LEARN TO LOVE R

A LINEAR MODELLING DEMONSTRATION

Hanna Granroth-Wilding
Medical Faculty Biostatistics Unit

SESSION OUTCOMES

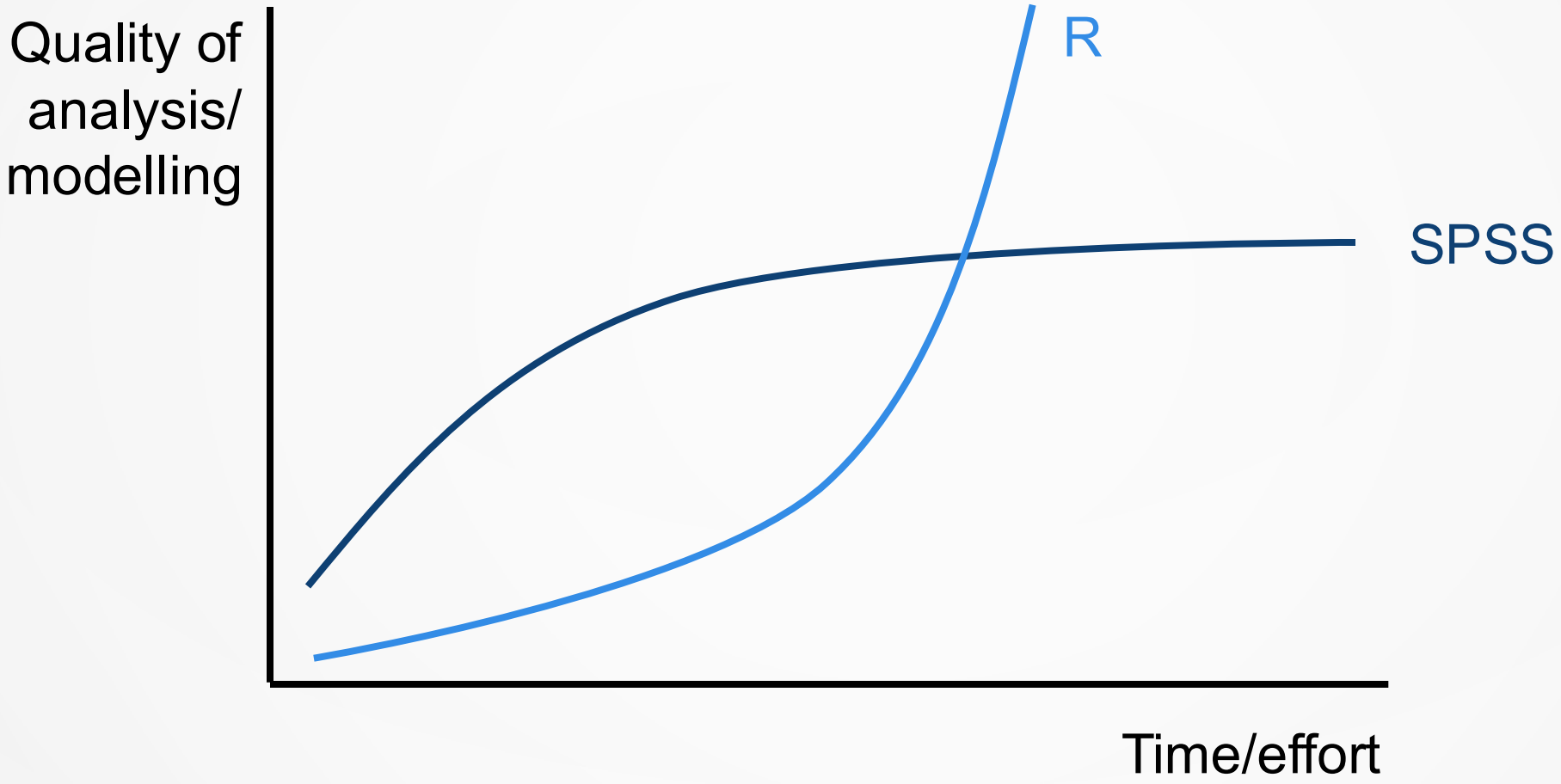
- Use (basic) R as a programming language for your data handling & statistical analysis
- Work through a real-life linear model fitting
  - R syntax
  - checking assumptions
  - Interpretations
  - avoiding common mistakes
- Appreciate how R's flexibility will save you time and enable more sophisticated analysis & plotting

R AS A LANGUAGE

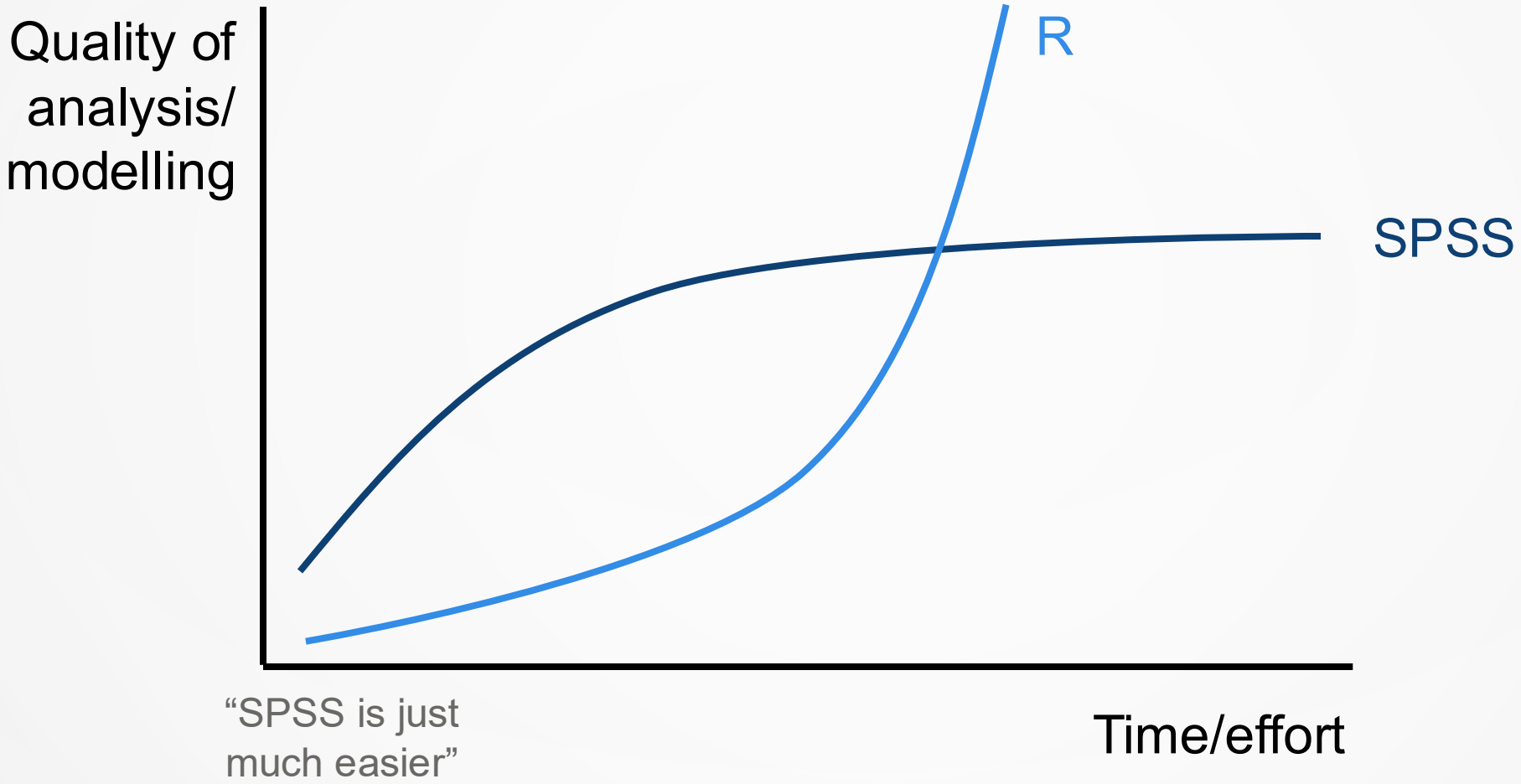
- R is NOT statistics software
  - Programming language that works particularly well for statsitics
- Learning R ≠ learning statistics – practical coding vs. mathematical theory
  - Doing both together is HARD

- Today we'll set the scene for more advanced use
  - code along with the examples

