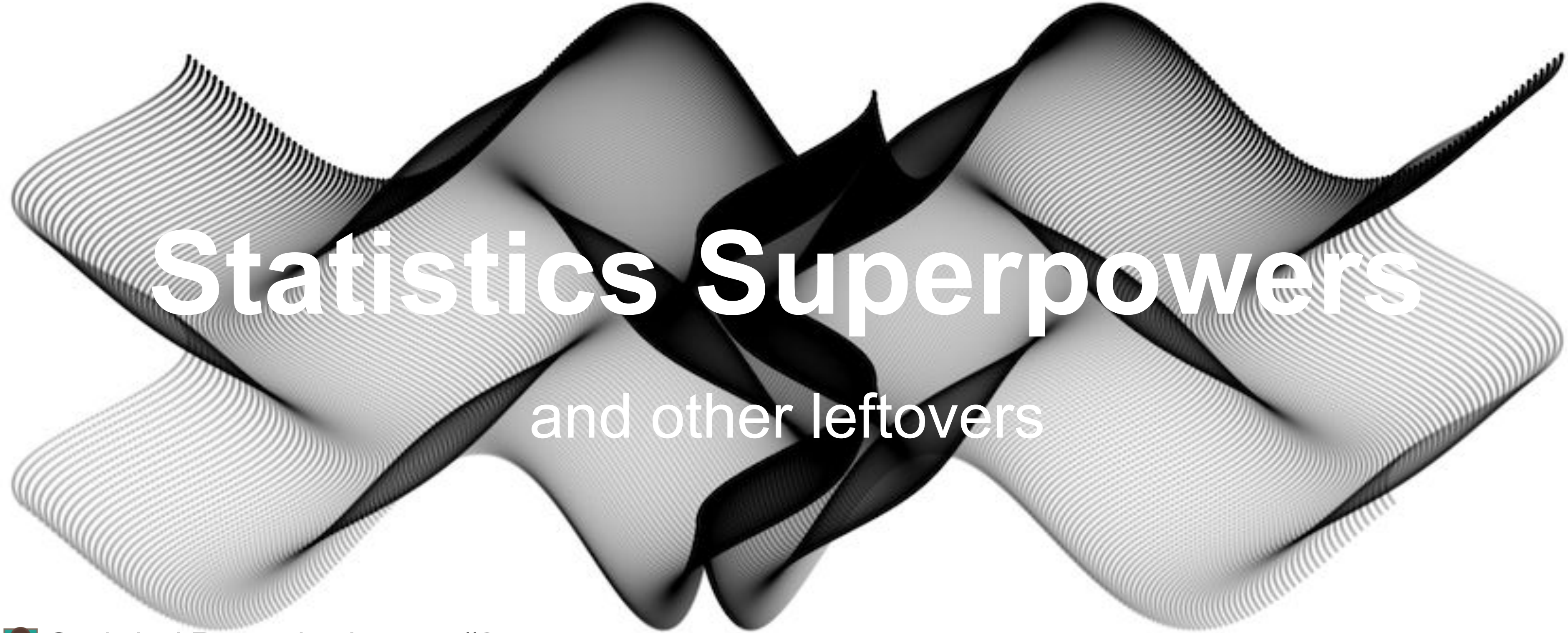
Statistics Superpowers

and other leftovers



👨‍🏫 Statistical Reasoning Lecture #6
Alexander Savi, 2025
📚 None

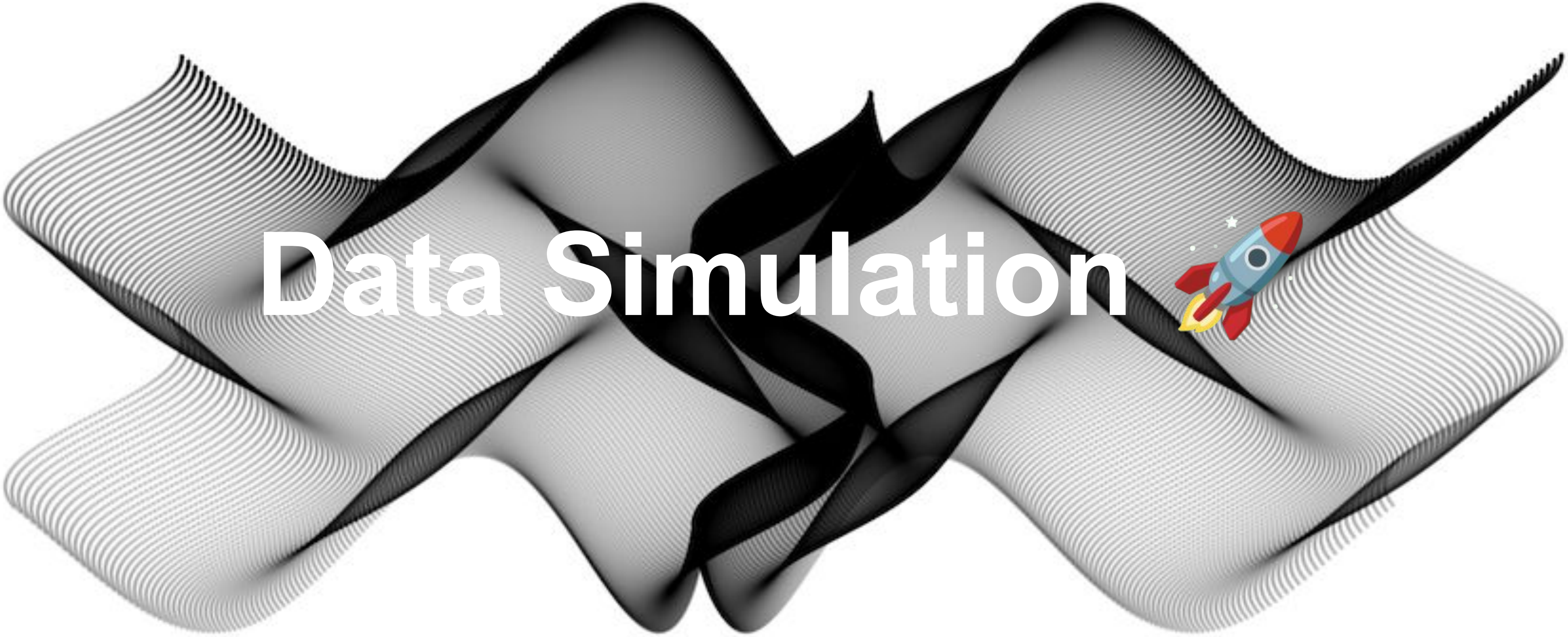
Untitled by Katharina Brunner (generativeart package) 🎨

🔮 Today

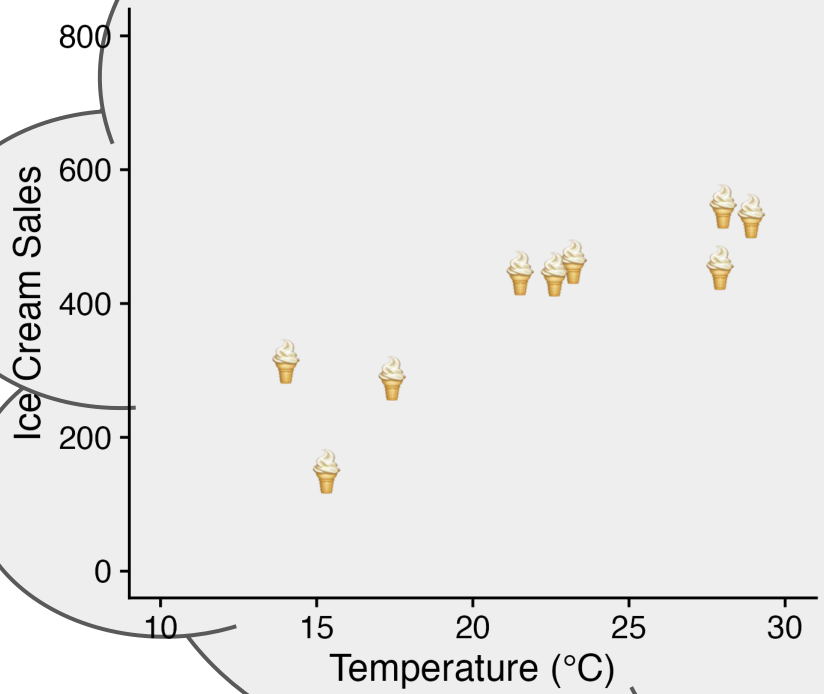
Topics

1 | Statistical reasoning with GLM
2 | Multiple linear regression
3 | Dummy-variable regression
4 | Logistic regression
5 | Multilevel and longitudinal analysis
6 | Statistics superpowers
    6.1 | Data simulation
    6.2 | A priori power analysis
    6.3 | Polynomial regression
    6.4 | Nesting (revisited)
    6.5 | Assumptions
7 | Bayesian statistics

