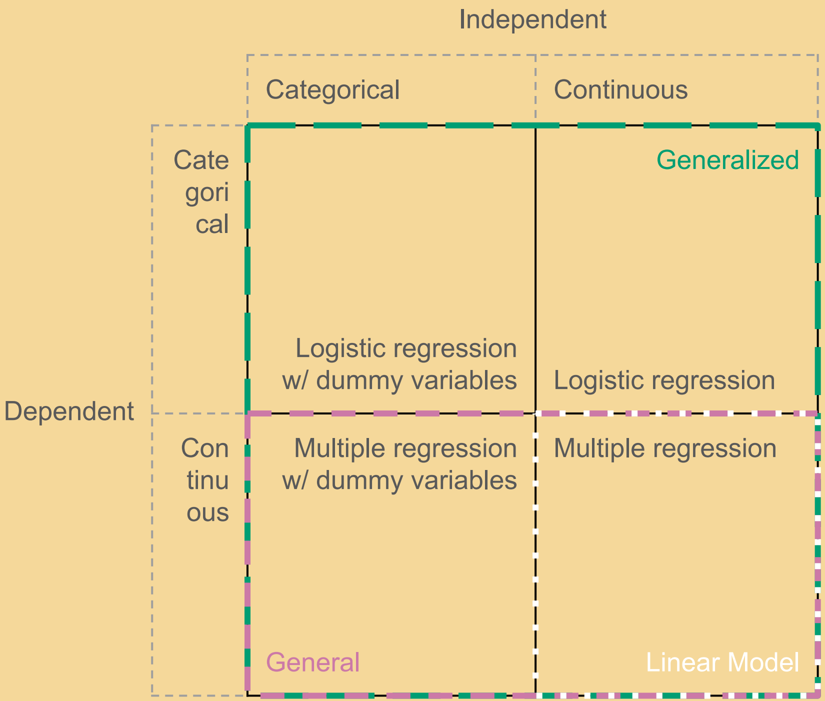
Multilevel & Longitudinal Analysis

for dependent observations

👨‍🏫 Statistical Reasoning Lecture #5
Alexander Savi, 2025
📚 Mehmetoglu & Mittner Ch. 12

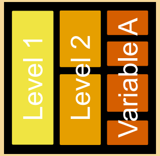
Orange by Ijeamaka Anyene 🎨

🔁 Recap



Independent data

Dependent data



🔮 Today

Topics

1 | Statistical reasoning with GLM
2 | Multiple linear regression
3 | Dummy-variable regression
4 | Logistic regression
5 | Multilevel and longitudinal analysis
    5.1 | Hierarchical analysis
    5.2 | 🚀 Polynomial regression
6 | Statistics superpowers
7 | Bayesian statistics

Examples

📙 Depression (longitudinal)
🧑‍💻 Depression (longitudinal)
👨‍🏫 Danceability (cross-sectional)