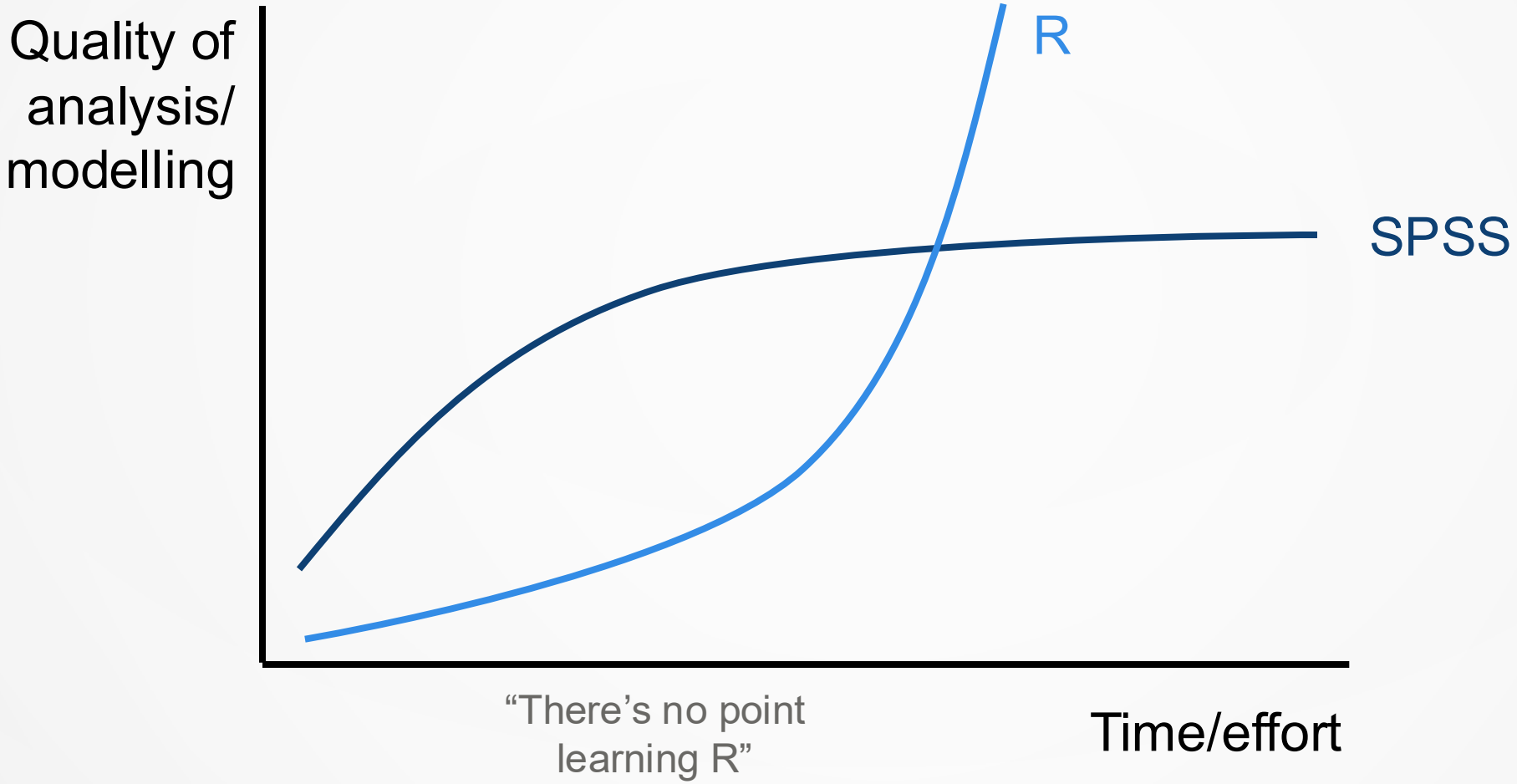
R AS A LEARNING CURVE



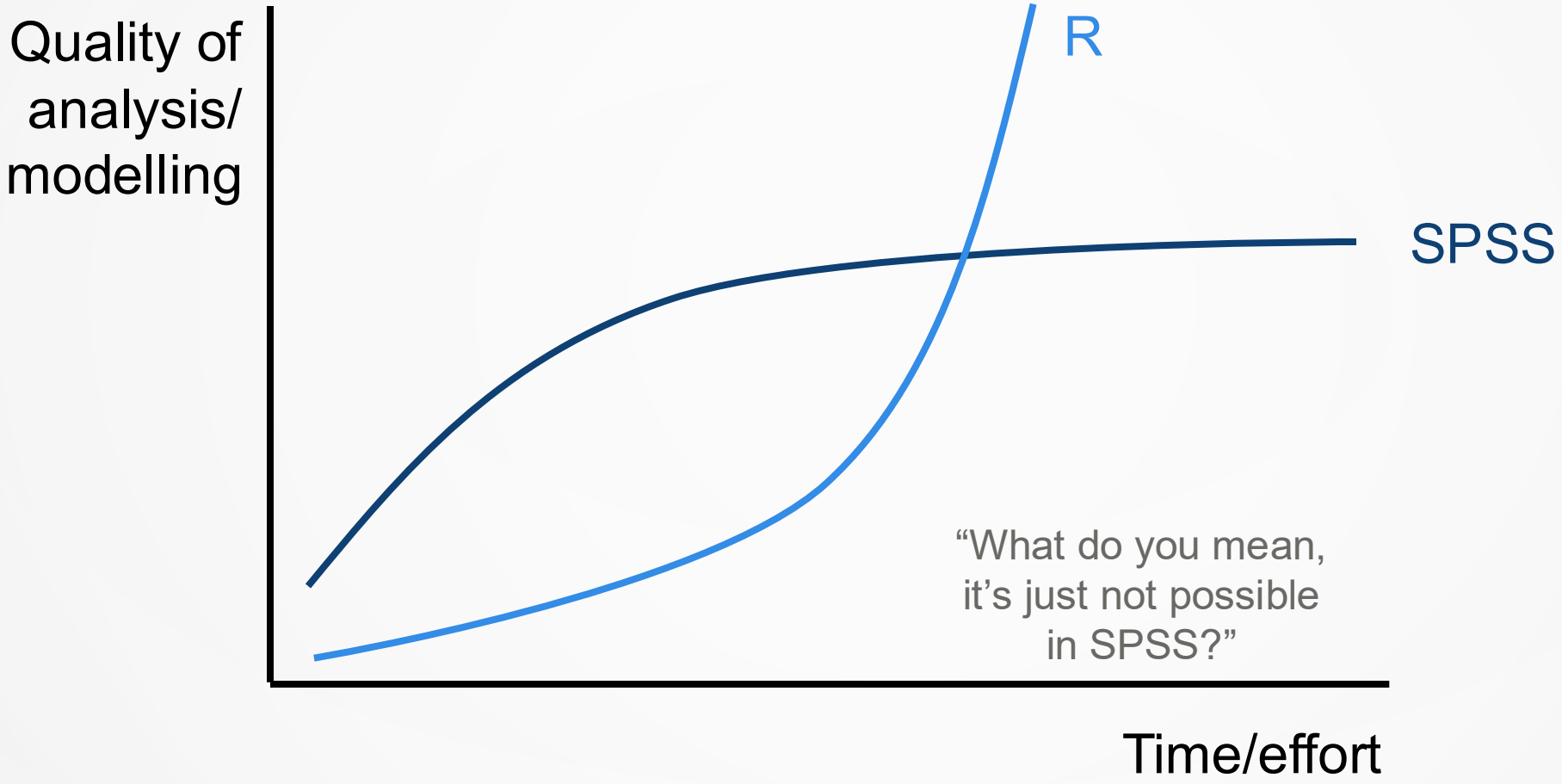
R AS A LEARNING CURVE



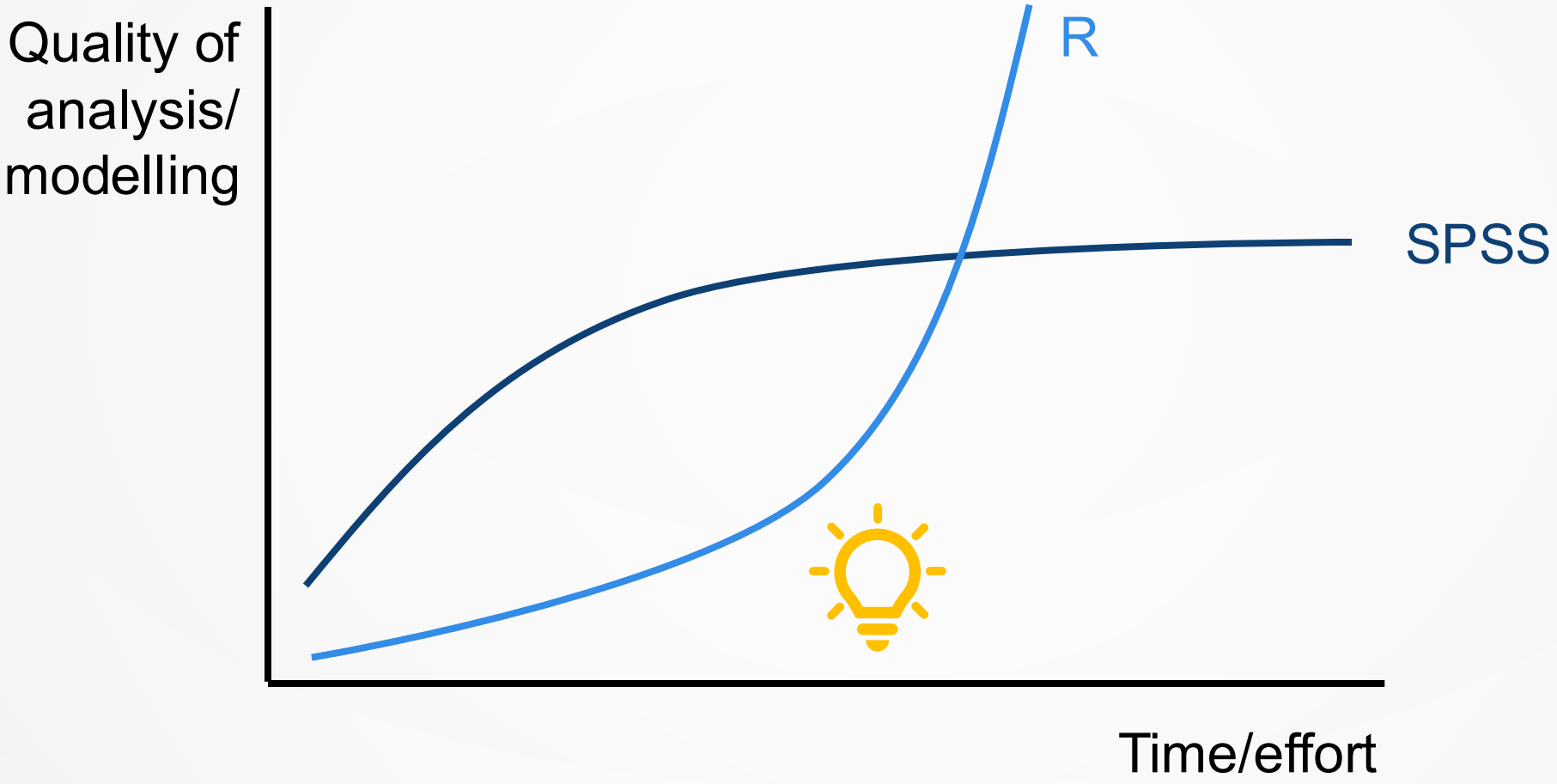
R AS A LEARNING CURVE



R AS A LEARNING CURVE



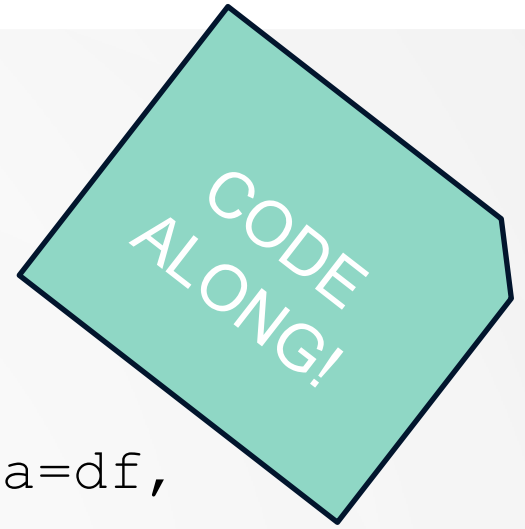
R AS A LEARNING CURVE



R AS A LANGUAGE

- Let's get stuck in
- Download the script from here:
  - https://github.com/asantoangles/linear_models_in_R

R AS A LANGUAGE




```
my_model <- lm(Weight_kg ~ Height_cm + Gender, data=df,
na.action=na.omit)
```

```
my_ids <- c(paste0("subj", c(1:5)), "subj_A", "subj_B")
```

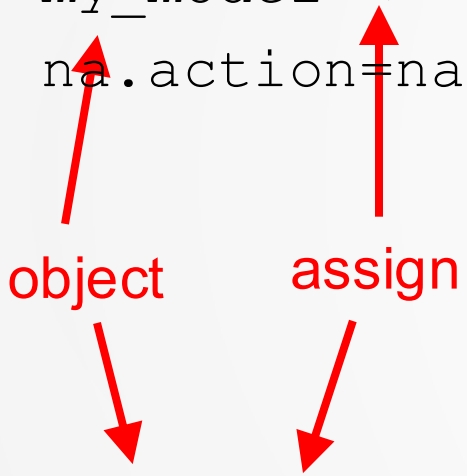
R AS A LANGUAGE

```
my_model <- lm(Weight_kg ~ Height_cm + Gender, data=df,
na.action=na.omit)
```


```
my_ids <- c(paste0("subj", c(1:5)), "subj_A", "subj_B")
```

R AS A LANGUAGE

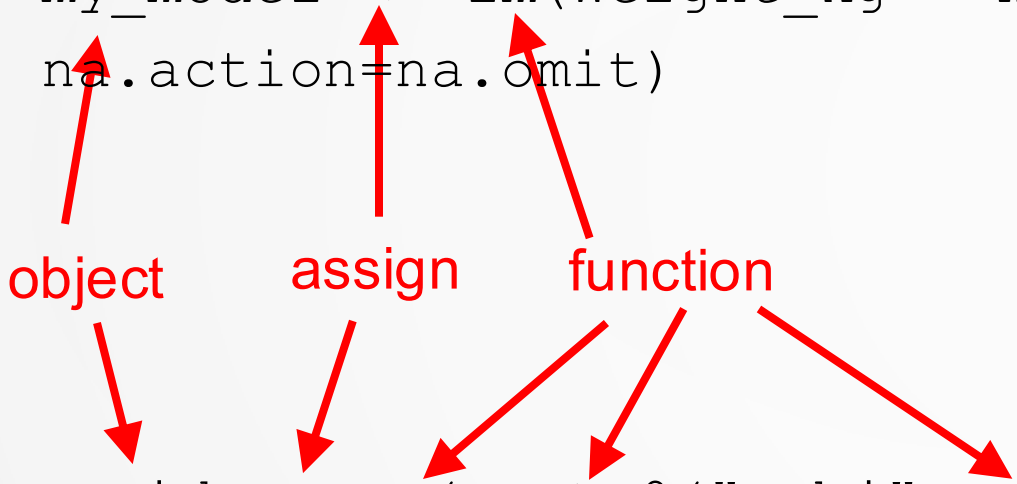
```
my_model <- lm(Weight_kg ~ Height_cm + Gender, data=df,
na.action=na.omit)
```




```
my_ids <- c(paste0("subj", c(1:5)), "subj_A", "subj_B")
```

R AS A LANGUAGE

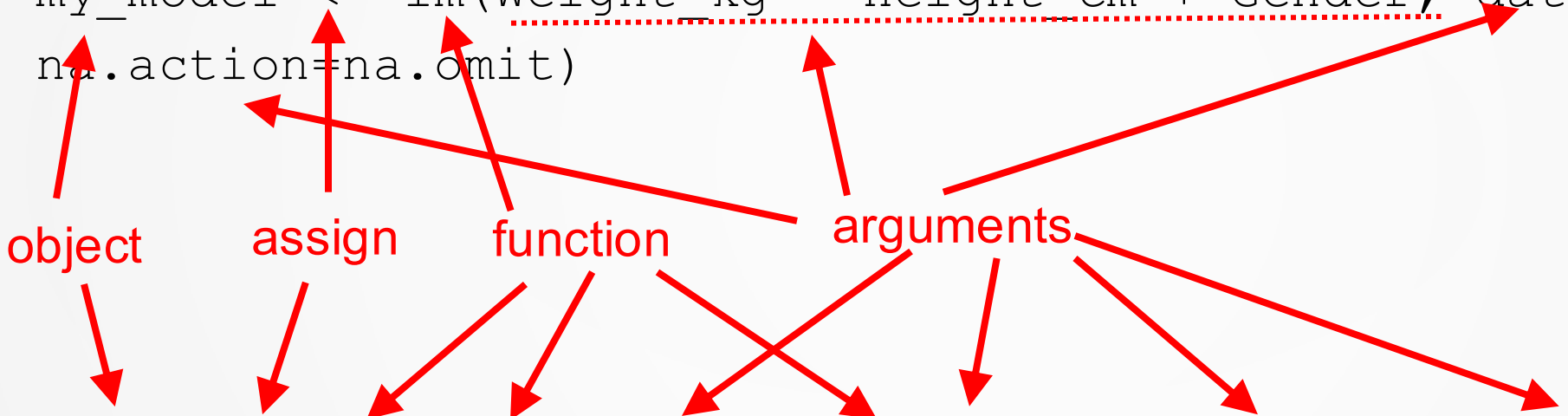
```
my_model <- lm(Weight_kg ~ Height_cm + Gender, data=df,
na.action=na.omit)
```




```
my_ids <- c(paste0("subj", c(1:5)), "subj_A", "subj_B")
```

R AS A LANGUAGE

```
my_model <- lm(Weight_kg ~ Height_cm + Gender, data=df,
na.action=na.omit)
```




```
my_ids <- c(paste0("subj", c(1:5)), "subj_A", "subj_B")
```

R AS A LANGUAGE


```
my_model        str(my_model)        summary(my_model)        coef(my_model)
```

- The same function can do different things to different types of objects
- Some functions only take certain object types as their arguments

```
my_ids          str(my_ids)          summary(my_ids)          coef(my_ids)
```

R AS A LANGUAGE

- You can just run a function call (like in SPSS)

```
lm(Weight_kg ~ Height_cm + Gender, data=df, na.action=na.omit)
```

- Or you can assign it to an object to use later on – recommended!

```
my_model <- lm(Weight_kg ~ Height_cm + Gender, data=df,
na.action=na.omit)

my_model
```

R AS A LANGUAGE

- Assigning to an object lets you do loads of things with your model

```
summary(my_model)

plot(predict(my_model), resid(my_model))

anova(my_model, another_model)
```

