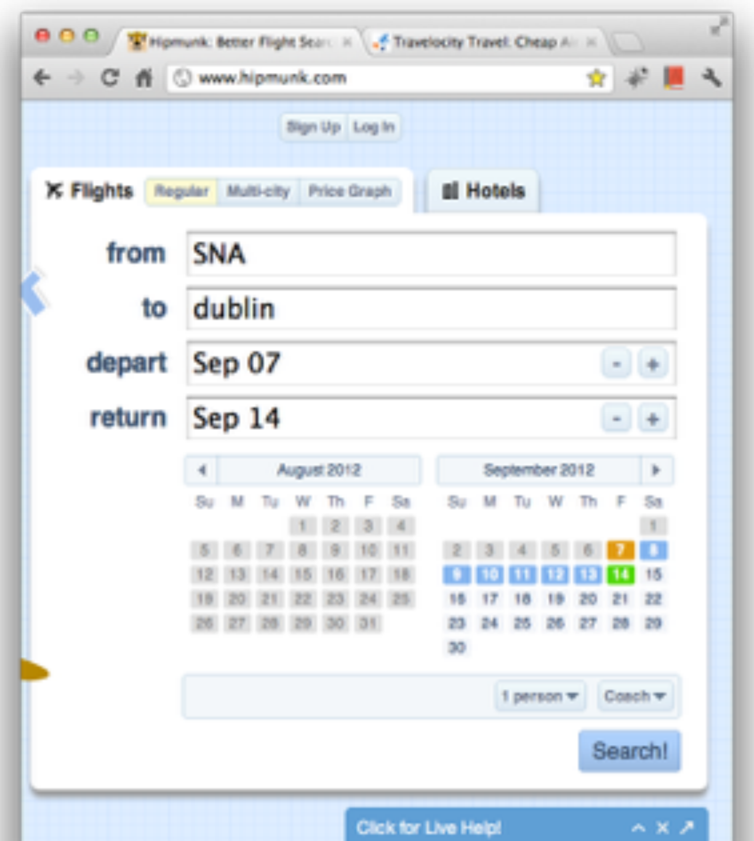
Measuring subjective valuations

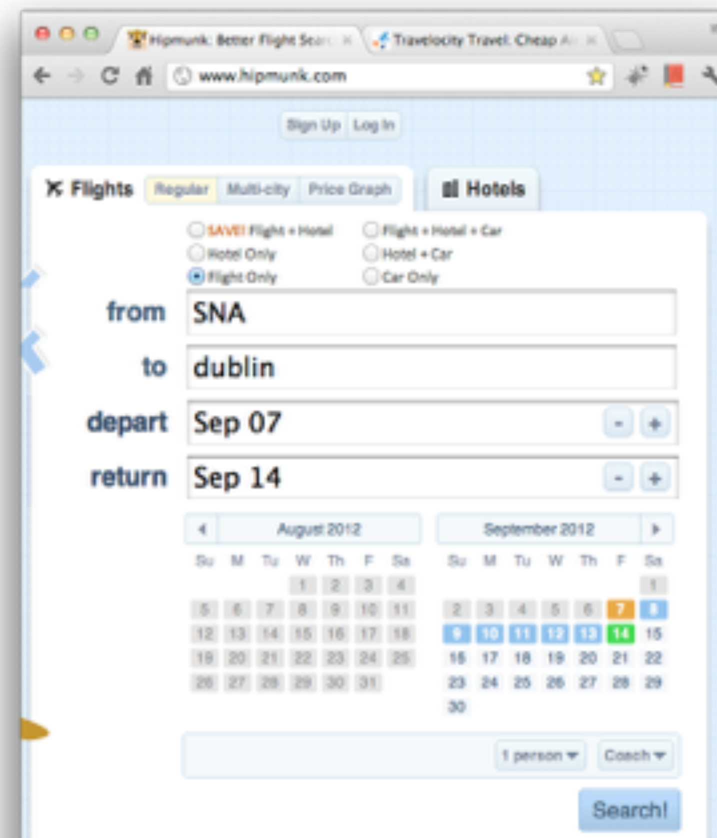
use **factor analysis**

# Evaluating Models

An introduction to Structural Equation Modeling

# Evaluating Models




Test whether fewer options leads to lower/higher usability

# Theory behind x->y

To learn something from a study, we need a **theory** behind the effect

- This makes the work generalizable
- This may suggest future work

Measure **mediating variables**

- Measure understandability (and a number of other concepts) as well
- Find out how they mediate the effect on usability

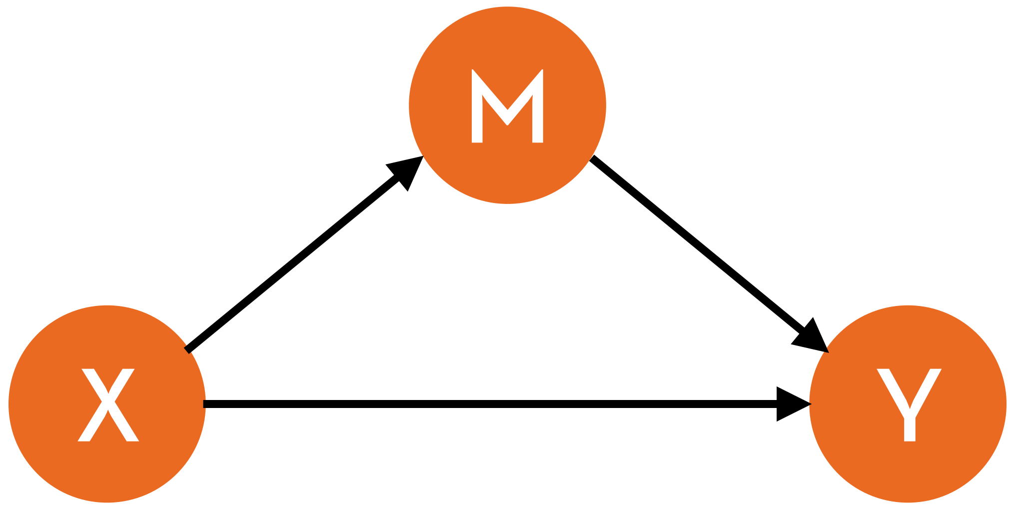
# Mediation Analysis

X -> M -> Y

- Does the system (X) influence usability (Y) via understandability (M)?

Types of mediation

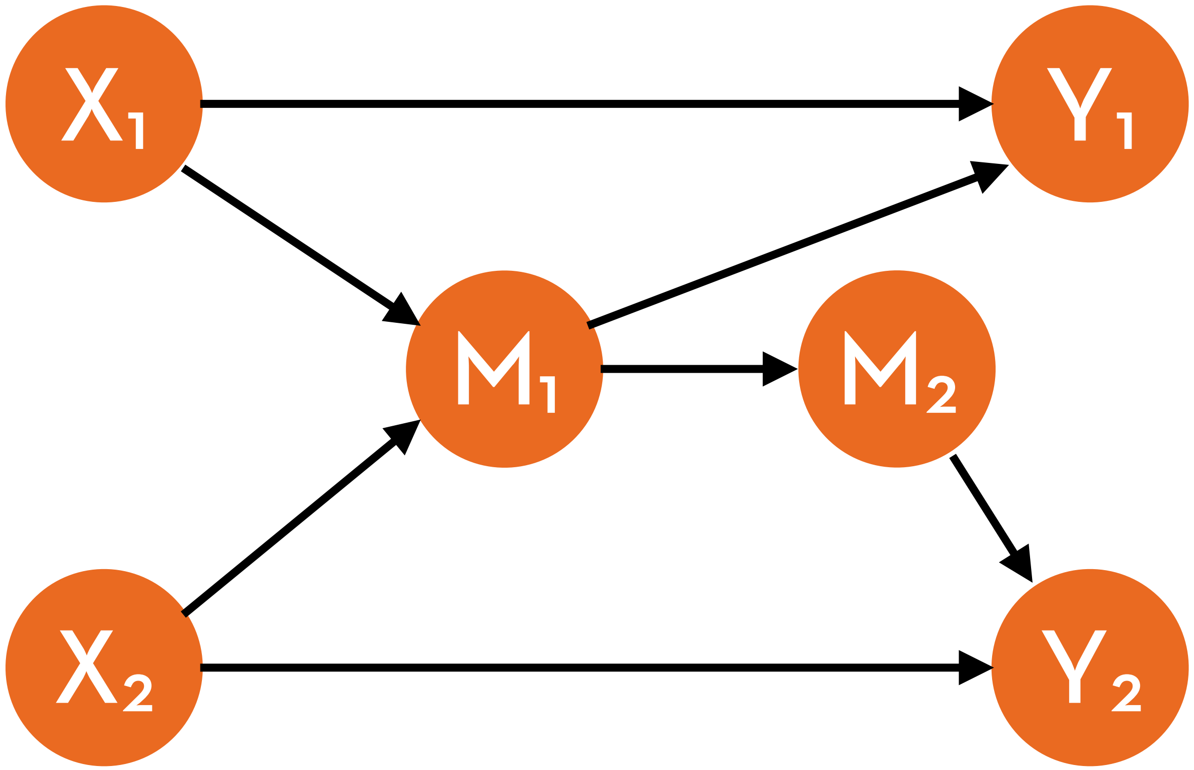
- Partial mediation
- Full mediation
- Negative mediation