Learning goals

Hierarchical Analysis

for dependent observations

Orange by Ijeamaka Anyene 🎨

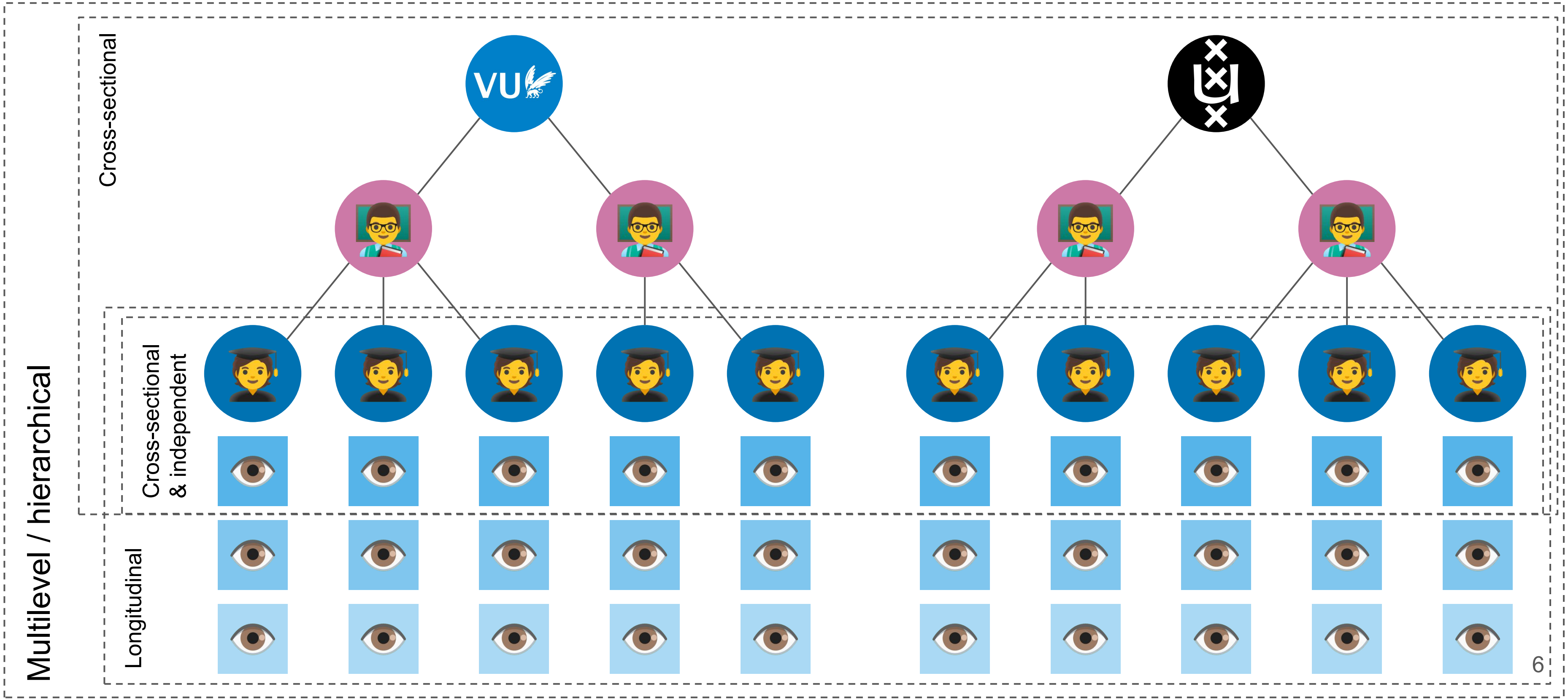
🎲 Wisdom of the crowd?

📏 How tall am I?

📅 How old am I?

🫶 [Social desirability bias]
⚓ [Anchoring effect]

Dependent observations



Danceability

"Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable."

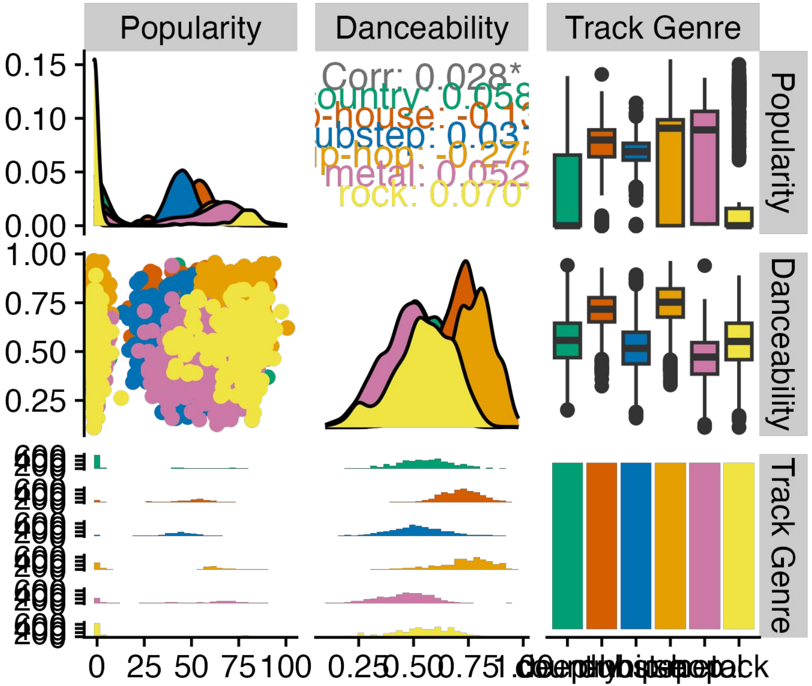
— Spotify

- ❏ Does a hierarchical model make sense?