Data Simulation 🚀



Data ~~collection~~simulation



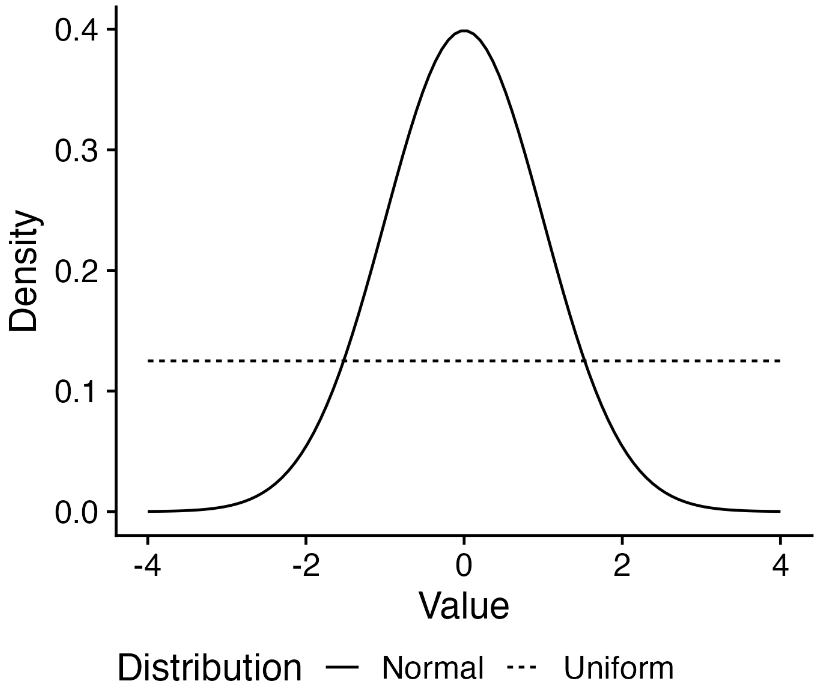

```r
ice_cream_data <- tibble::tribble(
  ~temperature, ~ice_cream_sales,
  27.9,         452,
  15.3,         148,
  17.4,         287,
  21.5,         444,
  28.2,         935,
  14,           312,
  28,           544,
  28.9,         529,
  23.2,         461,
  22.6,         442)
```

$$\text{ice\_cream\_sales} = \alpha + \beta_1(\text{temperature}) + \epsilon$$

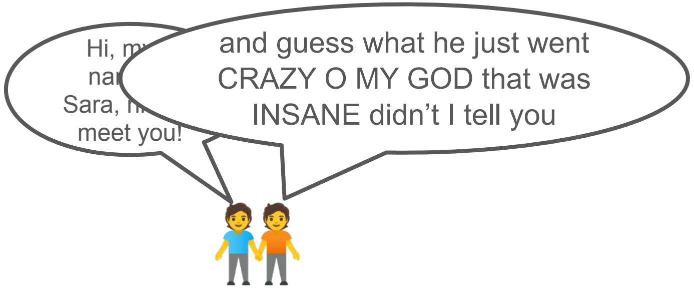
```r
mod <- ice_cream_sales ~ temperature   # y ~ x
```

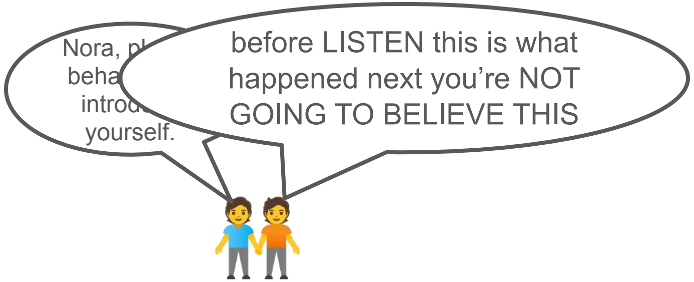
Simulation #1 | Ice cream data

```
set.seed(0)

n <- 10
temperature <- runif(n = n, min = 10, max
= 30)

intercept <- -200
slope <- 30
noise_sd <- 120
ice_cream_sales <- intercept + slope *
temperature + rnorm(n = n, mean = 0, sd =
noise_sd)

data <- tibble(temperature,
ice_cream_sales)
```



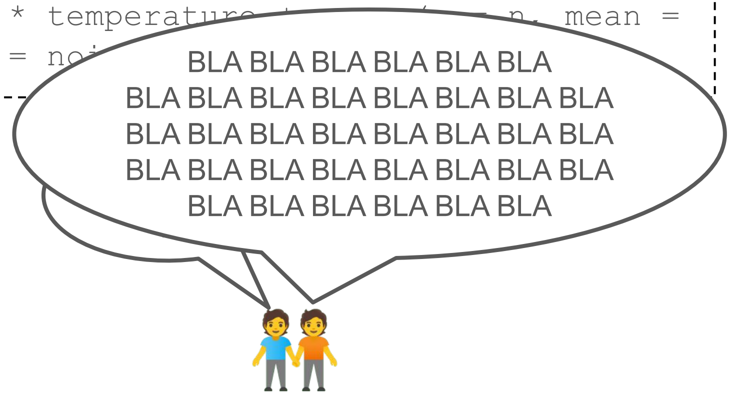
Signal and noise

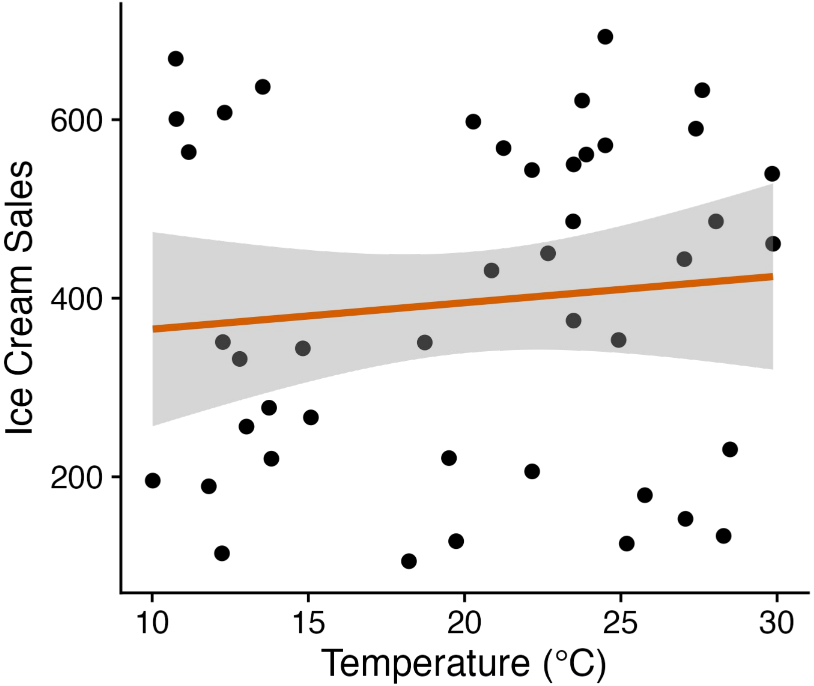




Simulation #2 | High noise, low noise

```
noise_sd_high <- 500
ice_cream_sales_sd_high <- intercept +
slope * temperature + rnorm(n = n, mean =
0, sd = noise_sd_high)

noise_sd_low <- 5
ice_cream_sales_sd_low <- intercept +
slope * temperatur            n, mean =
0, sd = no
```

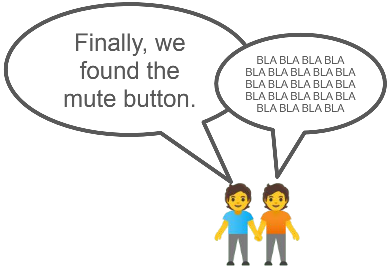


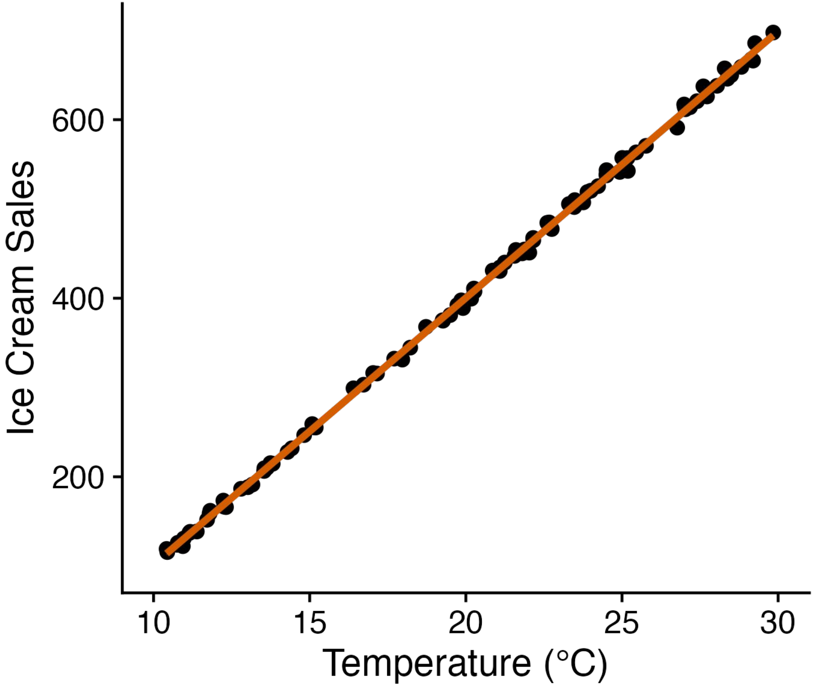


🎵 [Master Peace & Georgia - I Might Be Fake]

Simulation #2 | High noise, low noise

```
noise_sd_high <- 500
ice_cream_sales_sd_high <- intercept +
slope * temperature + rnorm(n = n, mean =
0, sd = noise_sd_high)

noise_sd_low <- 5
ice_cream_sales_sd_low <- intercept +
slope * temperature + rnorm(n = n, mean =
0, sd = noise_sd_low)
```