# Mediation Analysis

More complex models:

- What is the total effect of X1 on Y2?
- Is this effect significant?
- Is this effect fully or partially mediated by M1 and M2?



# What is SEM?

A Structural Equation Model (SEM) is a CFA where the factors are regressed on each other and on the experimental manipulations

(observed behaviors can also be incorporated)

The regressions are not estimated one-by-one, but **all at the same time**

(and so is the CFA part of the model, actually)

# Why SEM?

Easy way to test for **mediation**

…without doing many separate tests

You can **keep factors** as factors

This ascertains normality, and leads to more statistical power in the regressions

The model has several **overall fit indices**

You can judge the fit of an entire model, rather than just its parts

# Keep the factors!

Let's say we have a factor F measuring trait Y, with AVE = 0.64

- On average, 64% of the item variance is communality, 36% is uniqueness

If we **sum the items** of the factor as S, this results in 36% error

- This is random noise that does not measure Y

Result: no regression with S as dependent can have an R-squared > 0.64!

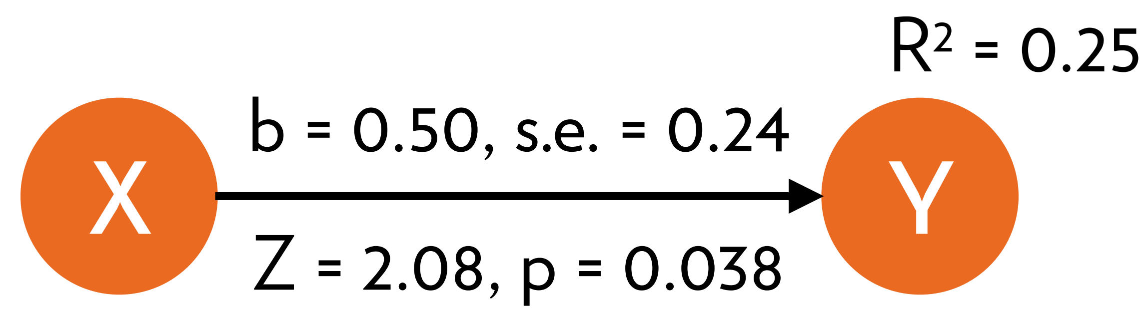
# Keep the factors!

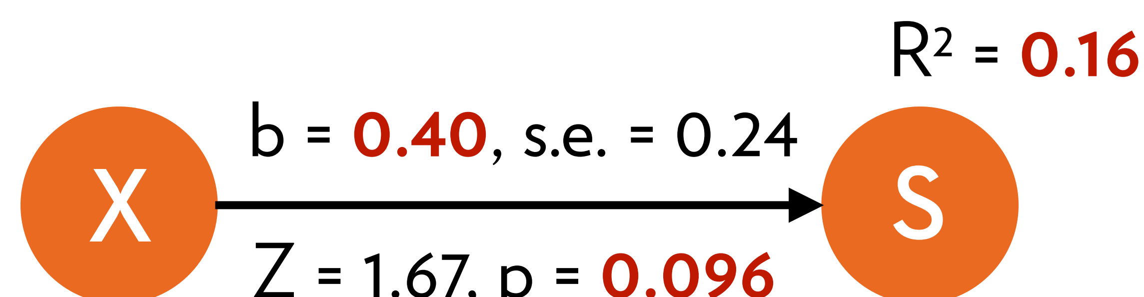
Any regression coefficient will be **attenuated** by the AVE of S!

Take for instance this X, which potentially explains 25% of the variance of Y…

…it only explains 16% of the variance of S!

…and the effect is non-significant!

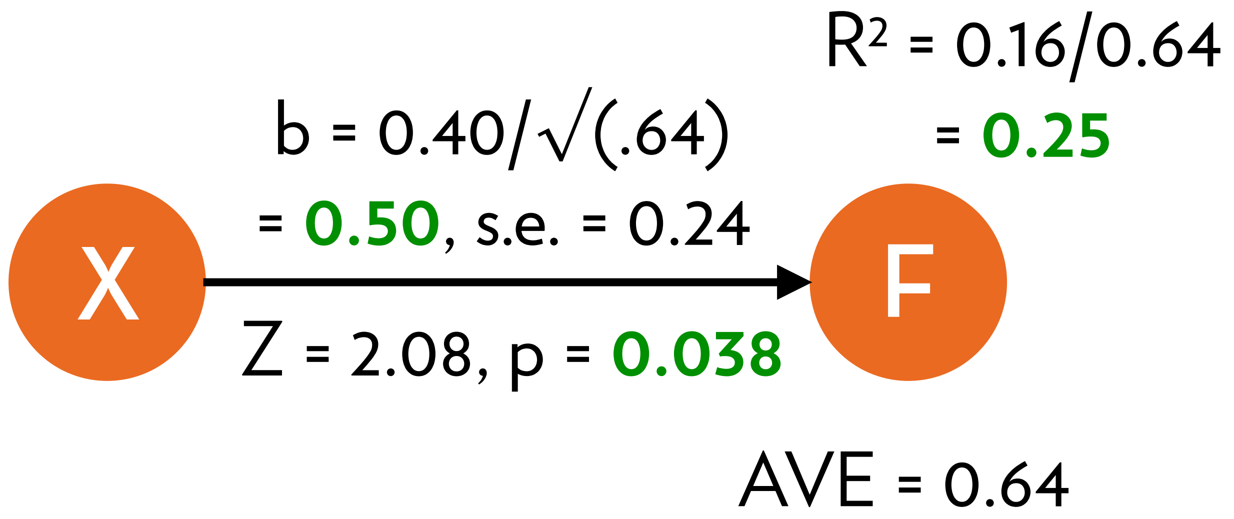


# Keep the factors!

If we use F instead of S, we **know** that the AVE is 0.64

...so we can **compensate** for the incurred measurement error!



# Estimates

In a SEM you can get the following estimates (all at once):

- Item loadings
- $R^2$ for every dependent variable
- Regression coefficients for all regressions (B, s.e., p-values)

Plus, you can get omnibus tests for testing manipulations with > 2 conditions

# Steps

Steps involved in constructing a SEM:

(a method that is confirmatory, but leaves room for data-driven changes in the model)

Step 1: Build your CFA ✓

Step 2: Analyze the marginal effects of the manipulations

Step 3: Set up a model based on theory

Step 4: Test and trim a saturated version of this model

# 2. Marginal effects

First analysis: manipulations —> factors

MIMIC model (Multiple Indicators, Multiple Causes)

The SEM equivalent of a t-test / (factorial) ANOVA

Only for experiments (not for surveys)

Steps involved:

- Build your CFA (see session 2 slides)
- Create dummies for your experimental conditions
- Run regressions factor-by-factor

# Create dummies

Main effects are already built for our dataset:

- Control conditions ("no control" is the baseline):

  `citem cfriend`

- Inspectability conditions ("list view" is the baseline):

  `cgraph`

What about the interaction effect?

- Use for citem*cgraph and cfriend*cgraph!