R AS A LANGUAGE

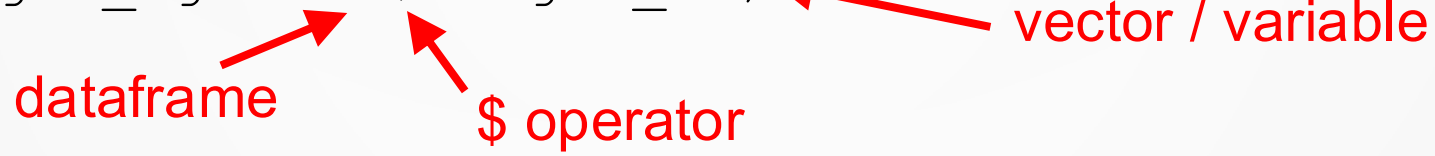
- Functions expect arguments in a certain order/way – specify if unsure

```
plot(df$Weight_kg, df$Height_cm)

plot(df$Weight_kg ~ df$Height_cm)

plot(x=df$Height_cm, y=df$Weight_kg)

plot(y=df$Height_cm, x=df$Weight_kg)
```

R AS A LANGUAGE

- Functions expect arguments in a certain order/way – specify if unsure

```
plot(df$Weight_kg, df$Height_cm)

plot(df$Weight_kg ~ df$Height_cm)

plot(x=df$Height_cm, y=df$Weight_kg)

plot(y=df$Height_cm, x=df$Weight_kg)
```



R AS A LANGUAGE

- Functions expect arguments in a certain order/way – specify if unsure

```
plot(df$Weight_kg, df$Height_cm)

plot(df$Weight_kg ~ df$Height_cm)

plot(x=df$Height_cm, y=df$Weight_kg)

plot(y=df$Height_cm, x=df$Weight_kg)
```

R AS A LANGUAGE

- Functions expect arguments in a certain order
- Easy to check:
  - ? opens the function documentation

```
?plot
```

- But function call depends on object

```
plot(my_model)
```

- Aside: What does the function itself look like?

```
lm
```

plot.default {graphics} R Documentation

The Default Scatterplot Function

Description

Draw a scatter plot with decorations such as axes and titles in the active graphics window.

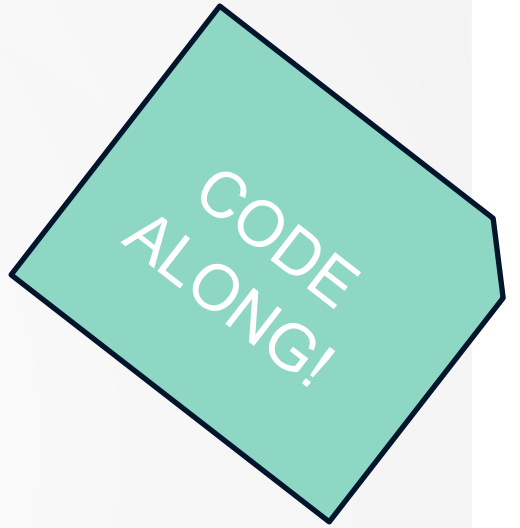
Usage

```
## Default S3 method:
plot(x, y = NULL, type = "p",  xlim = NULL, ylim = NULL,
     log = "", main = NULL, sub = NULL, xlab = NULL, ylab = NULL,
     ann = par("ann"), axes = TRUE, frame.plot = axes,
     panel.first = NULL, panel.last = NULL, asp = NA,
     xgap.axis = NA, ygap.axis = NA,
     ...)
```

Arguments

x, y	the x and y arguments provide the x and y coordinates for the plot. Any reasonable way of defining the coordinates is acceptable. See the function xy.coords for details. If supplied separately, they must be of the same length.
type	1-character string giving the type of plot desired. The following values are possible, for details, see plot: "p" for points, "l" for lines, "b" for both points and lines, "c" for empty points joined

DATA WRANGLING – SELECTING CASES



- Subsetting can be done in different ways!
  - With subset()

```
df_male <- subset(df, Gender==”Male”)

df_kids <- subset(df, Age<18)
```

  - Check that it worked as planned (or why it didn’t)

```
str(df_male)    head(df_male)    table(df_male$Gender)

str(df_kids)    head(df_kids)    table(df_kids$Age)
```

DATA WRANGLING – SELECTING CASES

- Subsetting can be done in different ways!
  - With indexing, using the operators [,]
    - [x,y] selects the xth row element(s) and the yth column element(s) of a dataframe
    - specify rows & columns explicitly
    - empty index = everything
    - 0 index = nothing
    - or use which() to find particular cases

```
df[1:3, c(1,3,4)]

df[0,]                df[,0]

df[1:3, c("ID", "Gender", "Height_cm")]

df[which(df$Age>75), ]
```

  - Also check out "piping syntax" from the tidyverse family of packages, e.g. dplyr

DATA WRANGLING – NEW VARIABLES

- Creating new variables is very smooth and flexible:
  - E.g. a logical (binary) variable for ”Is this patient obese”

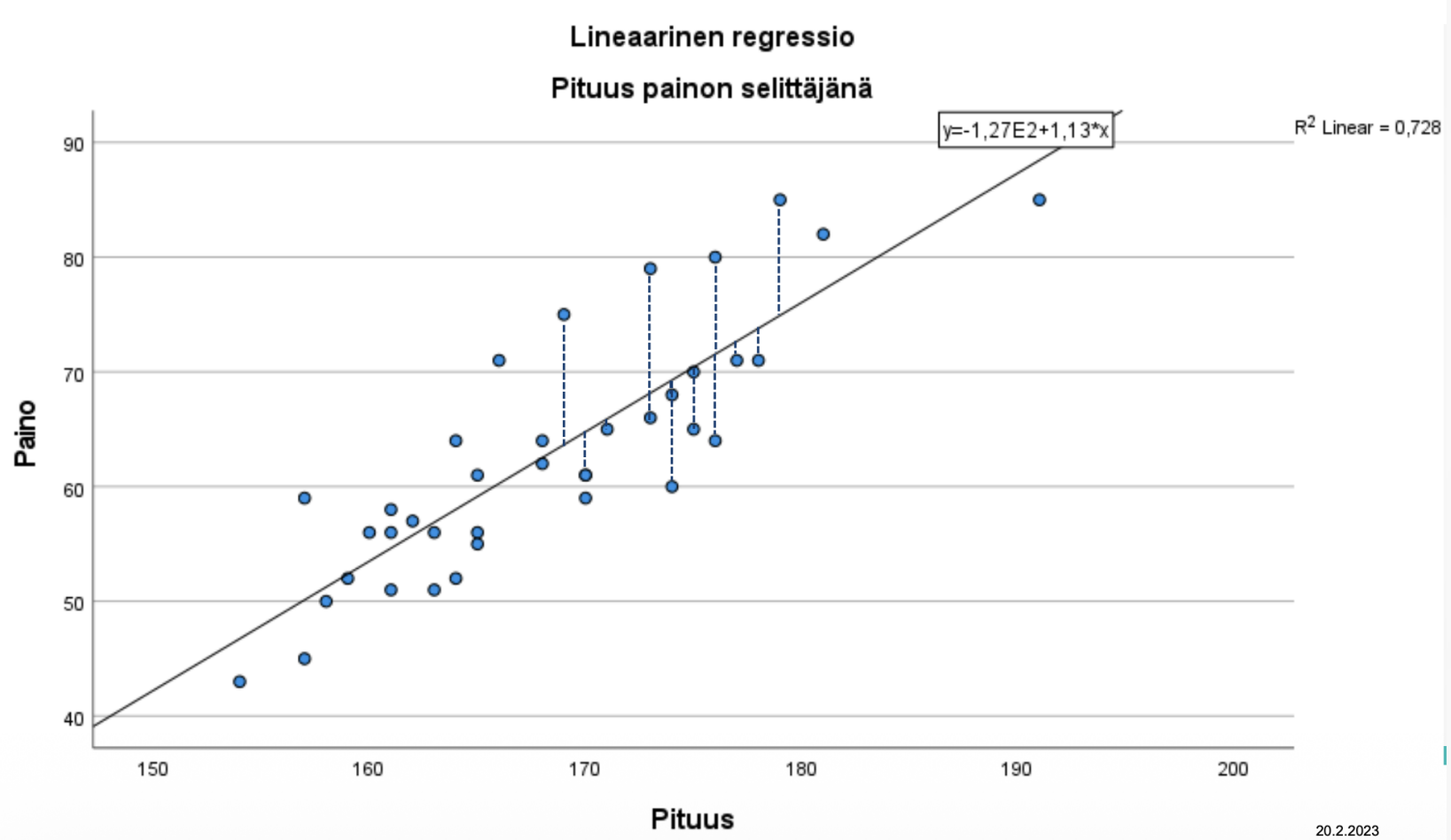
```
df$Obese<-ifelse(df$Weight_kg/((df$Height_cm/100)^2)>=25, 1, 0)
```

  - This single line of code says:
    – in the new variable called Obese in the dataframe df,
    – if the weight divided by height squared, i.e. BMI
    – is bigger than or equal to 25
    – assign to the new variable value 1
    – otherwise, assign it value 0