How are the danceability and popularity of tracks related?

Data

```
library("moderndive")
data("spotify_by_genre")
ggpairs(spotify_by_genre,
        mapping = aes(color = track_genre))
```



Complete pooling

🦹 *Musical genres imprison the creativity of humankind. We should model it without such artificial limitations:*

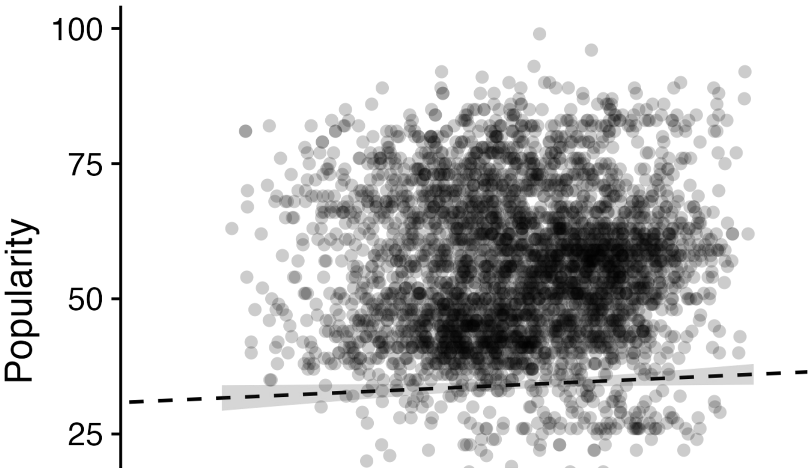
```
Call:
lm(formula = mod, data = spotify_by_genre)

Residuals:
    Min      1Q  Median      3Q     Max 
-36.034 -33.353   8.325  24.205  64.727 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   31.084      1.463  21.253   <2e-16 ***
danceability   5.134      2.404   2.136   0.0328 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 29.07 on 5998 degrees of freedom
Multiple R-squared:  0.0007598, Adjusted R-squared:  0.0005932 
F-statistic: 4.561 on 1 and 5998 DF,  p-value: 0.03275
```

```
mod <- popularity ~ danceability
fit <- lm(mod, data = spotify_by_genre)
```



Way to go, that's what we call complete pooling, or "naive regression". Naivety may fit you well, but it's ***bad for the fit of the model****. What if the relation differs across genres?* 🦸