  `cig cfg`

# Add regression

Add a regression to your final CFA model:

```
model <- 'satisf =~ s1+s2+s3+s4+s5+s6+s7
quality =~ q1+q2+q3+q4+q5+q6
control =~ c1+c2+c3+c4
underst =~ u2+u4+u5
satisf ~ citem+cfriend+cgraph+cig+cfg';

fit <-
sem(model,data=twq,ordered=names(twq[9:31]),std.lv=TRUE);

summary(fit);
```

# Results

Note: effects are not significant (but that's okay for now)

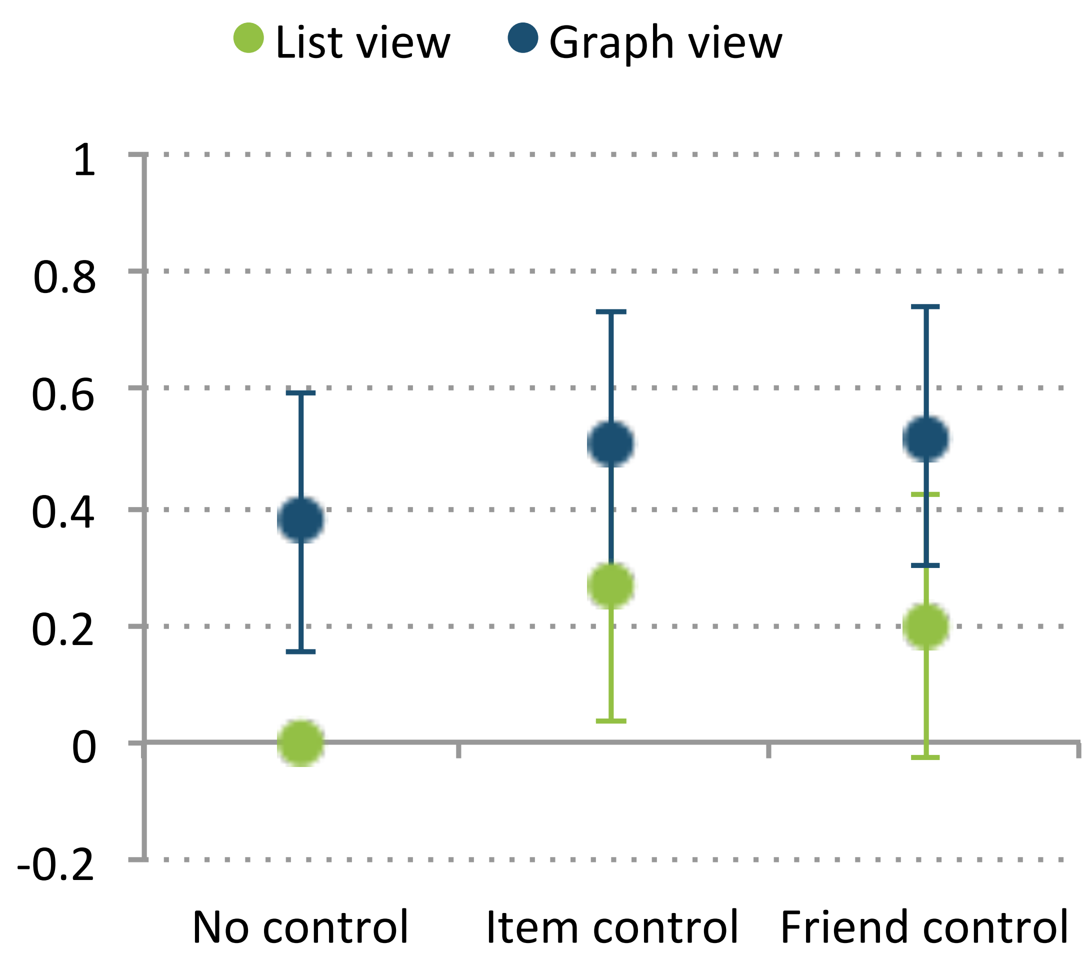
```
                      Estimate  Std.err  Z-value  P(>|z|)
...(factors)...            ...      ...      ...      ...
Regressions:
  satisf ~
    citem                0.269    0.234    1.153    0.249
    cfriend              0.197    0.223    0.882    0.378
    cgraph               0.375    0.221    1.694    0.090
    cig                 -0.131    0.320   -0.408    0.683
    cfg                 -0.048    0.309   -0.156    0.876
```

# Code for a graph

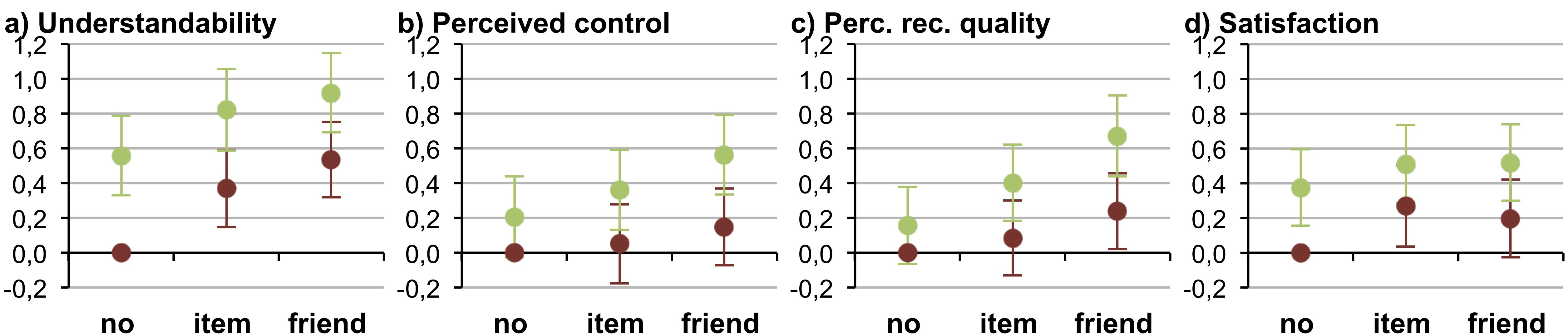
Use dummies for each condition (except "list view, no control" condition):

```
model <- 'satisf =~ s1+s2+s3+s4+s5+s6+s7
quality =~ q1+q2+q3+q4+q5+q6
control =~ c1+c2+c3+c4
underst =~ u2+u4+u5
satisf ~ cil+cfl+cng+cig+cfg';

fit <-
sem(model,data=twq,ordered=names(twq[1:23]),std.lv=TRUE);

summary(fit);
```

# Create a graph



# Repeat




From: Knijnenburg et al. (2012): "Inspectability and Control in Social Recommenders", *RecSys'12*

# Main finding

Main effects of inspectability and control conditions on understandability (no interaction effect)

Similar to regression!

```
                     Estimate  Std.err  Z-value  P(>|z|)
...(factors)...           ...      ...      ...      ...
Regressions:
  underst ~
    citem               0.367    0.220    1.666    0.096
    cfriend             0.534    0.216    2.466    0.014
    cgraph              0.556    0.227    2.450    0.014
    cig                -0.105    0.326   -0.323    0.746
    cfg                -0.178    0.320   -0.555    0.579
```

# 3. Modeling: theory

Do this **before** you do your study!

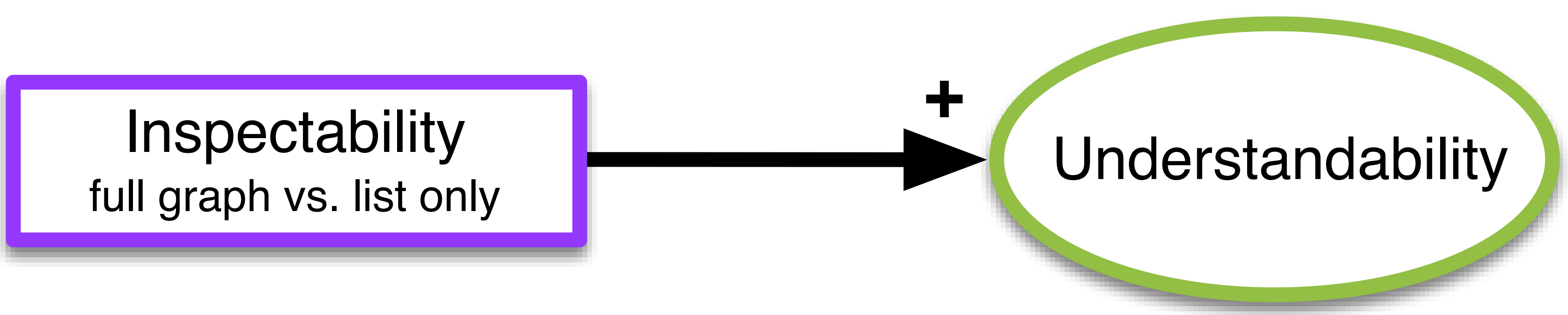
Motivate expected effects, based on:

- previous work
- theory
- common sense

If in doubt, create alternate specifications!