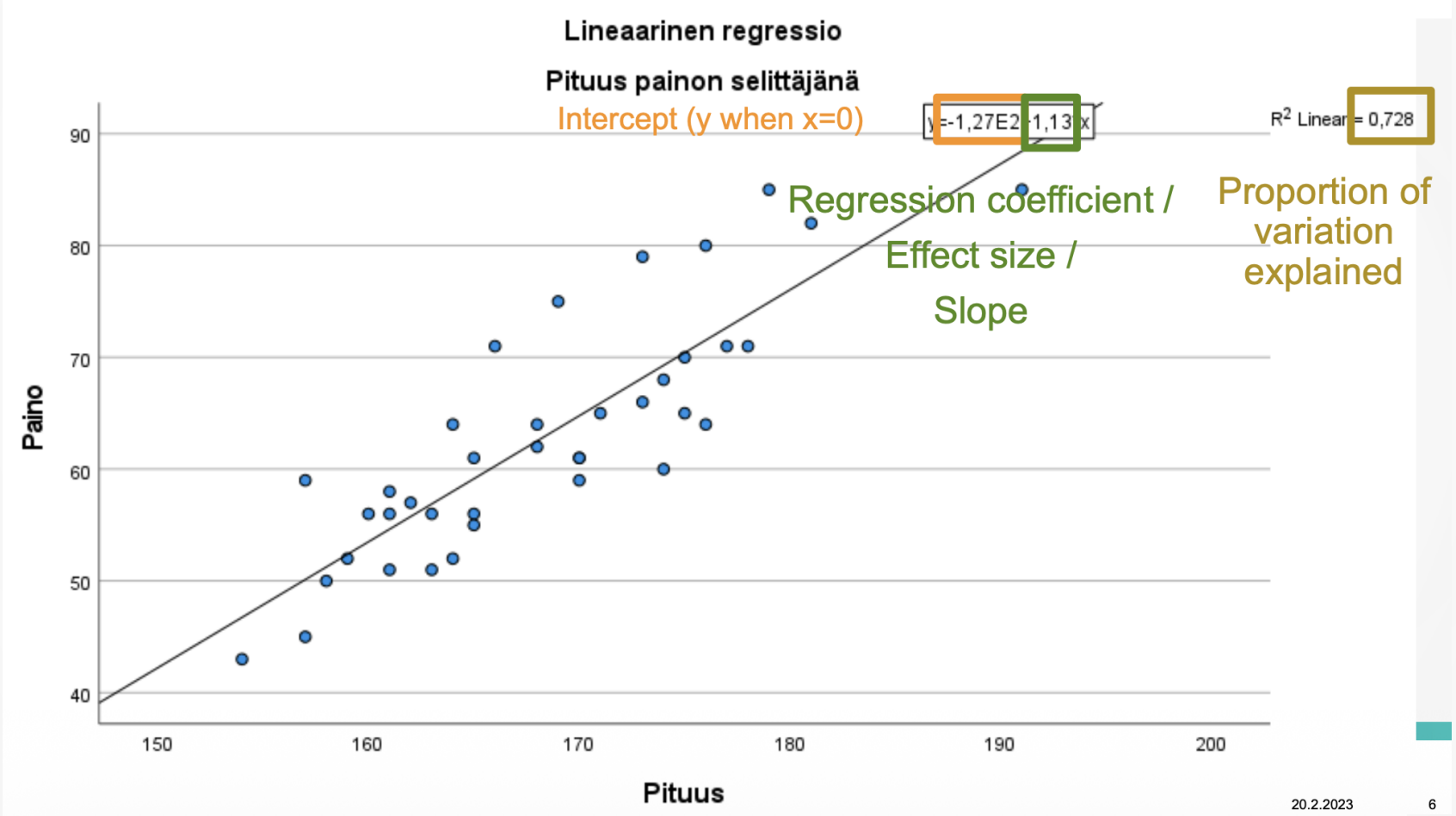
DATA WRANGLING - OULTLIERS

- ”I have these outliers. What should I do with them?”
  - 1. Check raw data
  - 2. If no obvious mistake, try with & without
  - 3. Consider biological meaning/role of outliers
  - 4. Be prepared to defend decisions! Much room for subjectivity
  - But also, avoid simple rules like ”exclude anything more than 3 s.d.s from mean” – those points could be valuable/meaningful

REGRESSION: LINEAR MODEL RECAP



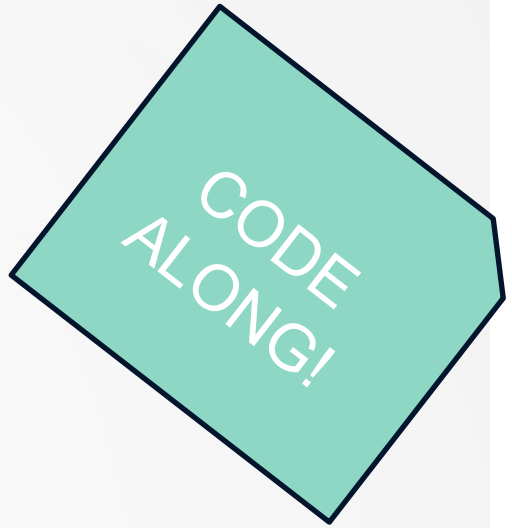
REGRESSION: LINEAR MODEL RECAP



REGRESSION: LM() IN PRACTICE

- Steps:
  - 1. Fit model
  - 2. Check residuals
  - 3. Interpret output
  - 4. Plot

REGRESSION: LM() IN PRACTICE



- Step 1: Fit model
  - Always assign to object!
  - Easy to code, not always easy to decide – what variables are important?

```
rm(my_model)

measures <-
read.table("https://www.mv.helsinki.fi/home/mjxpirin/medstat_course/mater
ial/Davis_height_weight.txt", as.is = TRUE, header = TRUE)

my_model <- lm(repwt ~ weight + sex, data=measures, na.action=na.exclude)
```

REGRESSION: LM() IN PRACTICE

- Step 2: Check assumptions (visually is most useful)
  - Assumption: residuals (errors) are normally distributed i.e. random
  - Assumption: homoskedasticity
  - Assumption: linearity
  - violation -> bias in estimates

```
resid(my_model)

hist(resid(my_model))

plot(my_model)

plot(measures$repwt, predict(my_model))
```

REGRESSION: LM() IN PRACTICE

- Step 3: Interpret results
  - Statistically significant/ meaningful – p-value
  - Clinically/biologically/ practically significant/meaningful – effect size
  - What is the model for?

REGRESSION: LM() IN PRACTICE

```
summary(my_model)
```

- Step 3: Interpret results
  - Statistically significant/ meaningful – p-value
  - Clinically/biologically/ practically significant/meaningful – effect size
  - What is the model for?

What you asked R to run (sanity check)

```
Call:
lm(formula = repwt ~ weight + sex, data = measures, na.action = na.exclude)

Residuals:
    Min      1Q  Median      3Q     Max
-6.4998 -1.3238  0.1816  1.0274  8.8360

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.02500    1.02371   1.001  0.31805
weight       0.97370    0.01745  55.807  < 2e-16 ***
sexM         1.55833    0.46901   3.323  0.00108 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.256 on 179 degrees of freedom
  (17 observations deleted due to missingness)
Multiple R-squared:  0.9735, Adjusted R-squared:  0.9733
F-statistic:  3294 on 2 and 179 DF,  p-value: < 2.2e-16
```

REGRESSION: LM() IN PRACTICE

```
summary(my_model)
```

- Step 3: Interpret results
  - Statistically significant/ meaningful – p-value
  - Clinically/biologically/ practically significant/meaningful – effect size
  - What is the model for?

What you asked R to run (sanity check)

```
Call:
lm(formula = repwt ~ weight + sex, data = measures, na.action = na.exclude)

Residuals:
    Min      1Q  Median      3Q     Max
-6.4998 -1.3238  0.1816  1.0274  8.8360

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.02500    1.02371   1.001  0.31805
weight       0.97370    0.01745  55.807  < 2e-16 ***
sexM         1.55833    0.46901   3.323  0.00108 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.256 on 179 degrees of freedom
  (17 observations deleted due to missingness)
Multiple R-squared:  0.9735, Adjusted R-squared:  0.9733
F-statistic:  3294 on 2 and 179 DF,  p-value: < 2.2e-16
```

Regression coefficient

(effect size / beta / B /slope / group difference)

REGRESSION: LM() IN PRACTICE

```
summary(my_model)
```

- Step 3: Interpret results
  - Statistically significant/ meaningful – p-value
  - Clinically/biologically/ practically significant/meaningful – effect size
  - What is the model for?

What you asked R to run (sanity check)

```
Call:
lm(formula = repwt ~ weight + sex, data = measures, na.action = na.exclude)

Residuals:
    Min      1Q  Median      3Q     Max
-6.4998 -1.3238  0.1816  1.0274  8.8360

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.02500    1.02371   1.001  0.31805
weight       0.97370    0.01745  55.807  < 2e-16 ***
sexM         1.55833    0.46901   3.323  0.00108 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.256 on 179 degrees of freedom
  (17 observations deleted due to missingness)
Multiple R-squared:  0.9735, Adjusted R-squared:  0.9733
F-statistic:  3294 on 2 and 179 DF,  p-value: < 2.2e-16
```

Regression coefficient

= Real-world meaning!

REGRESSION: LM() IN PRACTICE

```
summary(my_model)
```

- Step 3: Interpret results
  - Statistically significant/ meaningful – p-value
  - Clinically/biologically/ practically significant/meaningful – effect size
  - What is the model for?