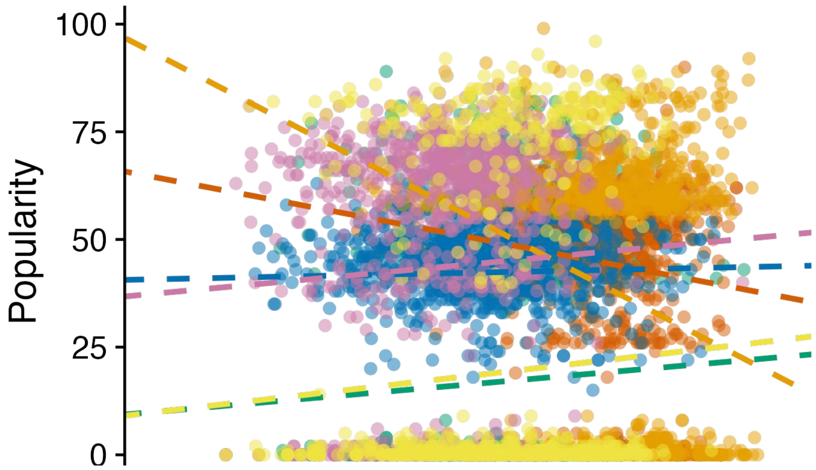
No pooling

🧛 *You win, I'll fit a separate model for each genre. Or, equivalently, an interaction model where the slopes are allowed to vary:*

```
track_genre danceability.trend   SE   df lower.CL upper.CL
country                   12.7 6.71 5988   -0.455     25.8
deep-house               -27.6 9.02 5988  -45.275     -9.9
dubstep                    3.0 6.59 5988   -9.921     15.9
hip-hop                  -75.1 7.01 5988  -88.865    -61.4
metal                     13.5 7.20 5988   -0.619     27.6
rock                      16.7 6.15 5988    4.653     28.8

Confidence level used: 0.95
```

```
mod <- popularity ~ danceability * track_genre
fit <- lm(mod, data = spotify_by_genre)
emmeans(fit, "track_genre", at = list(danceability
= 0)) # genre-specific intercepts
emtrends(fit, "track_genre", var = "danceability")
# genre-specific slopes
```

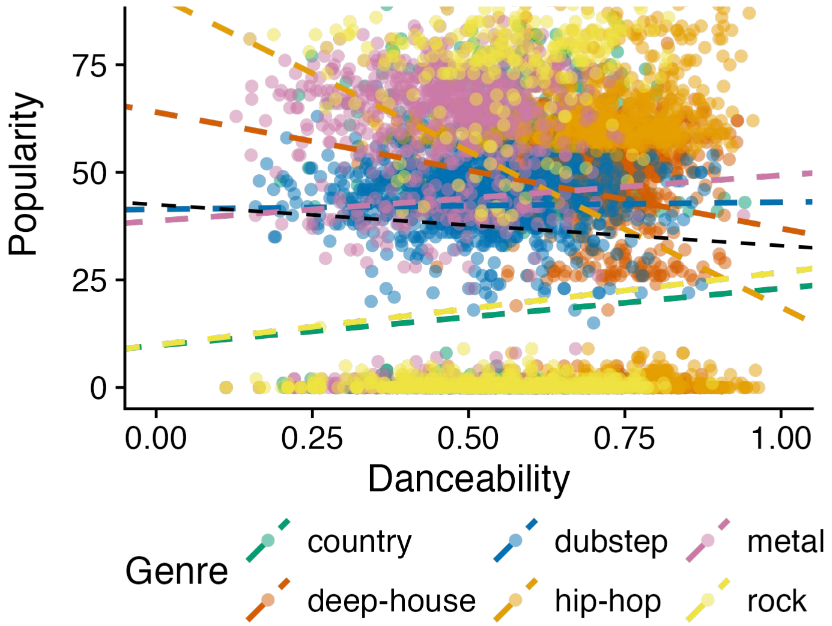


Clever, but it's ***an answer to a different question*** *and it* ***comes at the cost of statistical power****. While the relationships are permitted to differ, they can no longer benefit from what they share. What's in between?* 🦸

Partial pooling

$$
\begin{aligned}
Y_{i,j} &= \beta_0 + \beta_1 X_{i,j} + u_i + v_i X_{i,j} + \varepsilon_{i,j} \\
&= (\beta_0 + u_i) + (\beta_1 + v_i) X_{i,j} + \varepsilon_{i,j} \\
u_i &\sim \text{Normal}(0, \sigma_u) \\
v_i &\sim \text{Normal}(0, \sigma_v).
\end{aligned}
$$

Look, this is what we're aiming for 🦸

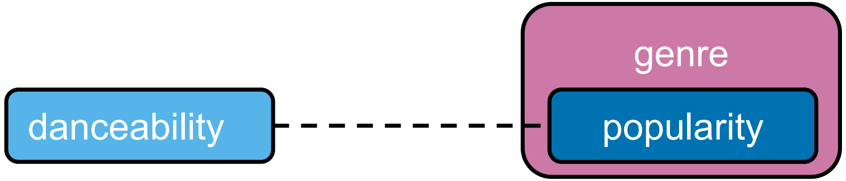


Fixed effect models the average relationship.

Random effects model the genres' deviations from the average, coming from a group-level normal distribution.