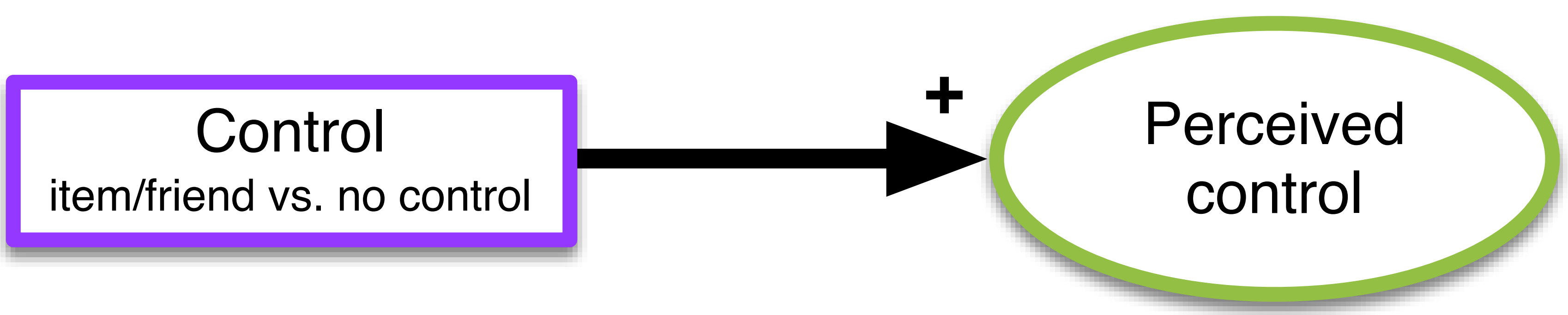
# Inspectability

Herlocker argues that explanation provides transparency, "exposing the reasoning behind a recommendation".



# Control

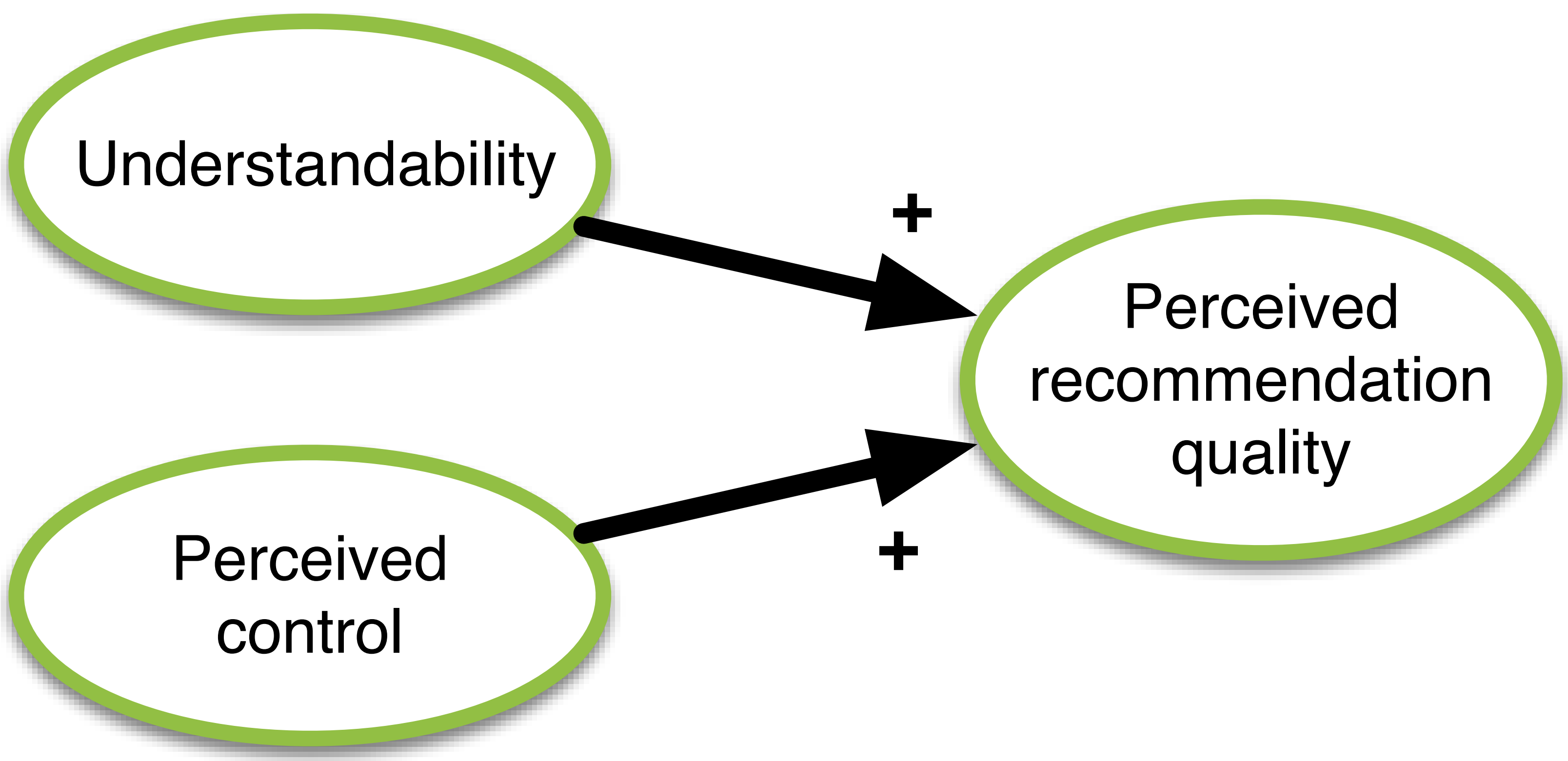
Multiple studies highlight the benefits of interactive interfaces that support control over the recommendation process.



# Perceived quality

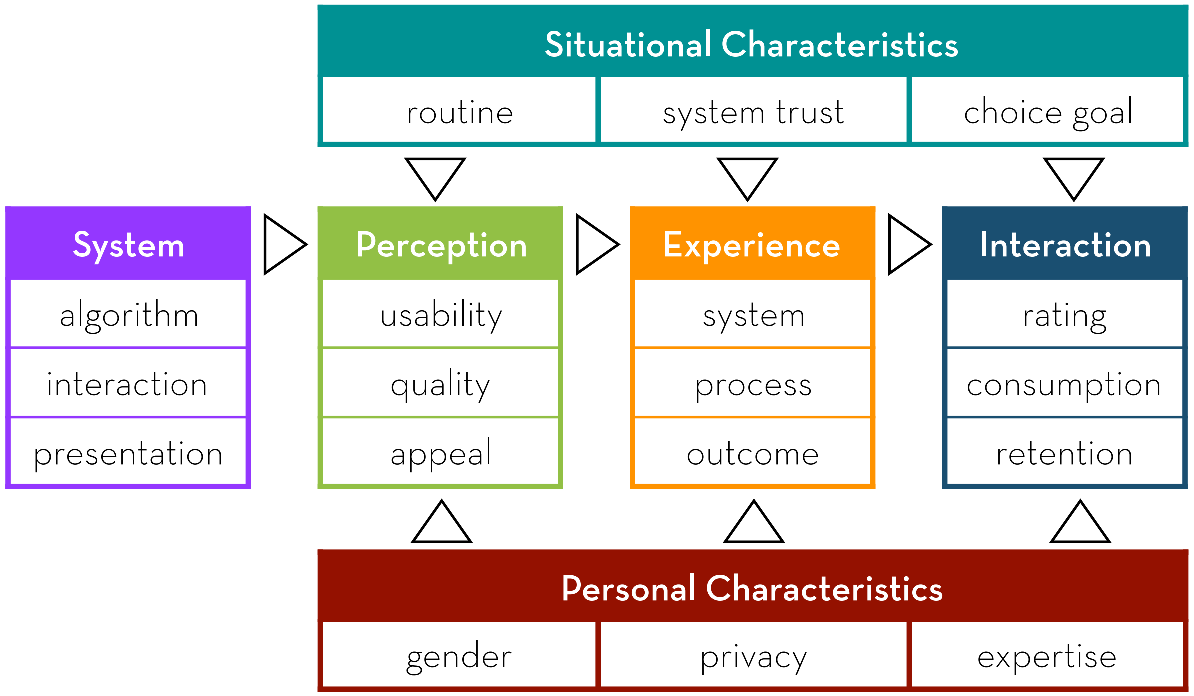
Tintarev and Masthoff show that explanations make it easier to judge the quality of recommendations.