What you asked R to run (sanity check)

Estimation error (-> Cis)

p-value for each parameter (is it different from 0?)

```
Call:
lm(formula = repwt ~ weight + sex, data = measures, na.action = na.exclude)

Residuals:
    Min      1Q  Median      3Q     Max
-6.4998 -1.3238  0.1816  1.0274  8.8360

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.02500    1.02371   1.001  0.31805
weight       0.97370    0.01745  55.807  < 2e-16 ***
sexM         1.55833    0.46901   3.323  0.00108 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.256 on 179 degrees of freedom
  (17 observations deleted due to missingness)
Multiple R-squared:  0.9735, Adjusted R-squared:  0.9733
F-statistic:  3294 on 2 and 179 DF,  p-value: < 2.2e-16
```

Regression coefficient

(effect size / beta / B /slope / group difference)

REGRESSION: LM() IN PRACTICE

```
summary(my_model)
```

- Step 3: Interpret results
  - Statistically significant/ meaningful – p-value
  - Clinically/biologically/ practically significant/meaningful – effect size
  - What is the model for?

What you asked R to run (sanity check)

```
Call:
lm(formula = repwt ~ weight + sex, data = measures, na.action = na.exclude)

Residuals:
    Min      1Q  Median      3Q     Max
-6.4998 -1.3238  0.1816  1.0274  8.8360

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.02500    1.02371   1.001  0.31805
weight       0.97370    0.01745  55.807  < 2e-16 ***
sexM         1.55833    0.46901   3.323  0.00108 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.256 on 179 degrees of freedom
  (17 observations deleted due to missingness)
Multiple R-squared:  0.9735, Adjusted R-squared:  0.9733
F-statistic:  3294 on 2 and 179 DF,  p-value: < 2.2e-16
```

Estimation error (-> Cis)

p-value for each parameter (is it different from 0?)

p-value for whole model (is it better than no predictors?)

Regression coefficient

(effect size / beta / B /slope / group difference)

REGRESSION: LM() IN PRACTICE

```
summary(my_model)
```

- Step 3: Interpret results
  - Statistically significant/ meaningful – p-value
  - Clinically/biologically/ practically significant/meaningful – effect size
  - What is the model for?

What you asked R to run (sanity check)

Estimation error (-> Cis)

p-value for each parameter (is it different from 0?)

p-value for whole model (is it better than no predictors?)

Regression coefficient

(effect size / beta / B /slope / group difference)

```
Call:
lm(formula = repwt ~ weight + sex, data = measures, na.action = na.exclude)

Residuals:
    Min      1Q  Median      3Q     Max
-6.4998 -1.3238  0.1816  1.0274  8.8360

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.02500    1.02371   1.001  0.31805
weight       0.97370    0.01745  55.807  < 2e-16 ***
sexM         1.55833    0.46901   3.323  0.00108 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.256 on 179 degrees of freedom
  (17 observations deleted due to missingness)
Multiple R-squared:  0.9735, Adjusted R-squared:  0.9733
F-statistic:  3294 on 2 and 179 DF,  p-value: < 2.2e-16
```

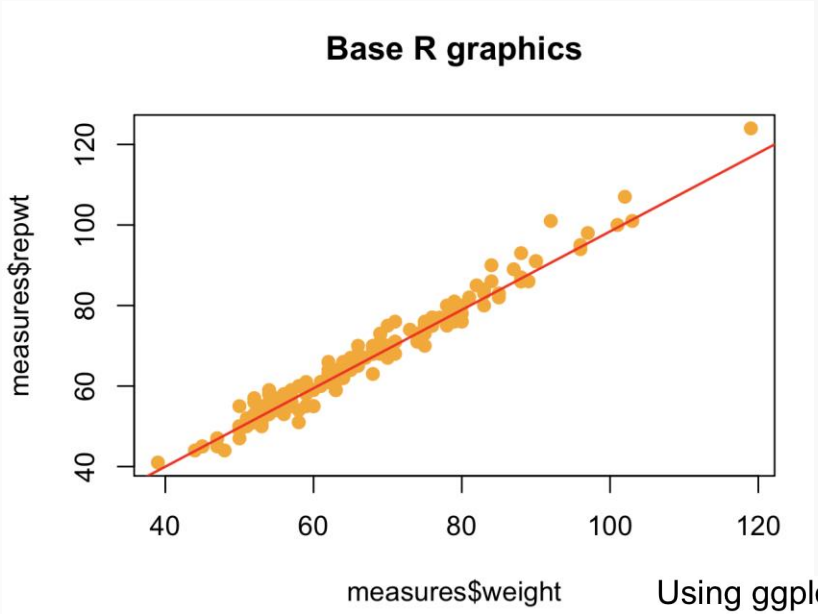
REGRESSION: LM() IN PRACTICE

- Step 4: Plot
  - Sanity check – model, outliers etc.
  - Intuitive interpretation
  - Effective messaging in publications

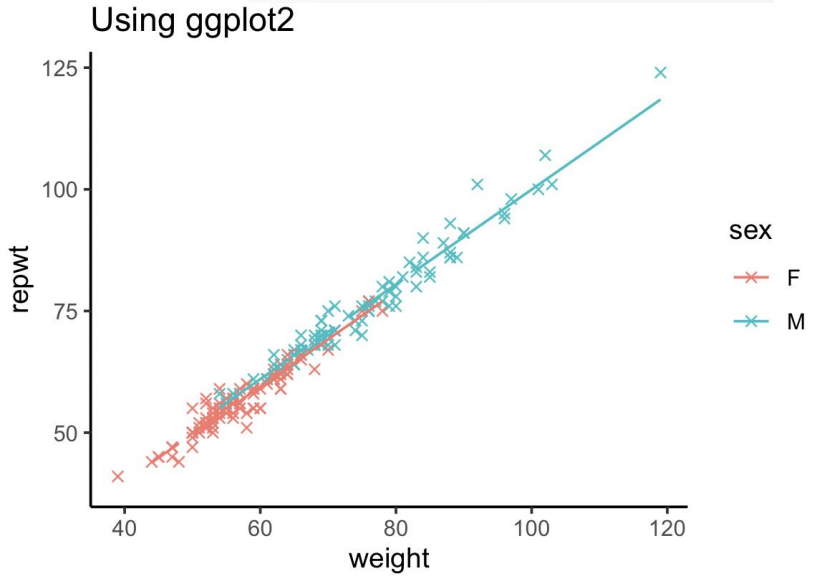
- Base R plot() good for super-quick basic plots, but little flexibility
- ggplot2 – harder to understand but SO powerful

REGRESSION: LM() IN PRACTICE

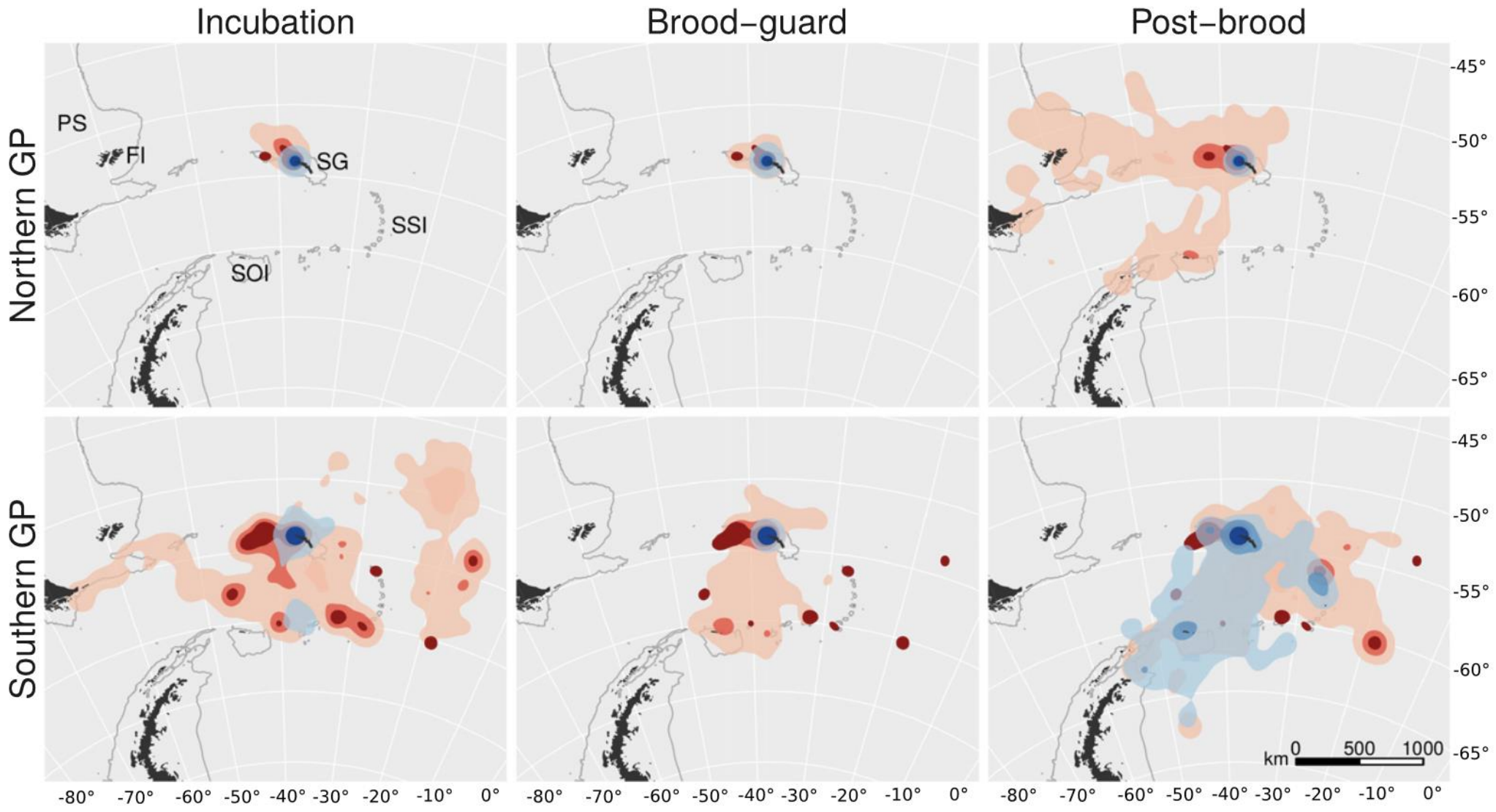
- Step 4: Plot
  - Sanity check – model, outliers etc.
  - Intuitive interpretation
  - Effective messaging in publications