Model thinking

How are the danceability and popularity of tracks related?



$$
\begin{aligned}
Y_{i,j} &= \beta_0 + \beta_1 X_{i,j} + u_i + v_i X_{i,j} + \varepsilon_{i,j} \\
&= (\beta_0 + u_i) + (\beta_1 + v_i) X_{i,j} + \varepsilon_{i,j}
\end{aligned}
$$

```
mod <- popularity ~ danceability + (1 +
danceability | track_genre)
library("lmerTest")  # don't use lme4
fit <- lmer(mod, data = spotify_by_genre)
```

- Fixed: `popularity ~ danceability`
- Random: `1 + danceability | track_genre)`
  - (what is nested | in what)
  - intercept + slope

Results

```
summary(fit)
```

```
Linear mixed model fit by REML. t-tests use Satterthwaite's method ['lmerModLmerTest']
Formula: mod
   Data: spotify_by_genre

REML criterion at convergence: 56338.6

Scaled residuals:
     Min       1Q   Median       3Q      Max
-2.35235 -0.69640 -0.01326  0.67670  2.81848

Random effects:
 Groups      Name         Variance Std.Dev. Corr
 track_genre (Intercept)  1014.8   31.86
             danceability 1250.9   35.37    -0.94
 Residual                  696.3   26.39
Number of obs: 6000, groups:  track_genre, 6

Fixed effects:
             Estimate Std. Error     df t value Pr(>|t|)
(Intercept)    42.477     13.131  4.971   3.235   0.0233 *
danceability   -9.522     14.731  5.014  -0.646   0.5464
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:
            (Intr)
danceabilty -0.936
```

```
lme4::lmer(fit); lmerTest::lmer(fit)
```

Fixed effect

Intercept: average popularity where danceability is 0, before genre offset

Slope: average decrease in popularity when danceability increases by 1, before genre offset (divide by 10 for danceability increase by .1) > not significant

Random effects

Intercept + slope: fluctuations of individual coefficients (for genres), summarized by standard deviations (mean = 0)

Evaluation | Confidence intervals

```
Random effects:
 Groups      Name        Variance Std.Dev. Corr
 track_genre (Intercept) 1014.8   31.86
             danceability 1250.9  35.37    -0.94
 Residual                 696.3   26.39
Number of obs: 6000, groups:  track_genre, 6

Fixed effects:
             Estimate Std. Error     df t value Pr(>|t|)
(Intercept)    42.477     13.131  4.971   3.235   0.0233 *
danceability   -9.522     14.731  5.014  -0.646   0.5464
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
confint(fit, method = "boot", level = .95, nsim =
1000)
```

```
                   2.5 %      97.5 %
.sig01        11.5578612  52.5127103
.sig02        -0.9987584  -0.6455147
.sig03        13.4642573  56.9573436
.sigma        25.9508460  26.8382483
(Intercept)   15.7986835  69.6890800
danceability -39.5344842  19.0832627
```

- Does it make sense to include the random intercept and random slope?
- Is there an average effect of danceability?

Evaluation | Fixed & random effects

```
fixef(fit); ranef(fit); coef(fit)
```

🧛 *Miró, these coefficients look an awful lot like the coefficients from my interaction analysis, don't they?*

```
$`(Intercept)`
[1] 42.47747

$danceability
[1] -9.522212

$track_genre
             (Intercept) danceability
country       -32.753747     22.81309
deep-house     21.501598    -17.51555
dubstep        -1.064435     11.17599
hip-hop        48.654123    -62.94547
metal          -3.749130     20.08089
rock          -32.588409     26.39105

$track_genre
             (Intercept) danceability
country         9.723725    13.290874
deep-house     63.979070   -27.037764
dubstep        41.413037     1.653777
hip-hop        91.131595   -72.467678
metal          38.728342    10.558677
rock            9.889063    16.868840
```

```
 track_genre danceability.trend   SE   df lower.CL upper.CL
 country                   12.7 6.71 5988   -0.455     25.8
 deep-house               -27.6 9.02 5988  -45.275     -9.9
 dubstep                    3.0 6.59 5988   -9.921     15.9
 hip-hop                  -75.1 7.01 5988  -88.865    -61.4
 metal                     13.5 7.20 5988   -0.619     27.6
 rock                      16.7 6.15 5988    4.653     28.8

Confidence level used: 0.95
```

Yes, it's referred to as shrinkage, can you see why? 🦸

Multilevel analysis vs. interaction analysis

Interaction analysis

Separate relations for each genre

❌ Dependent observations
❌ Few/varying observations per level
✅ Large heterogeneity across levels
✅ Few groups per level