McNee et al. found that study participants preferred user-controlled interfaces because these systems "best understood their tastes".



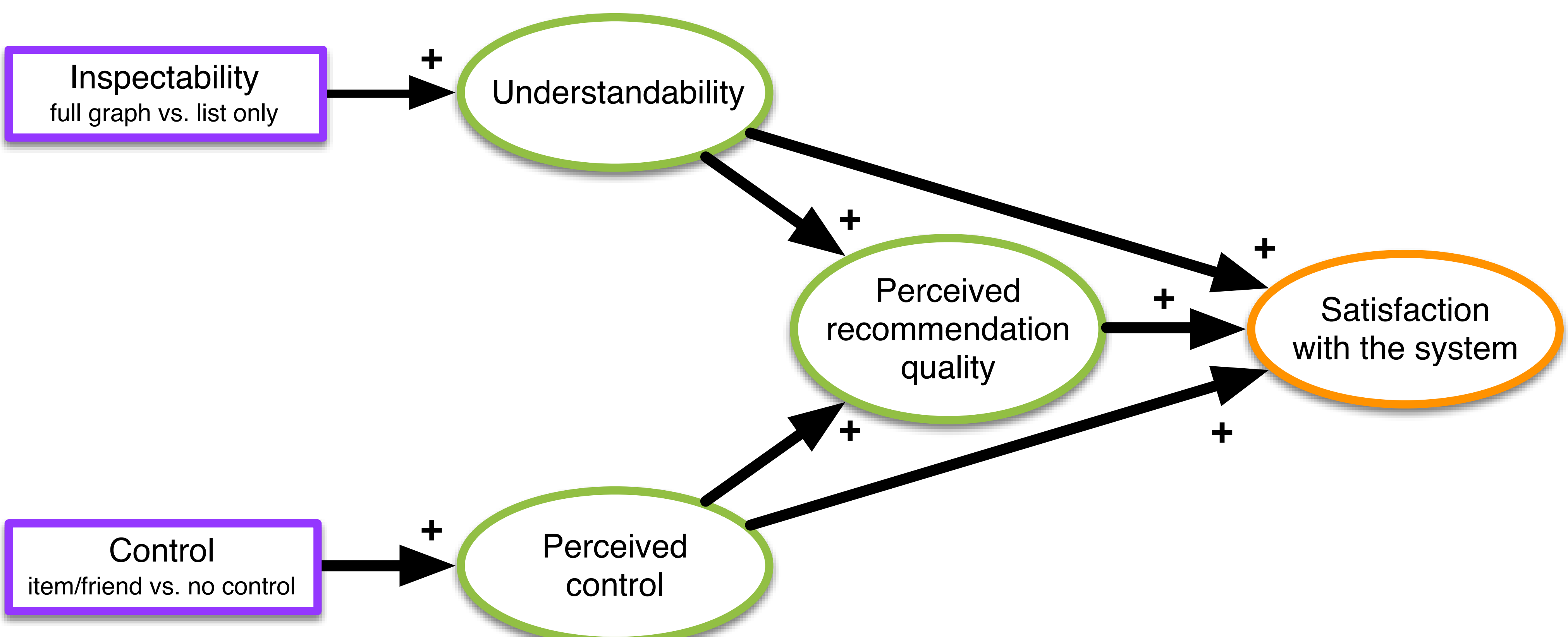
# Satisfaction

Knijnenburg et al. developed a framework that describes how certain manipulations influence subjective system aspects (i.e. understandability, perceived control and recommendation quality), which in turn influence user experience (i.e. system satisfaction).



# Satisfaction

Knijnenburg et al. developed a framework that describes how certain manipulations influence subjective system aspects (i.e. understandability, perceived control and recommendation quality), which in turn influence user experience (i.e. system satisfaction).



# 4. Test the model

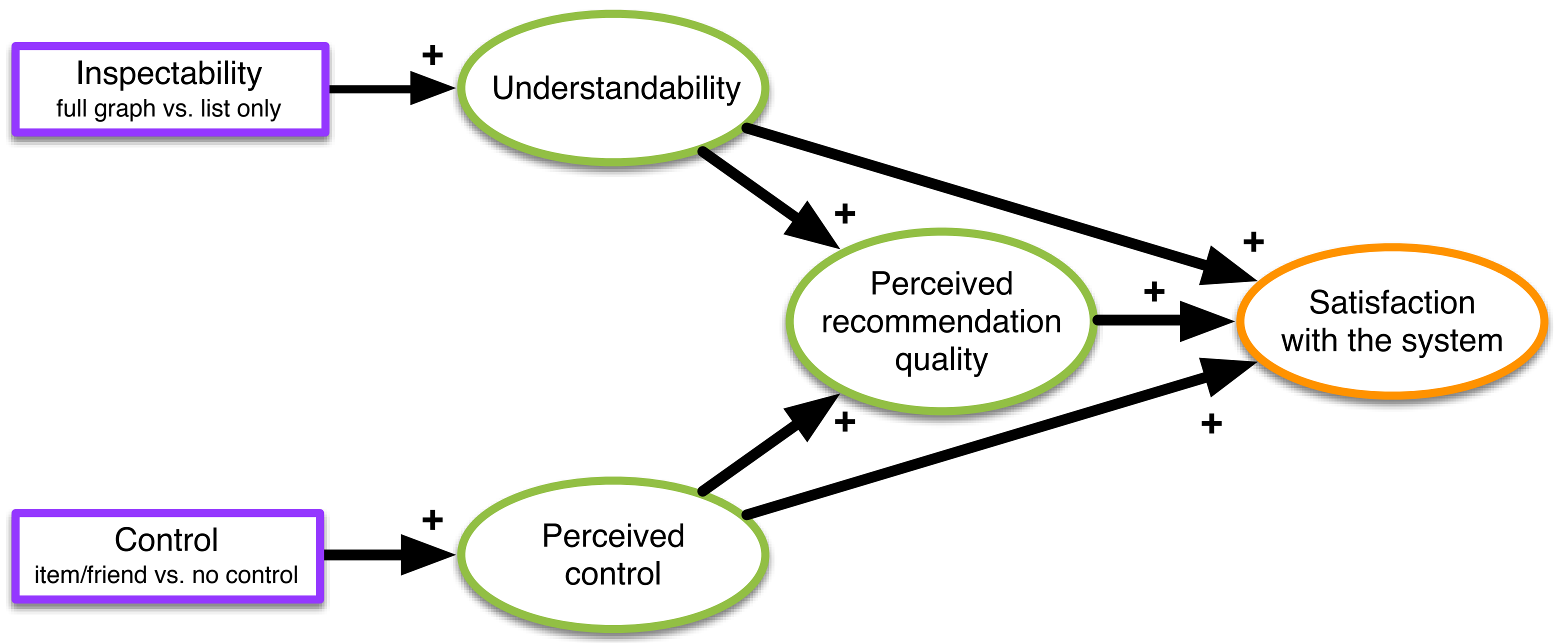
Steps:

- Build a saturated model
- Trim the model
- Get model fit statistics
- Optional: expand the model
- Reporting