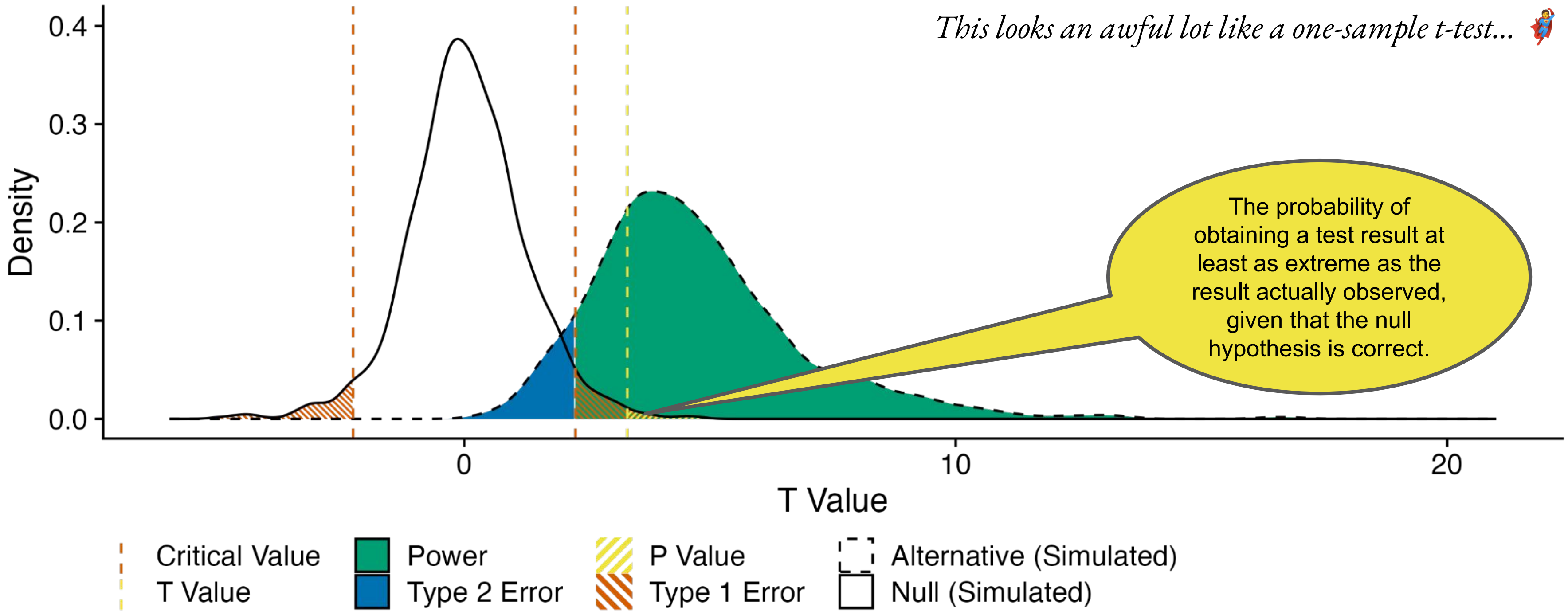
🎵 [Master Peace & Georgia - I Might Be Fake]

Simulation #3 | Frequentist inference (NHST)

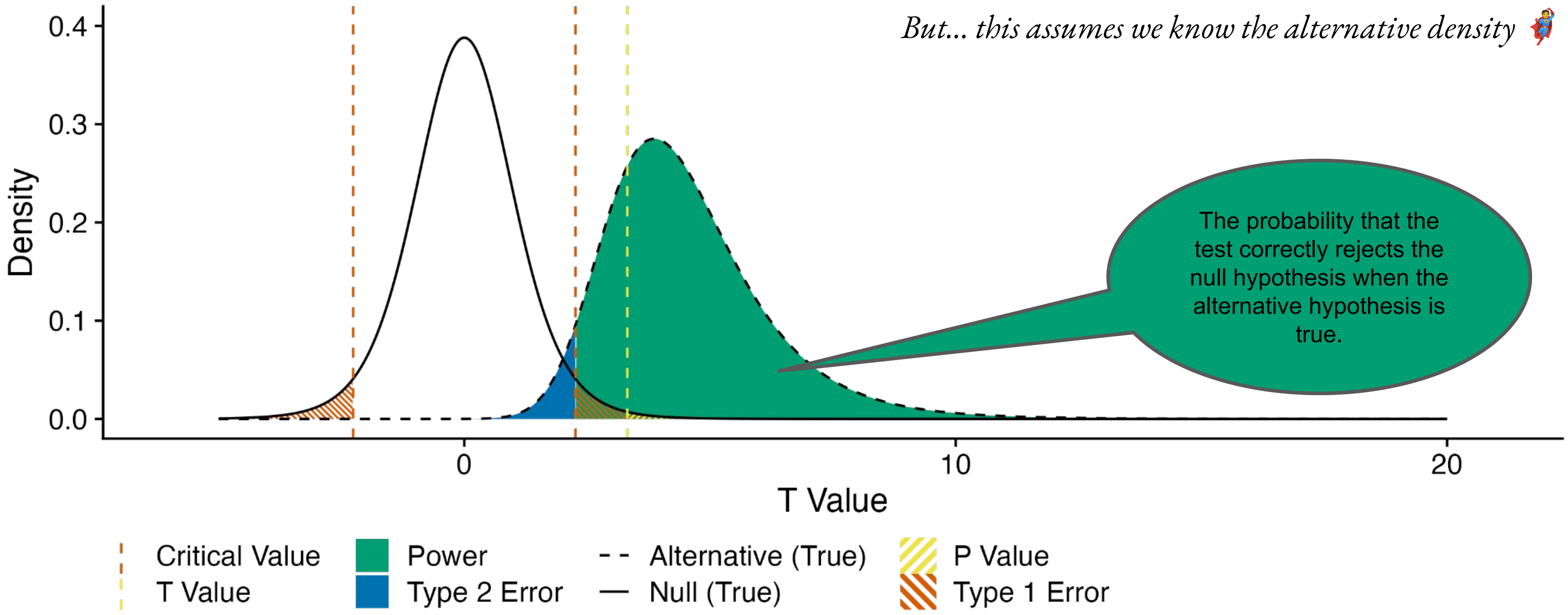
```
n_sim <- 1000
slope_null <- 0
slope_alt <- slope

null_t_stats <- numeric(n_sim)
for (i in 1:n_sim) {
  obs_temperature <- runif(n = n, min = 10, max = 30)
  ice_cream_sales <- intercept + slope_null * obs_temperature
  noise <- rnorm(n = n, mean = 0, sd = noise_sd)

  data <- tibble(
    temperature = obs_temperature,
    sales = ice_cream_sales + noise)

  fit <- lm(sales ~ temperature, data = data)
  null_t_stats[i] <- summary(fit)$coefficients[2, "t value"]
}  # and do the exact same for the alternative slope
```

Simulated densities

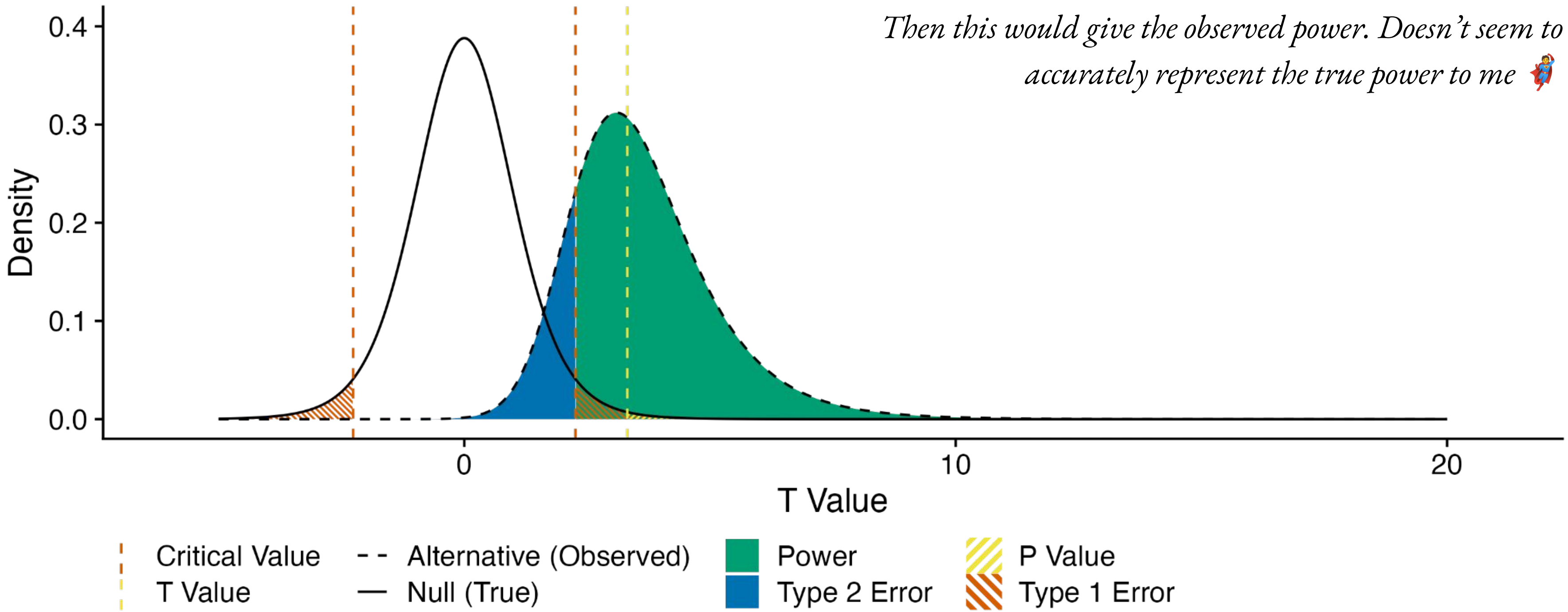
This looks an awful lot like a one-sample t-test… 🦸