- Base R plot() good for super-quick basic plots, but little flexibility
- ggplot2 – harder to understand but SO powerful



AN ASIDE: GGPLOT IS COOL!



Map of seabird foraging areas, made with ggplot. From Granroth-Wilding & Phillips 2019, *Ibis*

AN ASIDE: RELATED TESTS

- T-tests & ANOVAs are part of the same family as linear regression

AN ASIDE: RELATED TESTS

- T-tests & ANOVAs are part of the same family as linear regression

```
    Welch Two Sample t-test

data:  measures$repwt[which(measures$sex == "F")]
measures$repwt[which(measures$sex == "M")]
t = -13.064, df = 120.26, p-value < 2.2e-16
alternative hypothesis: true difference in means i
95 percent confidence interval:
 -22.81330 -16.80866
sample estimates:
mean of x mean of y
 56.75000  76.56098
```

```
Call:
lm(formula = repwt ~ sex, data = measures, na.action = na.exclude)

Residuals:
    Min      1Q  Median      3Q     Max
-20.561  -6.358  -1.561   5.047  47.439

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  56.7500     0.9651   58.80   <2e-16 ***
sexM         19.8110     1.4379   13.78   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9.651 on 180 degrees of freedom
  (17 observations deleted due to missingness)
Multiple R-squared:  0.5133, Adjusted R-squared:  0.5106
F-statistic: 189.8 on 1 and 180 DF,  p-value: < 2.2e-16
```

AN ASIDE: RELATED TESTS

- T-tests & ANOVAs are part of the same family as linear regression
- By the way:
  - aov() runs an ANOVA
  - anova() gives you an F-test of one or more model object(s)

aov {stats} R Documentation

Fit an Analysis of Variance Model

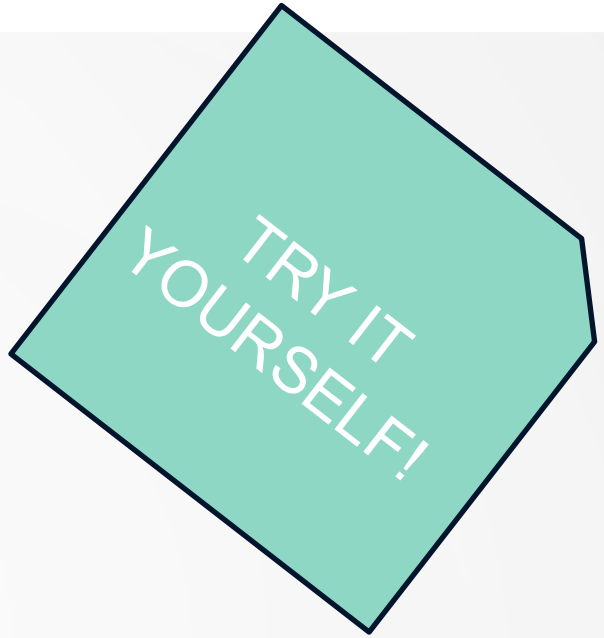
Description

Fit an analysis of variance model by a call to `lm` (for each stratum if an `Error(.)` is used).

Usage

```
aov(formula, data = NULL, projections = FALSE, qr = TRUE,
    contrasts = NULL, ...)
```

LM(): A MODELLING JOURNEY



- Using `measures` dataset, explore the role of height and sex in determining weight
- Which predictors are important?
- How big are their effects?
- What if males and females have different slopes??
- Which is the best model?
  - Hint: useful functions include `AIC(model_name_1, model_name_2)` and (from the `lmtest` package) `lrtest(model_name_1, model_name_2)`

LM(): MODEL COMPARISON

- Instead of focusing on individual terms, we can compare entire models
  - The model with the most informative combination of terms is preferred
  - BUT think about what your model is for! Maybe single coefficients ARE interesting
- More holistic approach to inference
- Options:
  - F-test (based on sum of squares): `anova()`
  - Likelihood ratio test: `lrtest()`
  - AIC or BIC (both also likelihood-based) for a comparative overview of several models: `AIC()` and `BIC()` respecitvely

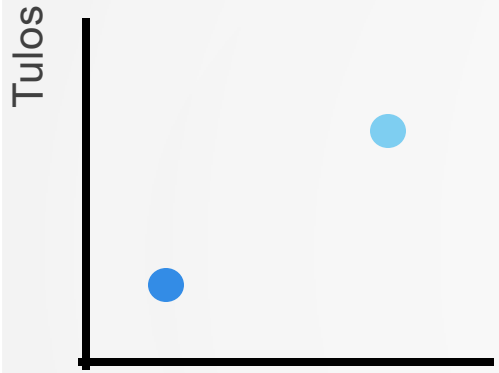
COMMON MISTAKES: TYPOS & ERROR MESSAGES

- Error messages often cryptic, hard to understand
- Often caused by a simple mistake
  - Bracket left open, or too many brackets
  - Commas in wrong places or missing
  - Using the wrong object e.g. dataframe instead of vector
- Google the error message!
  - See a few separate examples -> build an idea of how it might apply in your specific case
- Examples
  - “$ operator is invalid for atomic vectors”
  - “incorrect number of dimensions”

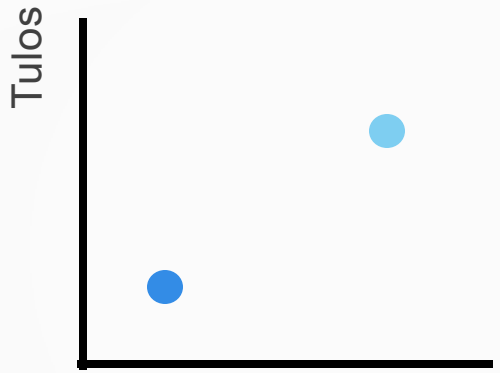
COMMON MISTAKES: P-VALUES

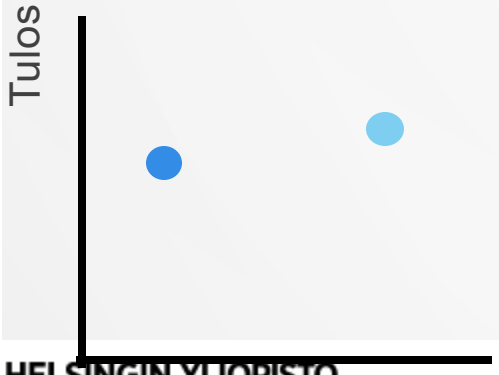
- What does the p-value really mean?
  - C.f. effect size