🧛 *You called me naive? Seems like a good choice after all.*

Multilevel analysis

Average relation given within-genre dependence

✅ Dependent observations
✅ Few/varying observations per level
❌ Large heterogeneity across levels
❌ Few groups per level

You may also interpret the random slopes for the (shrunken, partial-pooled) interaction, if you want to account for the genre-independent relation between danceability and popularity. But here, it might make little sense 🦸

Evaluation | Model comparisons

```
fit_0 <- lm(popularity ~ danceability, data =
spotify_by_genre)
fit_1 <- lmer(popularity ~ danceability + (1 |
track_genre), data = spotify_by_genre)
fit_2 <- lmer(popularity ~ danceability + (1 +
danceability | track_genre), data =
spotify_by_genre)

anova(fit_1, fit_2, fit_0)
```

```
Data: spotify_by_genre
Models:
fit_0: popularity ~ danceability
fit_1: popularity ~ danceability + (1 | track_genre)
fit_2: popularity ~ danceability + (1 + danceability | track_genre)
      npar   AIC   BIC logLik -2*log(L)   Chisq Df Pr(>Chisq)
fit_0    3 57469 57489 -28731     57463
fit_1    4 56468 56495 -28230     56460 1002.92  1  < 2.2e-16 ***
fit_2    6 56362 56403 -28175     56350  109.31  2  < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

- ❑ What’s the best fitting model?
- ❑ Why?

Across genres, what do the relationships between danceability and popularity have in common (fixed) and what sets them apart (random). Evidently significant, because there’s a lot that sets them apart: good reason to adjust the average relationship by their genre-dependencies. But, is the average relationship meaningful if there’s so much that sets them apart?

⚠️ In the `anova` function, the first object must be a `lmer` model

Cooling Down

Orange by Ijeamaka Anyene 🎨

Takeaways

If you hate statistics, choose

- ❏ Complete pooling
- ❏ No-pooling
- ❏ Partial pooling
- ❏ Carpooling

Takeaways

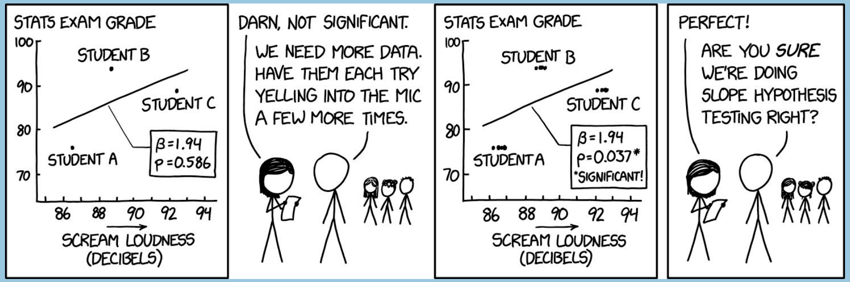


Illustration by [Randall Munroe](wtf) ([wtf]())

🥡 Takeaways

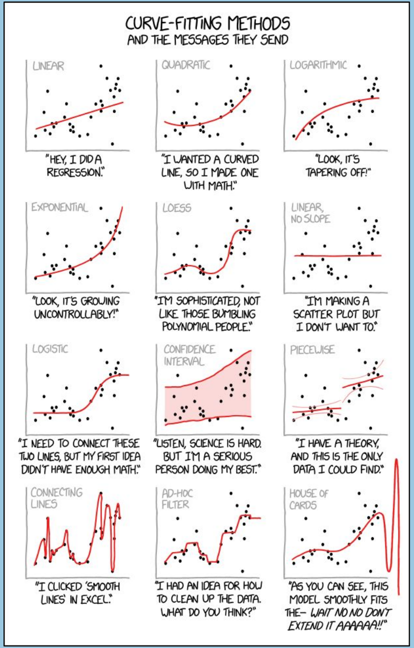


Illustration by [Randall Munroe](https://xkcd.com) ([wtf](https://xkcd.com))

✍️ Colophon

Slides

alexandersavi.nl/teaching/

License