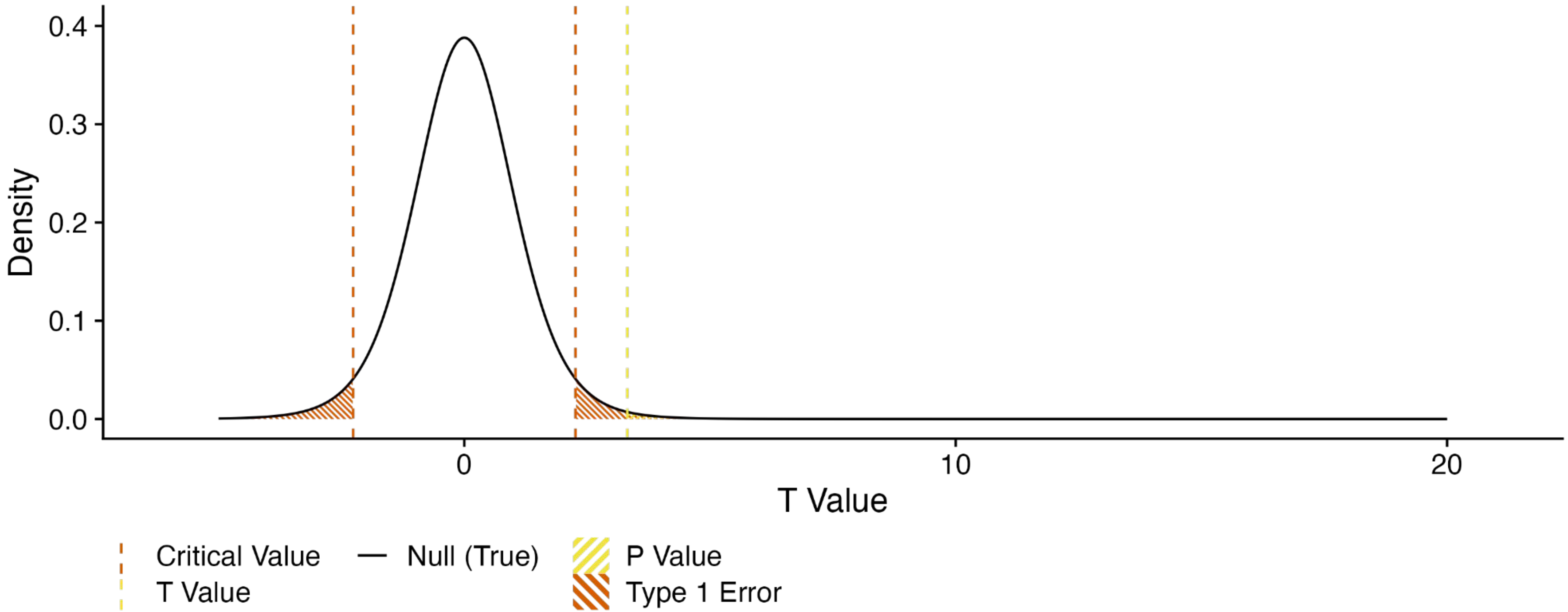
True densities



Observed densities



NHST in practice

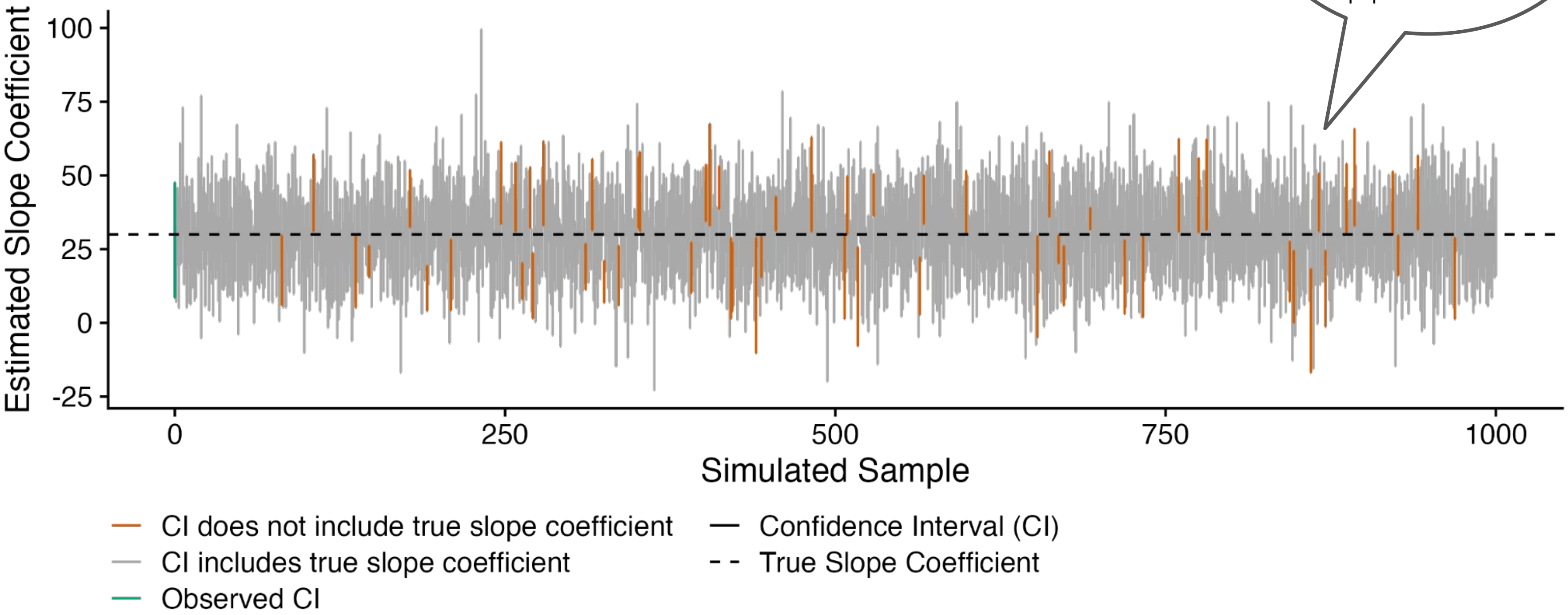


Simulation #3 | Confidence intervals (CI)

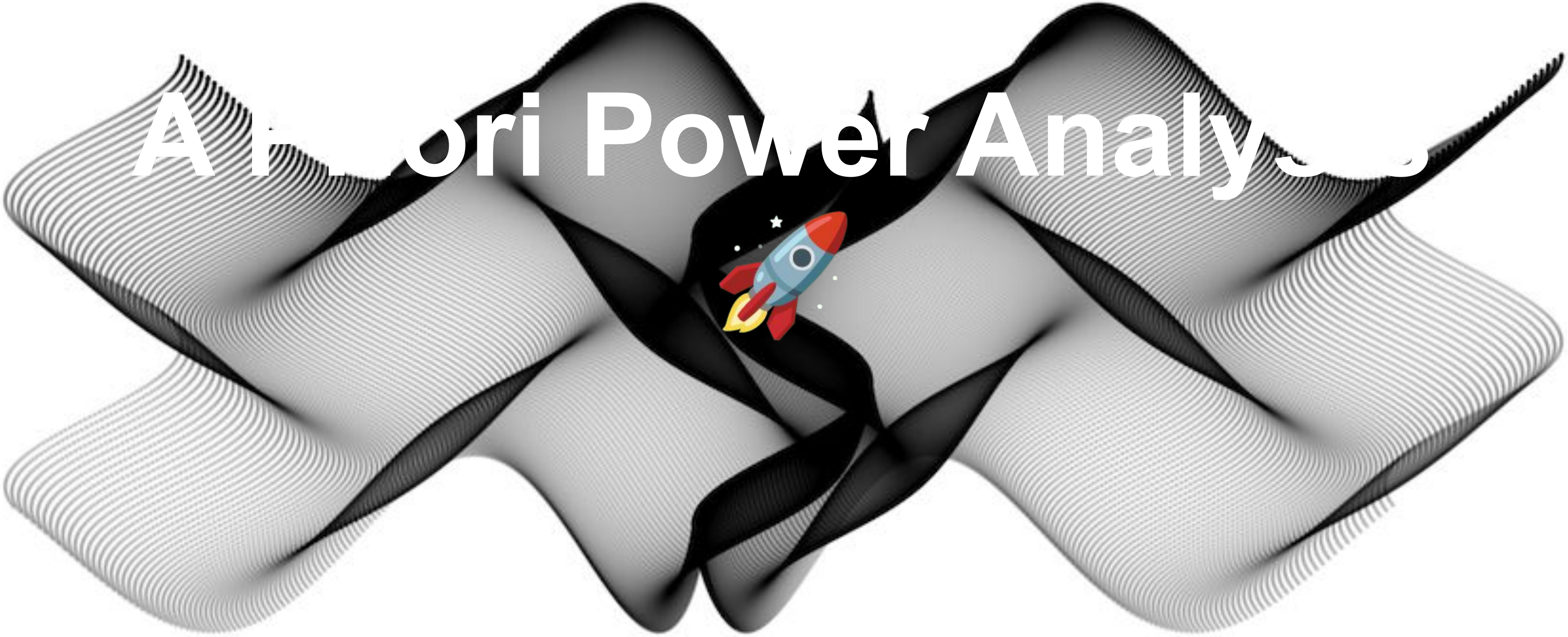
```
n_sim <- 1000
slope_alt <- slope

alt_ci <- matrix(NA, nrow = n_sim, ncol = 2)
for (i in 1:n_sim) {
  obs_temperature <- runif(n = n, min = 10, max = 30)
  ice_cream_sales <- intercept + slope_alt * obs_temperature
  noise <- rnorm(n = n, mean = 0, sd = noise_sd)

  data <- tibble(
    temperature = obs_temperature,
    sales = ice_cream_sales + noise)

  fit <- lm(sales ~ temperature, data = data)
  alt_ci[i, ] <- confint(fit)[2, ]
}
```

Confidence interval

When repeatedly sampling from the target population, 95% of the calculated CIs contain the actual population value.