# Causal order

Find the causal order of your model

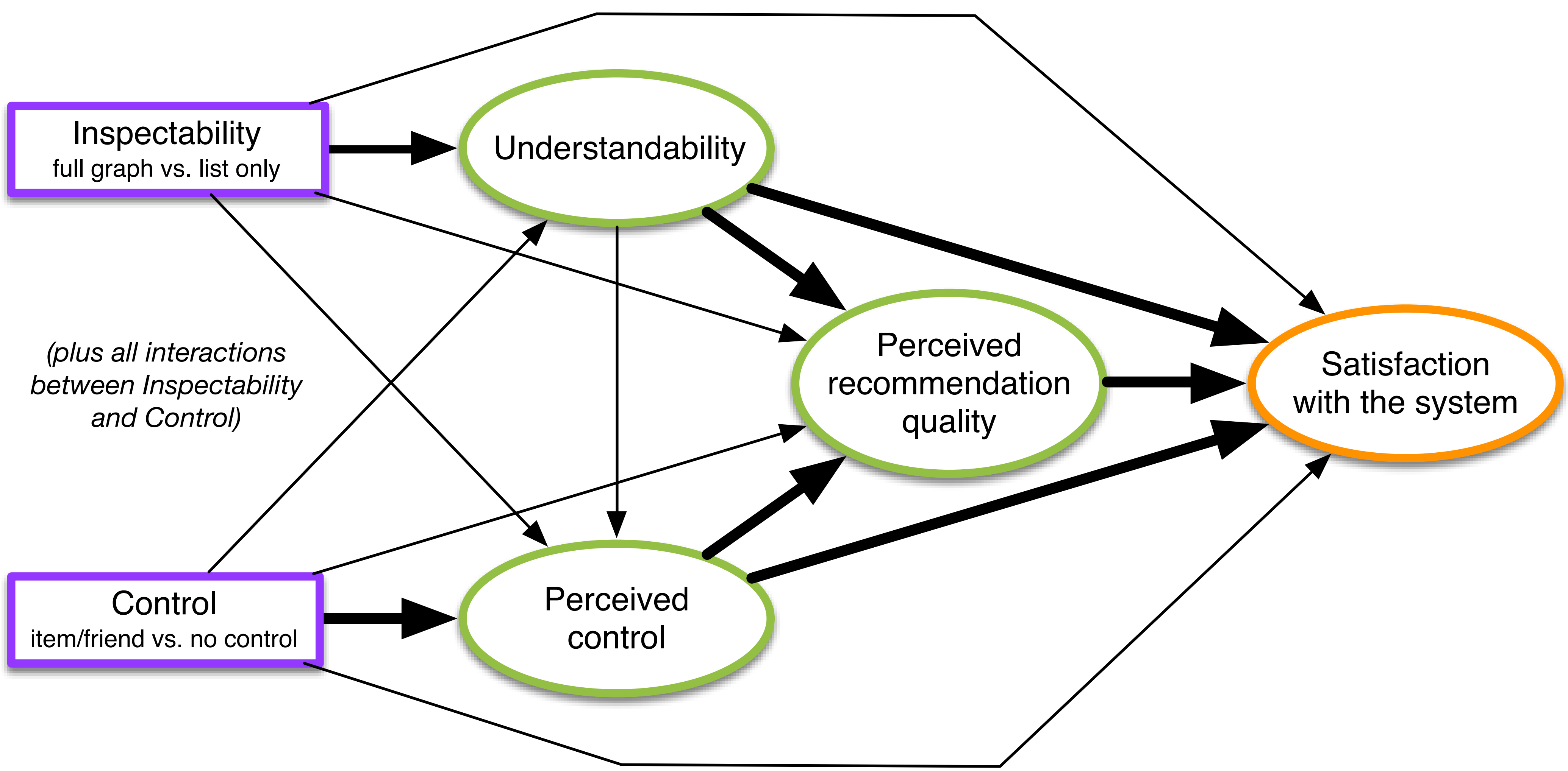
(fill the gaps where necessary)




**conditions -> understandability -> perceived control -> perceived recommendation quality -> satisfaction**

# Saturated model

Fill in all forward-going arrows



# Run model

In R:

```
model <- 'satisf =~ s1+s2+s3+s4+s5+s6+s7
  quality =~ q1+q2+q3+q4+q5+q6
  control =~ c1+c2+c3+c4
  underst =~ u2+u4+u5
  satisf ~ quality+control+underst+citem+cfriend+cgraph+cig+cfg
  quality ~ control+underst+citem+cfriend+cgraph+cig+cfg
  control ~ underst+citem+cfriend+cgraph+cig+cfg
  underst ~ citem+cfriend+cgraph+cig+cfg';

fit <- sem(model,data=twq,ordered=names(twq[9:31]),std.lv=TRUE);

summary(fit);
```

# Trim model

Rules:

- Start with the least significant and least interesting effects (those that were added for saturation)
- Work iteratively
- Manipulations with >2 conditions: remove all dummies at once (if one is significant, keep the others as well)
- Interaction+main effects: never remove main effect before the interaction effect (if the interaction is significant, keep the main effect regardless)

# Results

```
                   Estimate  Std.err  Z-value  P(>|z|)
...(factors)...       ...      ...      ...      ...
Regressions:
  satisf ~
    quality          0.439    0.076    5.753    0.000
    control         -0.838    0.107   -7.804    0.000
    underst          0.090    0.073    1.229    0.219
    citem            0.318    0.265    1.198    0.231
    cfriend          0.014    0.257    0.054    0.957
    cgraph           0.308    0.229    1.346    0.178
    cig             -0.386    0.356   -1.082    0.279
    cfg             -0.394    0.357   -1.103    0.270
  quality ~
    control         -0.764    0.086   -8.899    0.000
    underst          0.044    0.073    0.595    0.552
    citem            0.046    0.204    0.226    0.821
    cfriend          0.165    0.251    0.659    0.510
    cgraph           0.009    0.236    0.038    0.970
    cig              0.106    0.317    0.334    0.738
    cfg              0.179    0.374    0.478    0.632
```

# Results

```
control ~
  underst              -0.308    0.066   -4.695    0.000
  citem                 0.053    0.240    0.220    0.826
  cfriend               0.009    0.221    0.038    0.969
  cgraph               -0.043    0.239   -0.181    0.857
  cig                  -0.148    0.341   -0.434    0.664
  cfg                  -0.273    0.331   -0.824    0.410
underst ~
  citem                 0.367    0.220    1.666    0.096
  cfriend               0.534    0.217    2.465    0.014
  cgraph                0.556    0.227    2.451    0.014
  cig                  -0.106    0.326   -0.324    0.746
  cfg                  -0.178    0.320   -0.555    0.579
```

# Trimming steps

Remove interactions -> (1) understandability, (2) quality, (3) control, and (4) satisfaction

Remove cgraph -> (1) satisfaction, and (2) quality

# Trimming steps

Remove citem and cfriend -> control

But wait… did we not hypothesize that effect?

- Yes, but we still have citem+cfriend -> underst -> control!

In other words: the effect of item and friend control on perceived control is mediated by understandability!

- Argument: "Controlling items/friends gives me a better understanding of how the system works, so in turn I feel more in control"

# Trimming steps

Remove citem and cfriend -> satisfaction

Remove understandability -> recommendation quality

- We hypothesized this effect, but it is still mediated by control.
- Argument: "Understanding the recommendations gives me a feeling of control, which in turn makes me like the recommendations better."

Remove understandability -> satisfaction

- Same thing

# Trimming steps

Remove citem and cfriend -> recommendation quality

Remove cgraph -> control

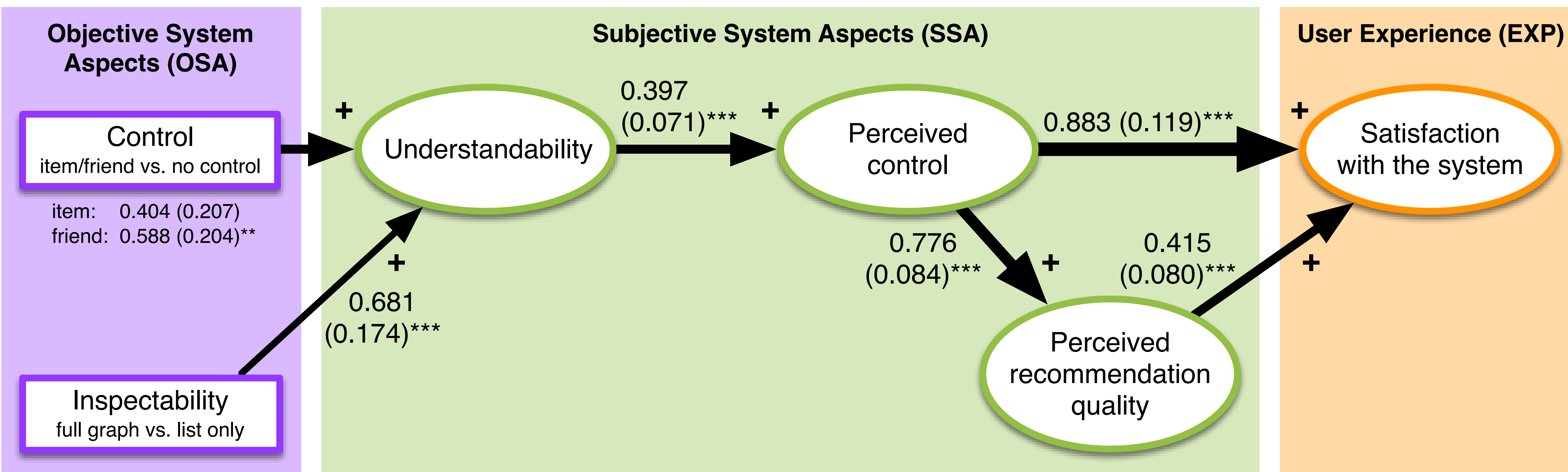
- Again: still mediated by understandability

Stop! All remaining effects are significant!

# Trimmed model

```
                   Estimate  Std.err  Z-value  P(>|z|)
...(factors)...         ...      ...      ...      ...
Regressions:
  satisf ~
    quality           0.418    0.080    5.228    0.000
    control          -0.887    0.120   -7.395    0.000
  quality ~
    control          -0.779    0.084   -9.232    0.000
  control ~
    underst          -0.371    0.067   -5.522    0.000
  underst ~
    citem             0.382    0.200    1.915    0.056
    cfriend           0.559    0.195    2.861    0.004
    cgraph            0.628    0.166    3.786    0.000
```

# Trimmed model



# Modindices

|   | lhs | op | rhs | mi | mi.scaled | epc | sepc.lv | sepc.all | sepc.nox | delta | ncp | power | decision |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | satisf | =~ | q2 | 7.008 | 5.838 | -0.078 | -0.132 | -0.132 | -0.132 | 0.1 | 11.522 | 0.924 | epc |
| 2 | satisf | =~ | q6 | 6.200 | 5.164 | -0.084 | -0.142 | -0.141 | -0.141 | 0.1 | 8.883 | 0.846 | epc |
| 3 | s2 | ~~ | s7 | 10.021 | 8.347 | 0.101 | 0.101 | 0.100 | 0.100 | 0.1 | 9.815 | 0.880 | epc |
| 4 | s3 | ~~ | s4 | 20.785 | 17.313 | 0.157 | 0.157 | 0.156 | 0.156 | 0.1 | 8.381 | 0.825 | epc |
| 5 | s4 | ~~ | s5 | 5.211 | 4.341 | 0.067 | 0.067 | 0.066 | 0.066 | 0.1 | 11.625 | 0.926 | epc |
| 6 | q1 | ~~ | q2 | 5.249 | 4.372 | 0.067 | 0.067 | 0.066 | 0.066 | 0.1 | 11.800 | 0.930 | epc |

No substantial and significant modification indices in the regression part of the model (only stuff we had left from the CFA)

# Assess model fit

Item and factor fit should not have changed much

(please double-check!)

Great model fit!

- Chi-Square value: 306.685, df: 223 (value/df = 1.38)
- CFI: 0.994, TLI: 0.993
- RMSEA: 0.037 (great), 90% CI: [0.026, 0.047]

# Regression $R^2$

Satisfaction: 0.654

Perceived Recommendation Quality: 0.416

Perceived Control: 0.156

Understandability: 0.151

These are all quite okay

# Omnibus test

In model definition:

```
underst ~ cgraph+p1*citem+p2*cfriend
```

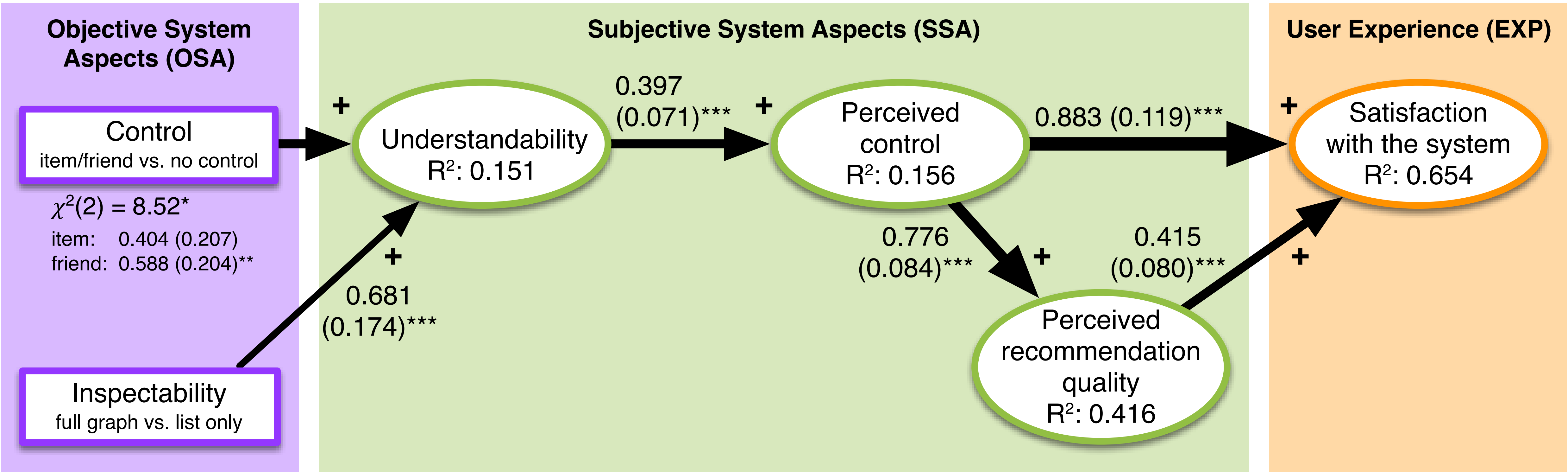
Then run:

```
lavTestWald(fit,'p1==0;p2==0');
```

Result: Omnibus effect of control is significant (this is a chi-square test)

```
$stat
[1] 8.386272

$df
[1] 2

$p.value
[1] 0.01509886
```

# Final core model



# Reporting

We subjected the 4 factors and the experimental conditions to structural equation modeling, which simultaneously fits the factor measurement model and the structural relations between factors and other variables. The model has a good* model fit: chi-square(223) = 306.685, $p = .0002$; RMSEA = 0.037, 90% CI: [0.026, 0.047], CFI = 0.994, TLI = 0.993.

* A model should not have a non-significant chi-square ($p > .05$), but this statistic is often regarded as too sensitive. Hu and Bentler propose cut-off values for other fit indices to be: CFI > .96, TLI > .95, and RMSEA < .05, with the upper bound of its 90% CI below 0.10.

# Reporting

The model shows that the inspectability and control manipulations each have an independent positive effect on the understandability of the system: the full graph condition is more understandable than the list only condition, and the item control and friend control conditions are more understandable than the no control condition. Understandability is in turn related to users' perception of control, which is in turn related to the perceived quality of the recommendations. The perceived control and the perceived recommendation quality finally determine participants' satisfaction with the system.