A Priori Power Analysis



Manipulating statistical power

🦹 *Let's increase* ***α***

And accept more false positives? 🦸

🦹 *Let's increase* ***the magnitude of the effect***

Interesting, but we're not running an experiment here 🦸

🦹 *Let's decrease* ***the noise***

Nora: NOOOOOOOO! I HAVE FEELINGS TOO! 🧑‍🤝‍🧑
Shut up, Nora, I'll use more precise measurements 🦸

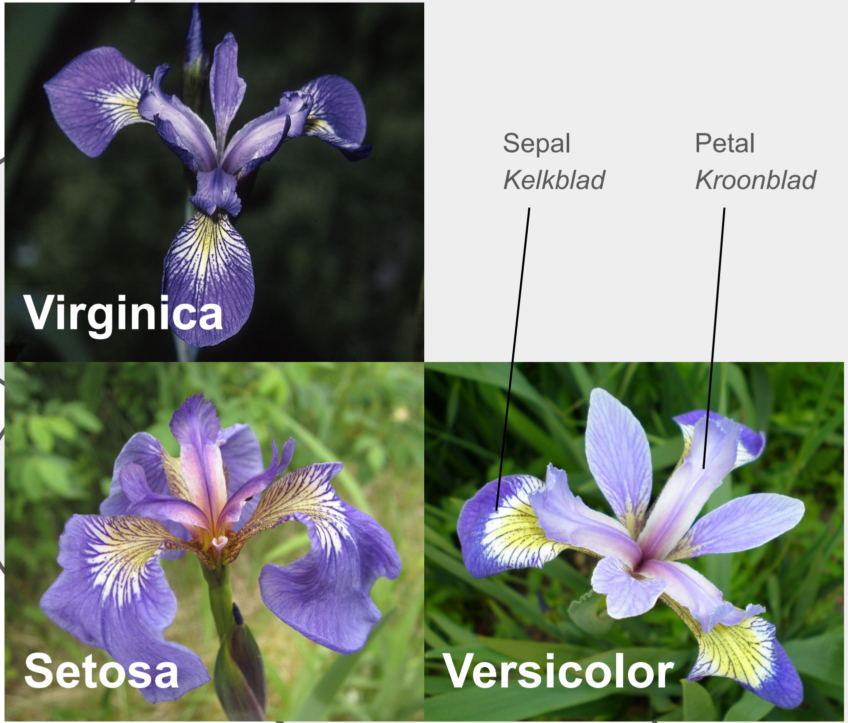
🦹 *Let's use a* ***more powerful test***

I'm already using the [most powerful test] 🦸

🦹 *Let's increase* ***the sample size***

Data collection is costly and takes time, how many observations do I need? 🦸

Iris



Q. Are the dimensions of the petals and sepals of the iris flower related?

H. The length of a petal is related to the length and the width of a sepal.

E. […]

💡 A data set made famous by [Ronald Fisher]() and with its very own [Wikipedia page]().

How many observations do I need?

What effect size do I want to be able to detect?

- ❏ Pick values from multiple robust studies
- ❏ Pick magnitude of practical interest
- ❏ Do not pick from a single noisy study

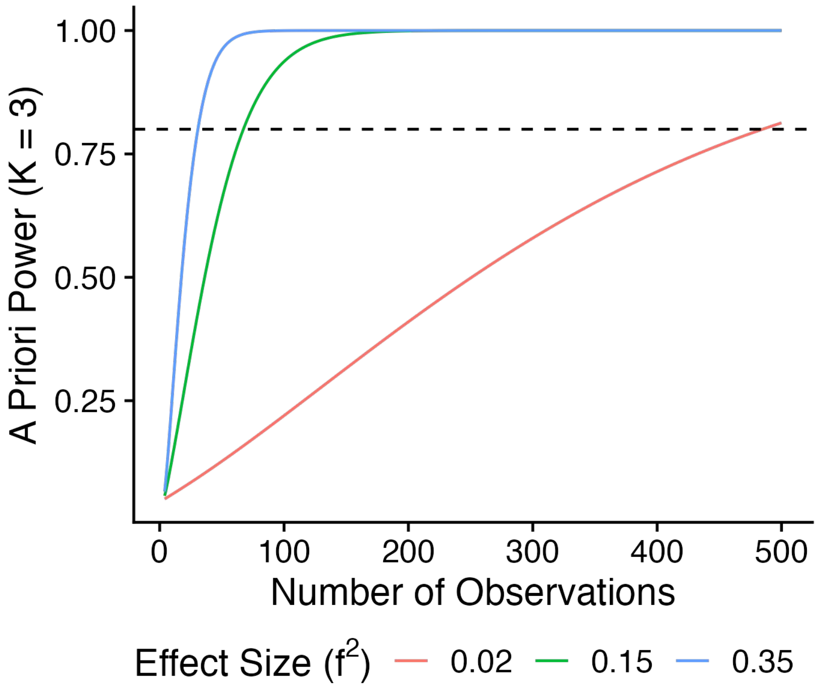
What power do I want to detect an effect?

- ❏ ≥ .8

What's the required significance level?

- ❏ .05

What's the required sample size?



📖 not in book; Gelman, Hill, & Vehtari (2021, p. 293)

How many observations do I need?

What effect size do I want to be able to detect?

- ❏ Pick values from multiple robust studies
- ❏ Pick magnitude of practical interest
- ❏ Do not pick from a single noisy study

What power do I want to detect an effect?

- ❏ ≥ .8

What's the required significance level?

- ❏ .05

What's the required sample size?

- ❏ 67.319

```
K <- 3  # number of variables

library("pwr")
a_priori_power <- pwr.f2.test(
  u = K - 1,
  v = NULL,
  f2 = .15,
  sig.level = 0.05,
  power = 0.8)

ceiling(a_priori_power$v + K)  # required number of observations
```

📖 not in book; Gelman, Hill, & Vehtari (2021, p. 293)