# Expand the model

Expanding the model by adding additional variables

- This is typically where behavior comes in

Redo model tests and additional stats

# Expanded model

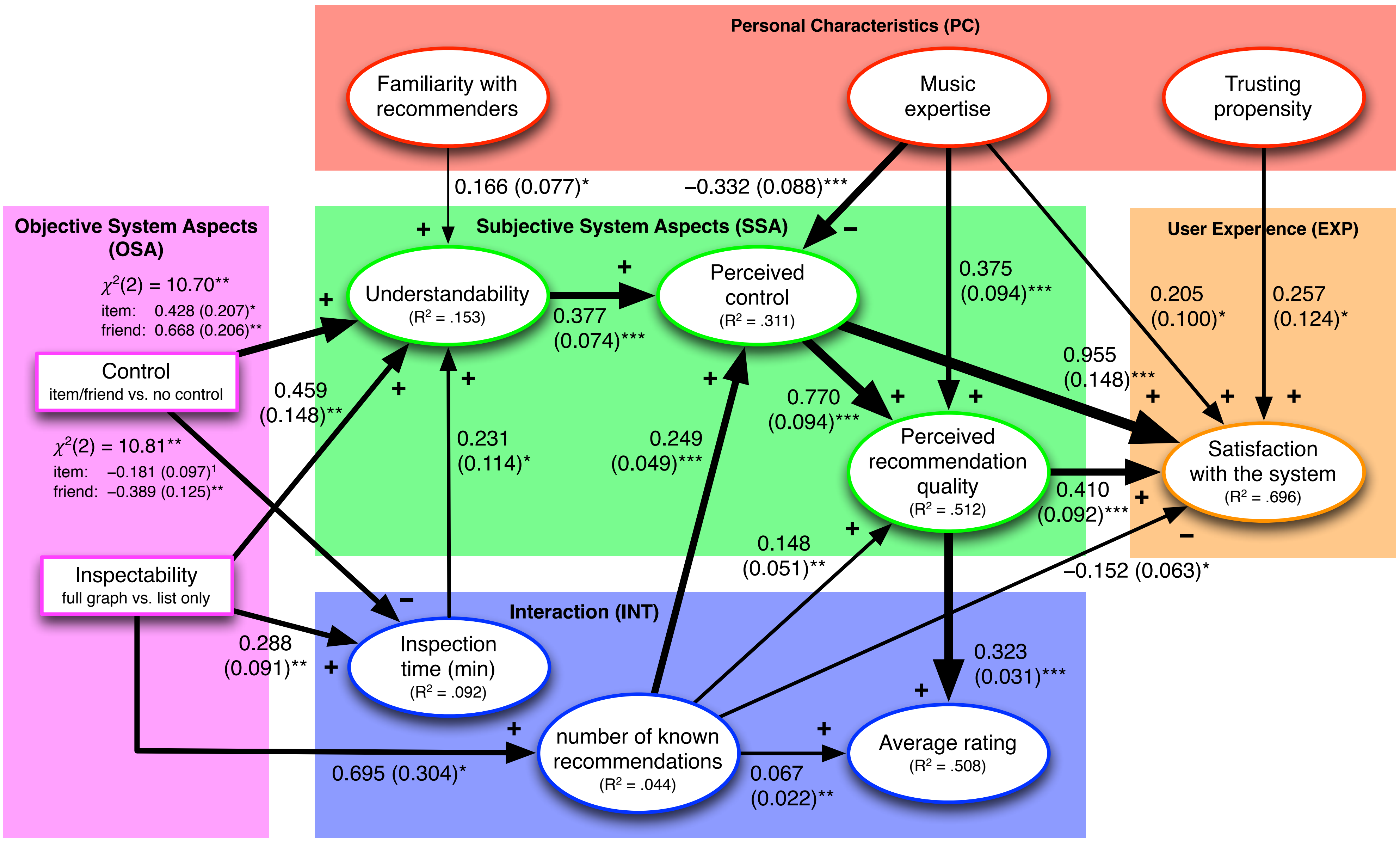
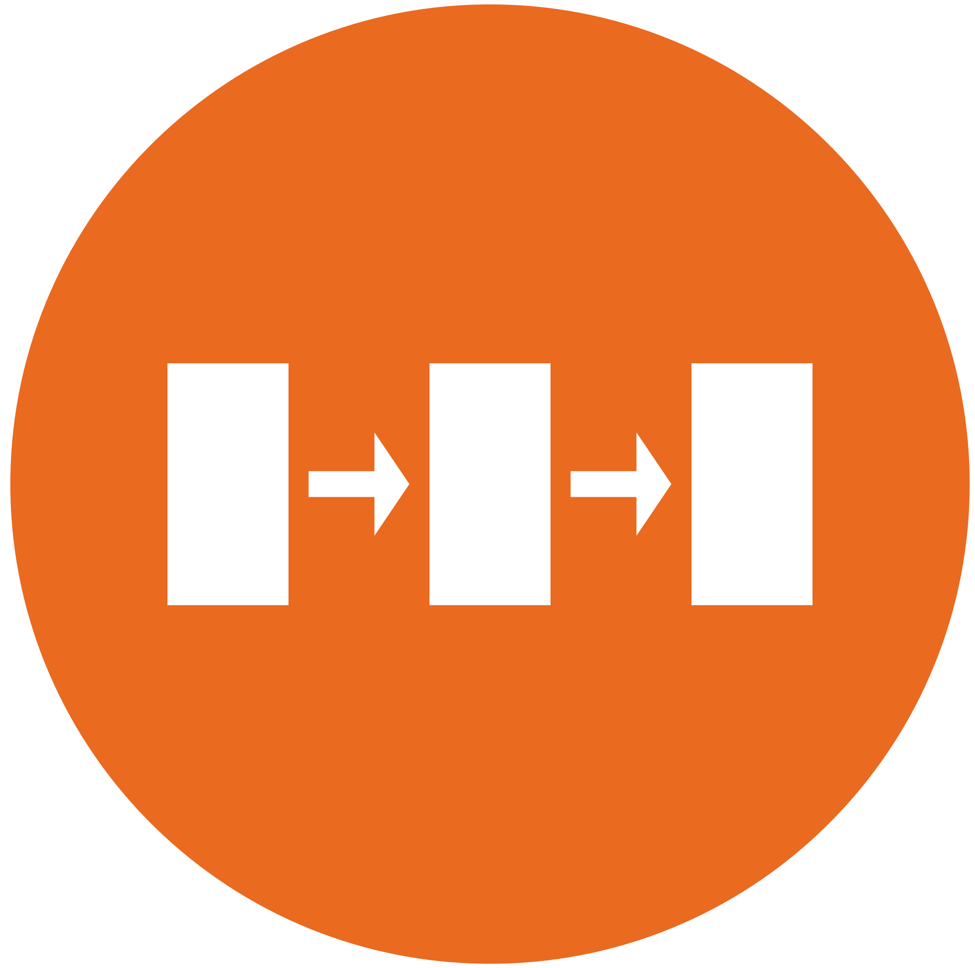



**Figure 3. The structural equation model for the data of the experiment. Significance levels: *** $p < .001$, ** $p < .01$, 'ns' $p > .05$. $R^2$ is the proportion of variance explained by the model. Numbers on the arrows (and their thickness) represent the $\beta$ coefficients (and standard error) of the effect. Factors are scaled to have an SD of 1.**

use **structural equation models**

use correct methods for **non-normal data**



use correct methods for **repeated measures**

# Evaluating Models

An introduction to Structural Equation Modeling

use manipulations and theory to make inferences about **causality**

## Introduction
Welcome everyone!

## Hypotheses
Developing a research model

## Participants
Population and sampling

## Testing A vs. B
Experimental manipulations

## Analysis
Statistical evaluation of the results

## Measurement
Measuring subjective valuations

## Evaluating Models
An introduction to Structural Equation Modeling


$h_o$
AB
$x \rightarrow y$

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

T H A N K S !

George Bernard Shaw

# Resources

Slides and data:

www.usabart.nl/QRMS

Class slides (more detailed)

www.usabart.nl/eval

Handbook chapter:

bit.ly/userexperiments

Framework:

bit.ly/umuai

# Resources

Questions? Suggestions? Collaboration proposals?

Contact me!

Contact info

E: bartk@clemson.edu

W: www.usabart.nl

T: @usabart