F distribution

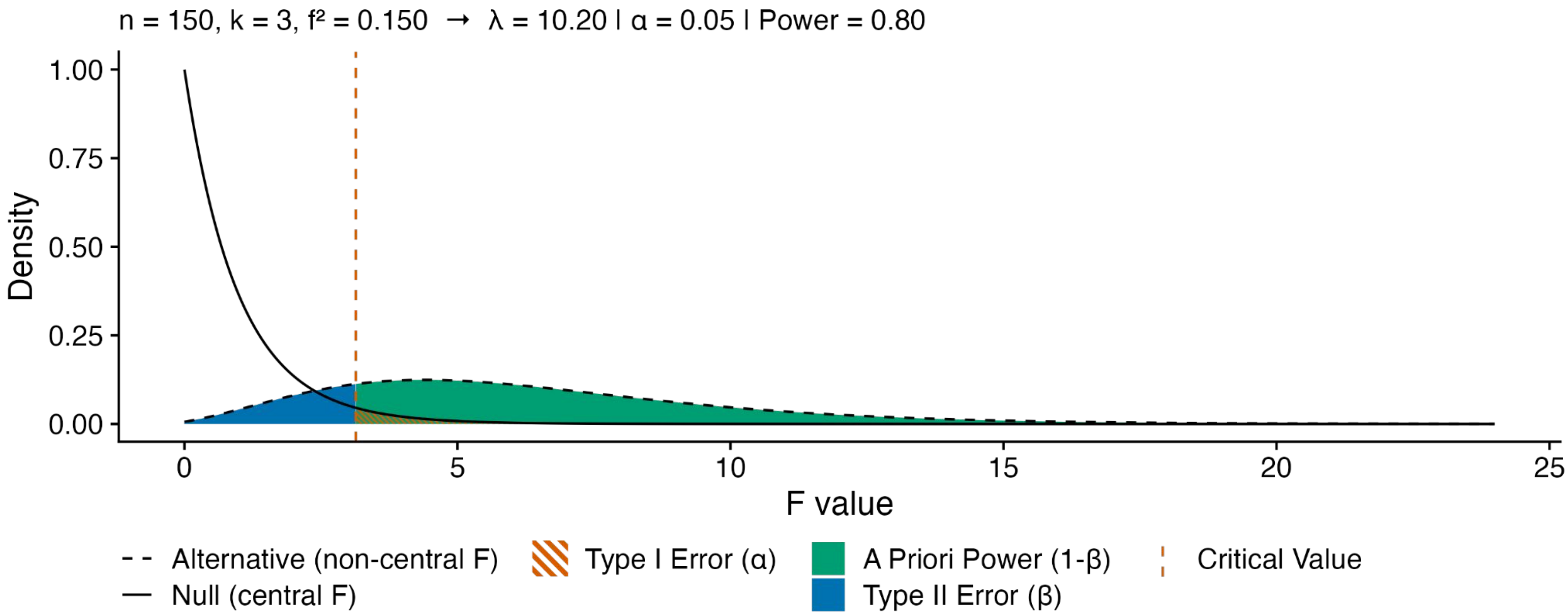


Image Reveal 🎲

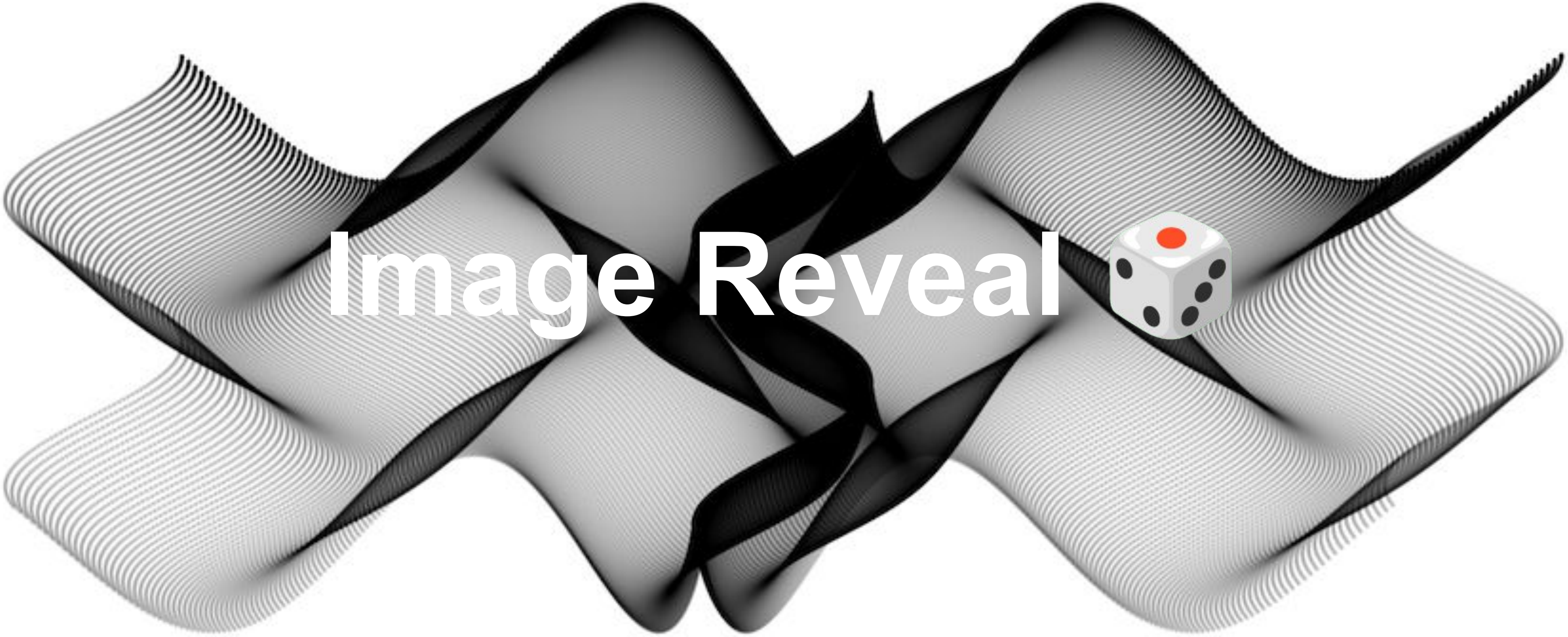
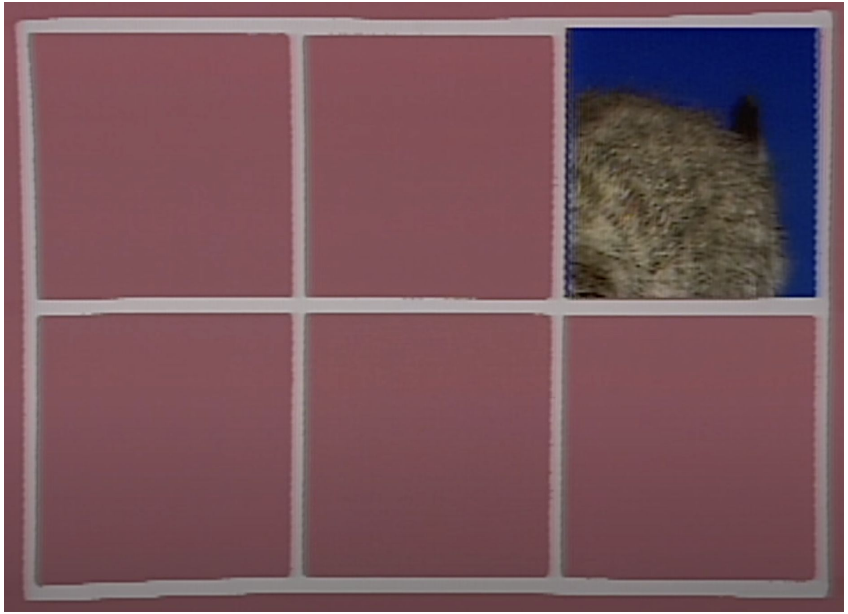


Photo reveal



— Tik Tak

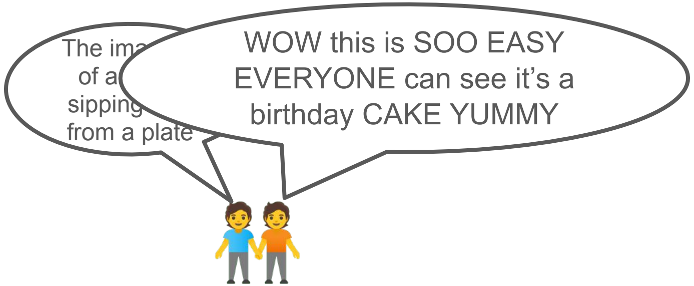
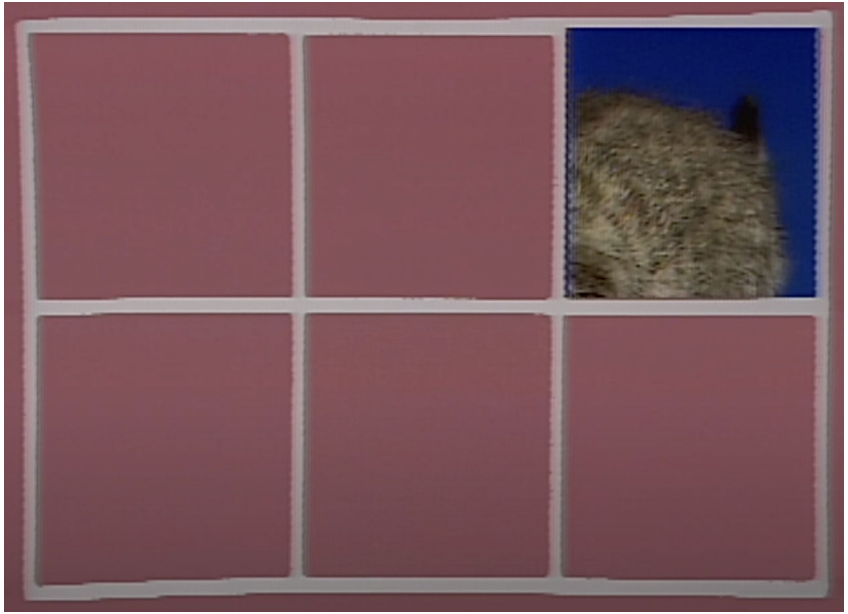


Photo reveal & statistical power



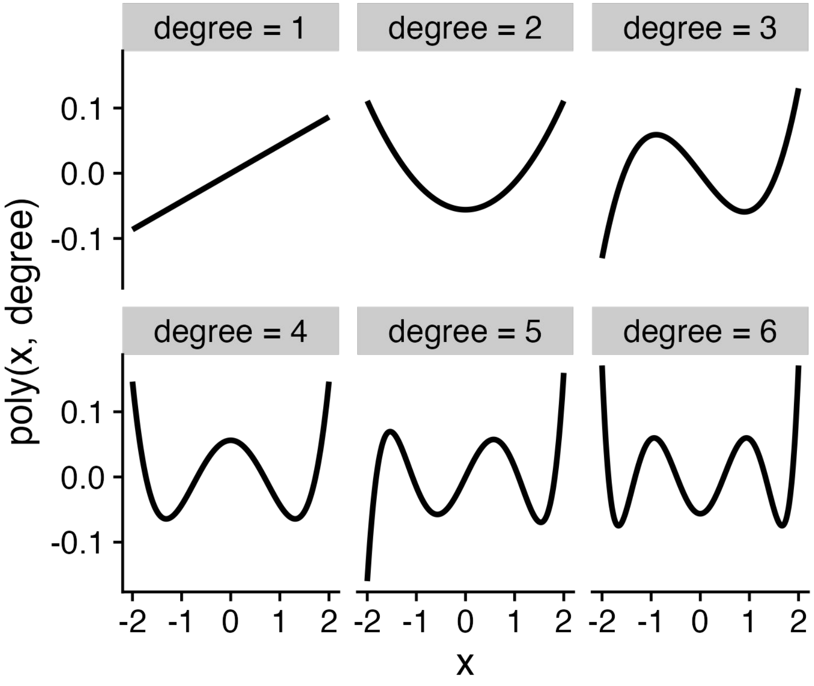
Number of observations

Noise level


Created with R 💡

Polynomial Regression 🚀

Functional form | Nonlinear relationships

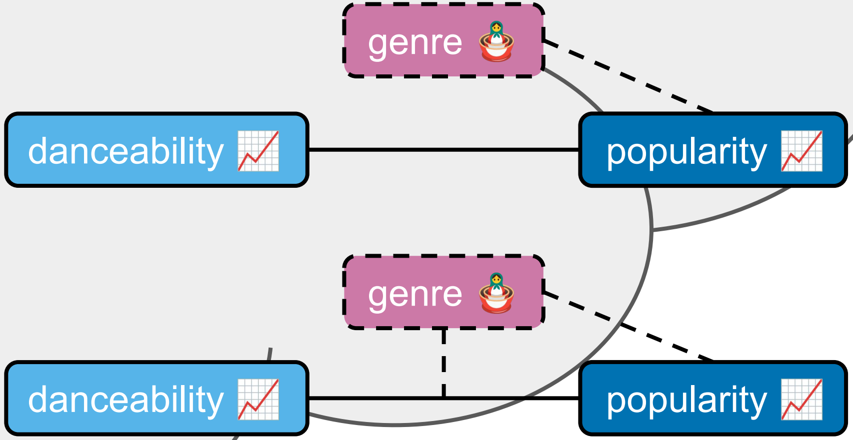


Danceability

"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?
- ❏ What do we expect to be dependent?
  - ❏ Intercept: danceability depends on genre
  - ❏ Slope: the relationship depends on genre
  - ❏ Both

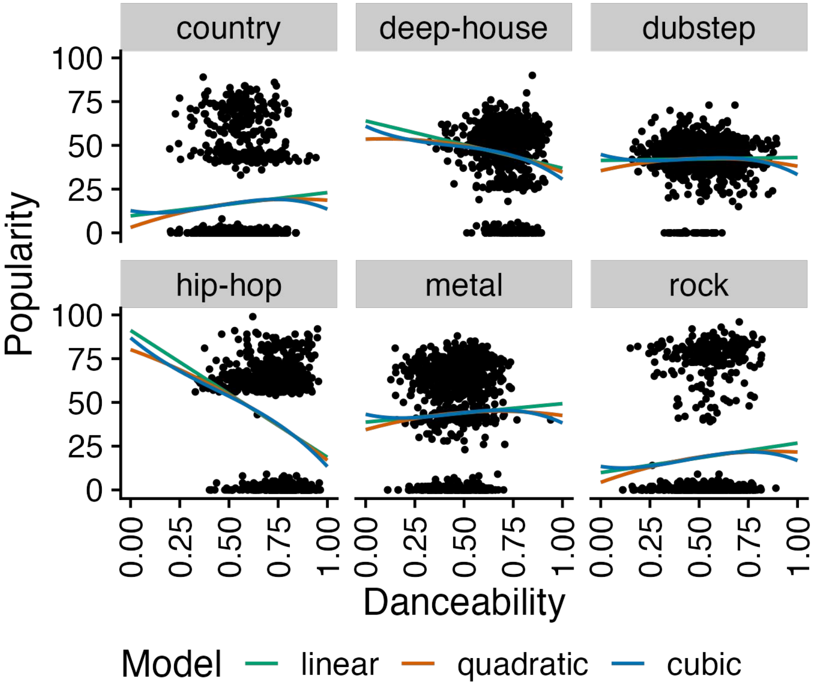


How are the danceability and popularity of tracks related?

Polynomial regression

```
mod_linear <- popularity ~ poly(danceability, 1) +
(1 + danceability | track_genre)
mod_quadratic <- popularity ~ poly(danceability, 2)
+ (1 + danceability | track_genre)
mod_cubic <- popularity ~ poly(danceability, 3) +
(1 + danceability | track_genre)

anova(fit_linear, fit_quadratic, fit_cubic)
```

```
Data: spotify_by_genre
Models:
fit_linear: mod_linear
fit_quadratic: mod_quadratic
fit_cubic: mod_cubic
              npar   AIC   BIC logLik -2*log(L)  Chisq Df Pr(>Chisq)
fit_linear       6 56362 56403 -28175     56350
fit_quadratic    7 56362 56409 -28174     56348 2.1599  1     0.1417
fit_cubic        8 56363 56416 -28174     56347 1.4380  1     0.2305
```



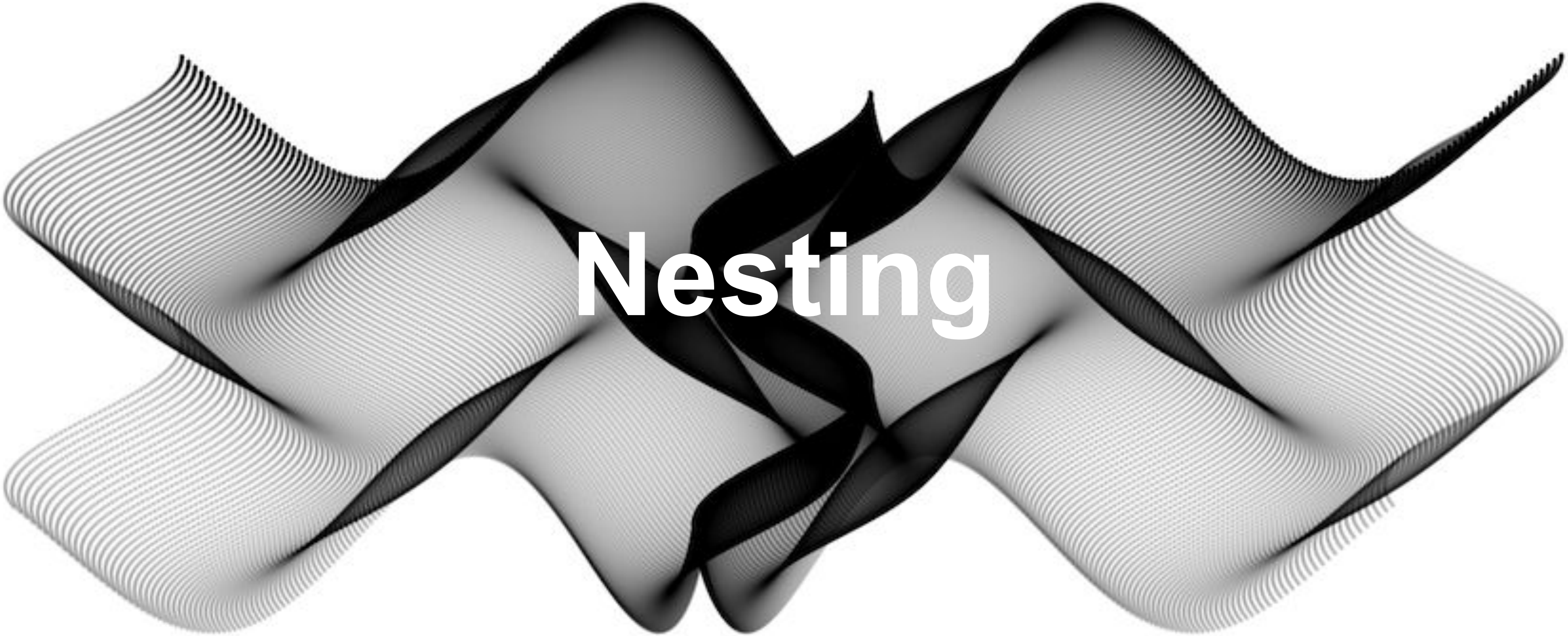
🦸 *(Non-)linearity is not our biggest problem...*

Quote of the week

"All models are wrong, but some are useful."

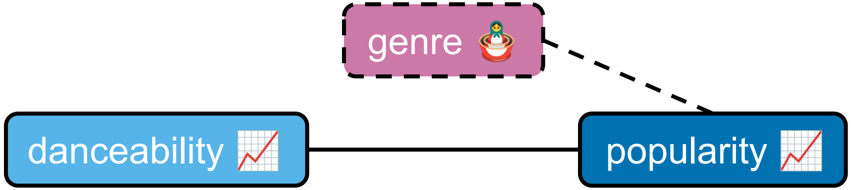
– George Box (1976)

Nesting

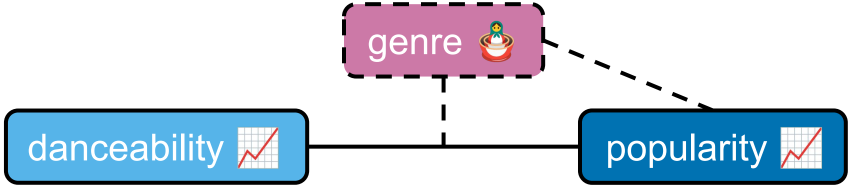


Nested data

Nested intercept



Nested intercept + slope



Nested models

Nested models can be compared

- ❏ Are the parameters of the one model fully contained in the other?
- ❏ Use [insight::is_nested_models()]() (only for multiple regression, ignores random effects parameters)

Nested 🪺 or unnested 🪹?

1. 🐣 `y ~ x1`
   `y ~ x1 + x2`
2. 🍳 `y ~ x1`
   `y ~ x2`
3. 🐣 `y ~ x1`
   `y ~ x2 + x1`
4. 🐣 `y ~ x1`
   `y ~ x1 + x2 + x1:x2`
5. 🍳 `y ~ x1 + x2 + x3`
   `y ~ x1 + x2 + x4 + x5`
6. 🐣 `y ~ 1`
   `y ~ x1`
7. 🍳 `y ~ 1 + x1`
   `y ~ 0 + x1 + x2`

⚠️ The `anova` function doesn't complain if models are not nested.

Assumptions 🚀

“Conducting data analysis is like drinking a fine wine. It is important to swirl and sniff the wine, to unpack the complex bouquet and to appreciate the experience. Gulping the wine doesn’t work.”
— Daniel B. Wright (2003)

Validity

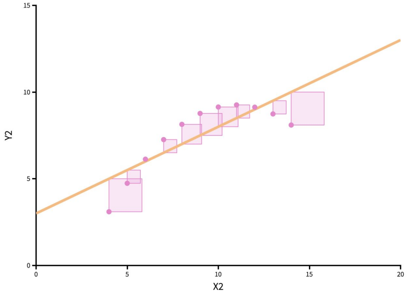
- Outcome reflects phenomenon of interest?
- Inputs are relevant and necessary?
- Sample represents population of interest?

⚠️ Important

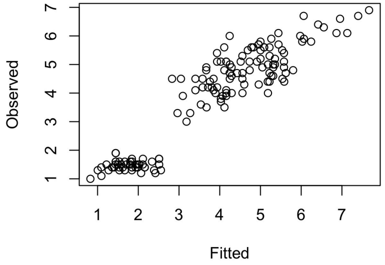
`predict(fit)` # external validity

💡 Prioritization taken from Andrew Gelman.

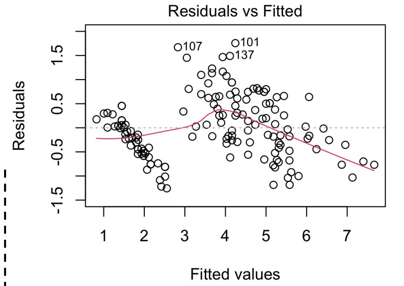
Additivity & linearity




```
plot(x = fitted(fit), y =
iris$Petal.Length)
```




```
plot(x = fit, which = 1)
```



`car::residualPlots(model = fit)` # for each predictor

👍 Approximately linear; horizontal at 0

⚠️ Important

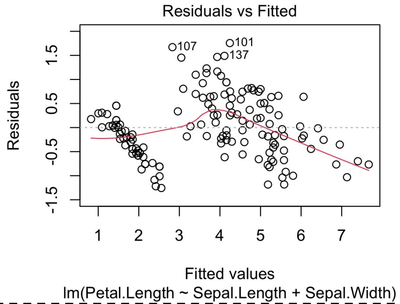
Independence of errors

✅ Independent observations
❌ Dependent observations

We know how to deal with this, do we? 🦸‍♀️

👍 Horizontal
⚠️ Important

```
plot(x = fit, which = 1)
```

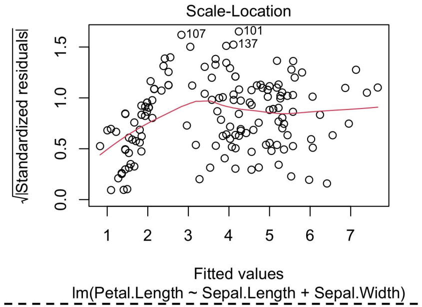


Equal variance of errors

Homogeneity of variance, homoscedasticity. Similar to sphericity in repeated measures ANOVA.

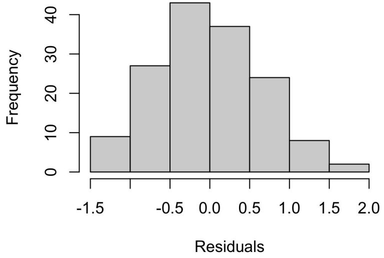
👍 Horizontal
⚠️ Issue with prediction, otherwise minor

```
plot(x = fit, which = 3)
```

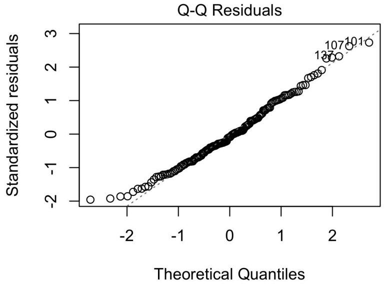


Normality of errors

```r
hist(x = resid(fit))
```




```r
plot(x = fit, which = 2)
```



👍 Approximately normal; linear
⚠️ Issue with predicting individual data points, otherwise not an issue

(Multi)collinearity

👍 Low correlations between predictors; low VIF
⚠️ For explanation, less/not for prediction

```
cor(iris[, c("Sepal.Length",
"Sepal.Width")])
```

```
car::vif(mod = fit)
```

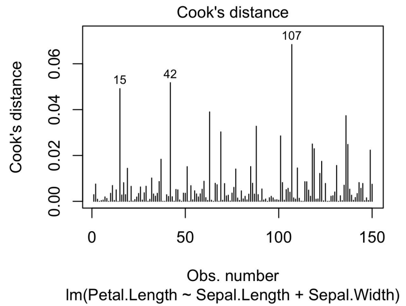
Influential observations

Influential observation = Outlier + Leverage

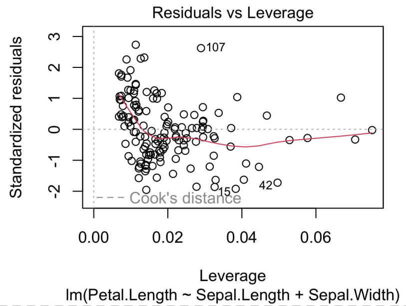
❌ Error
✅ Interesting
✅ Random

👍 Cook's distance < 1 or < 4/N; horizontal at 0
⚠️ It depends

```
plot(x = fit, which = 4)
```




```
plot(x = fit, which = 5)
```

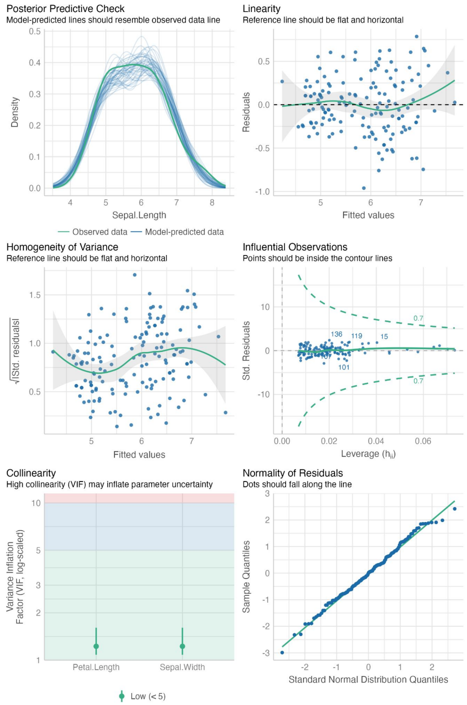


📖 Leys et al. (2019)

🦥 Take it easy

```
library("easystats")
performance::check_model(fit)

astatur::regression.diagnostics
```

💡 [Interpretations and solutions]()



📖 8.1.7 & 8.2

Cooling Down

💯 Exam*

(*course manual is leading)

❓ Questions (SR)

- Open and closed
- Practical and theoretical (learning goals)
- R
- No follow-up (like in weekly assignments)
- From literature & lectures
- Statistics exam, not a programming exam

💯 Points (see course manual)

- Grade = .8 × Exam + .2 × PhS Assignment
- Exam = ⅝ × SR + ⅜ × PhS (must be ≥5.5)
- Exam: 50 points ≈ 75 minutes (SR); 30 p ≈ 45 m (PhS); 10 p ≈ 15 m
- Correction for guessing (applied after exam)

📚 Resources at the exam

- R, RStudio
- Course literature (.pdf)
- Lecture slides (.pdf, SR only, broken links)
- Scrap paper, if needed

🎒 What to bring

- Student ID card, **UvAnetID credentials**
- Pen (no calculator, use R)
- Water, snack

🖥️ At the exam

- Come early
  - Take into account delays (traffic, etc.)
  - Visit the restroom prior to the exam
  - Logging in may take up to 6 minutes
- Don't switch computers
- Open the resources folder, RStudio, ANS exam (*'Start'*, requires password)
- RStudio terminates occasionally: all you can do is start a new session, scripts are almost always recovered
- Late arrivals allowed during first 30 minutes
- Not allowed to leave during first/last 30/15 minutes
- Exam automatically stops after 2 hours *or* at 10 minutes past the official end time—whichever comes first
- When leaving: turn in scrap paper, take ID, be quiet

⛑️ Examination ICT can fail in unimaginable ways, keep it cool

🎵 Merijn Scholten - Had Ik Maar

🥡 Takeaways

Illustration by [Amii Illustrates]

Takeaways

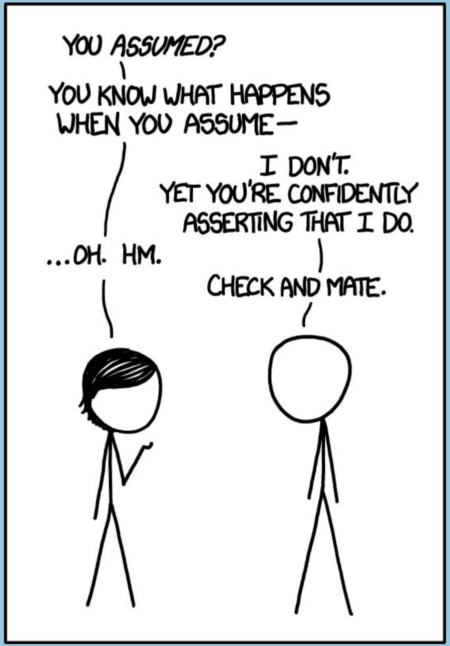


Illustration by [Randall Munroe](#) ([wtf](#))

Takeaways

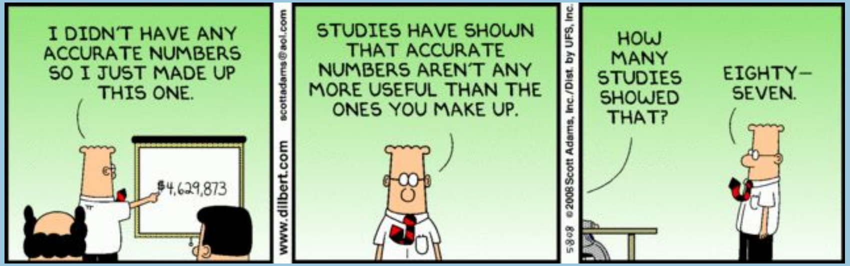


Illustration by Scott Adams

Takeaways

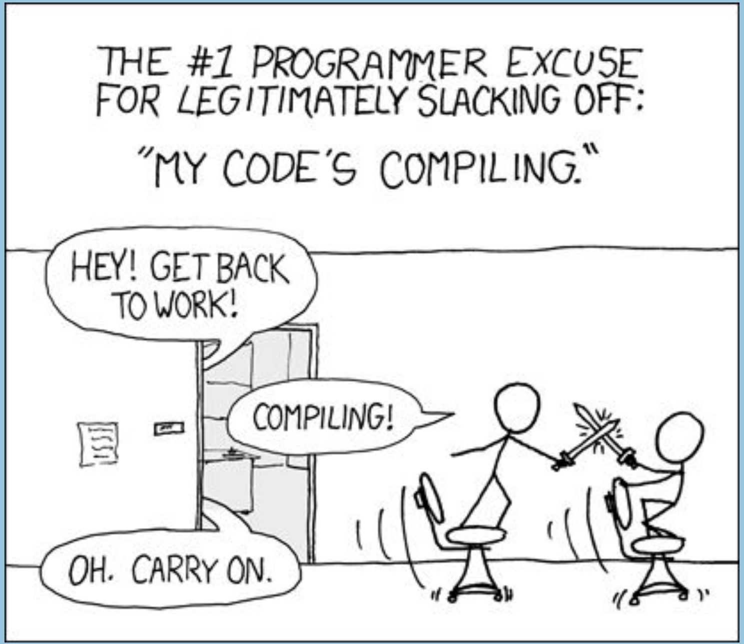


Illustration by [Randall Munroe](#) ([wtf](#))

🥡 Takeaways

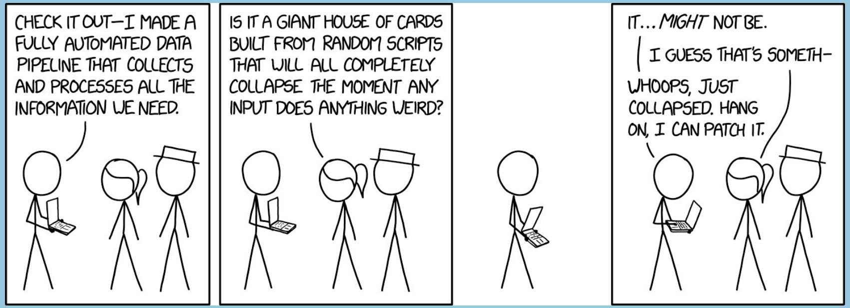


Illustration by [Randall Munroe](#) ([wtf](#))

✍️ Colophon

Slides

alexandersavi.nl/teaching/

License