COMMON MISTAKES: P-VALUES

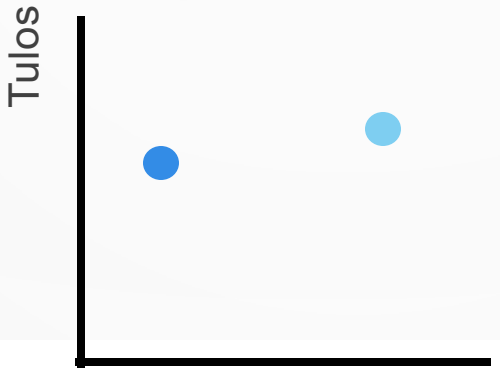


Big effect

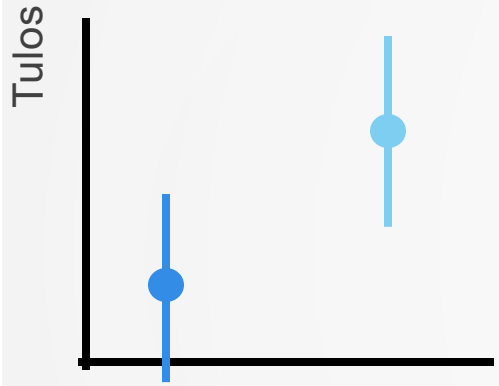




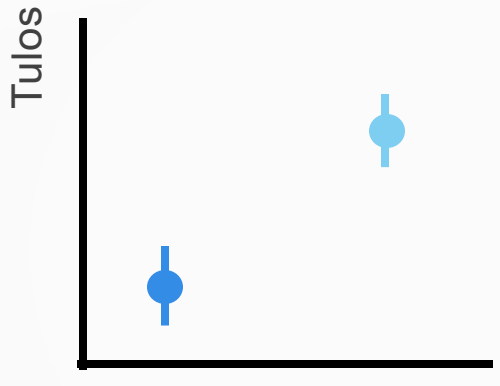
Small effect

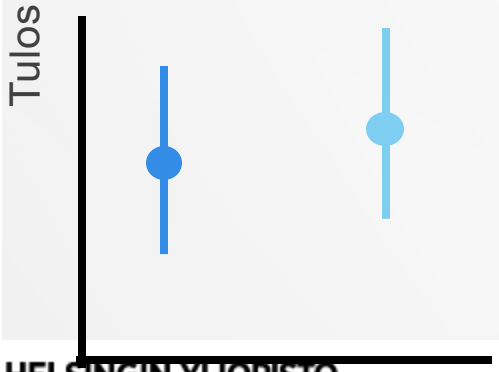


COMMON MISTAKES: P-VALUES

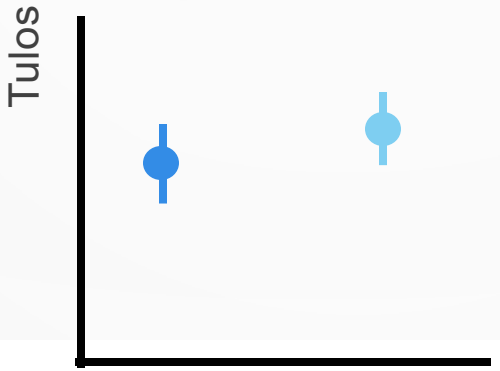


Big effect

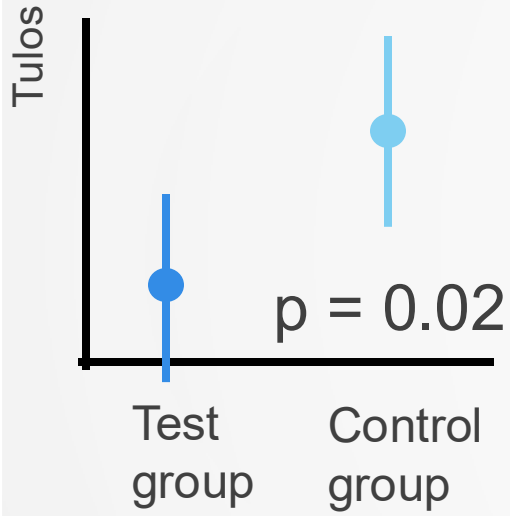




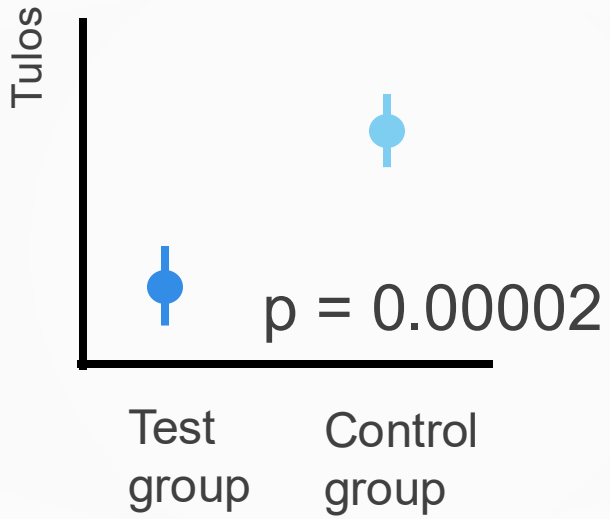
Small effect

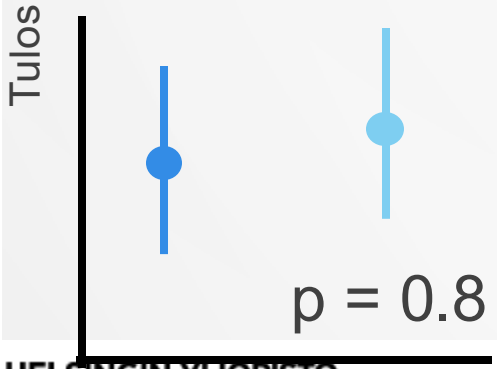


COMMON MISTAKES: P-VALUES

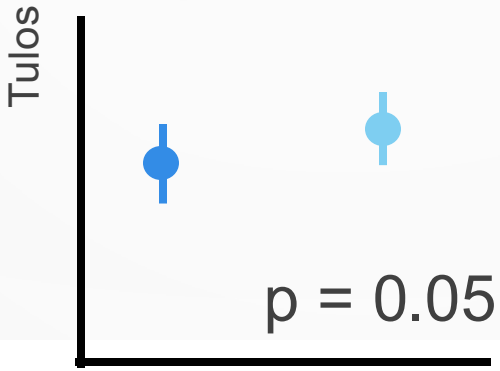


Big effect





Small effect



- Statistical significance ≠ practical/clinical/ biological significance
- p-value vs. effect size

COMMON MISTAKES: P-VALUES

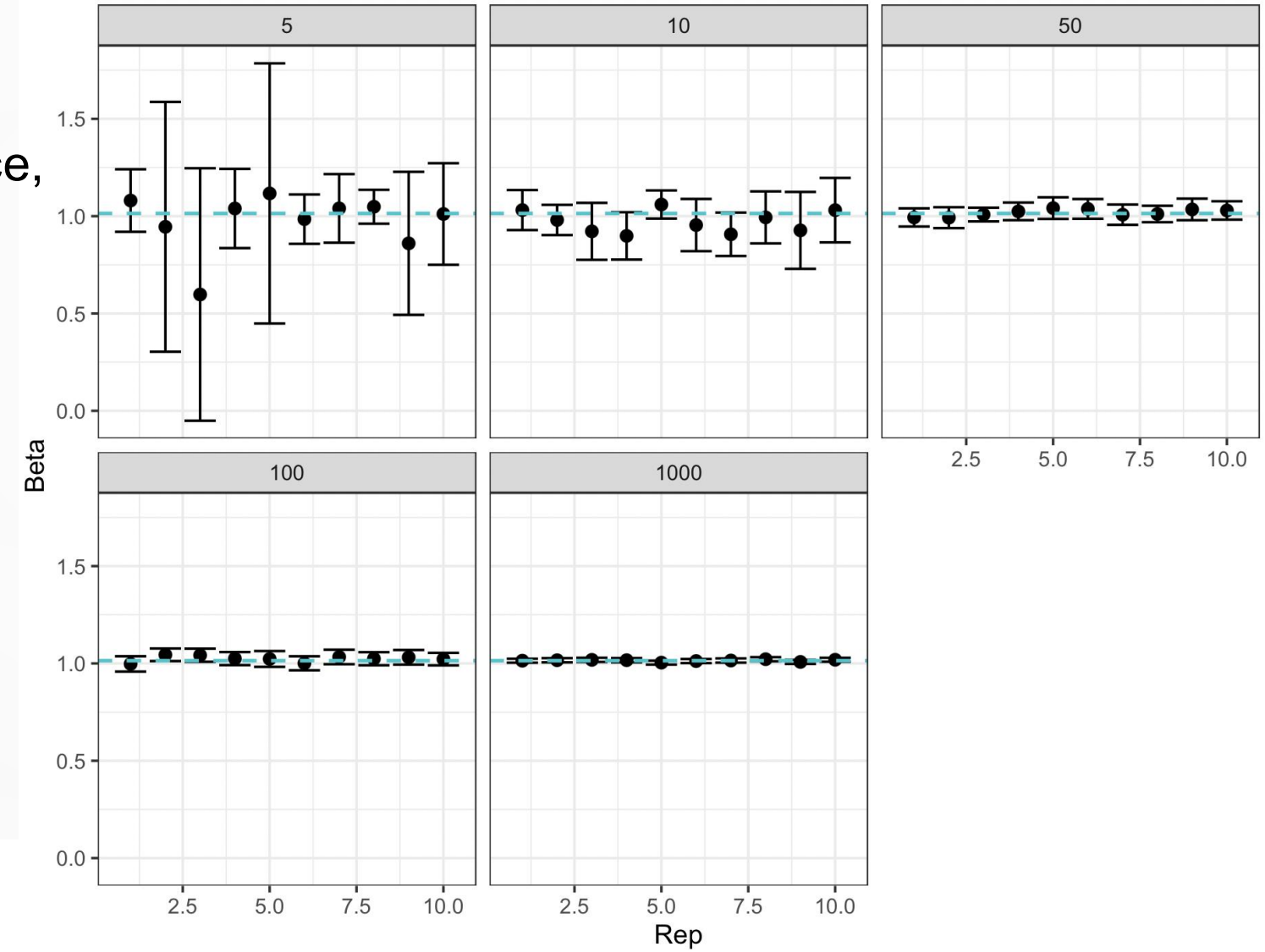
- What does the p-value really mean?
  - C.f. effect size
  - Stand up to journals if they ask for inappropriate p-values!

COMMON MISTAKES: SMALL SAMPLES

- Small sample -> less confidence, more likely wrong
  - p-values less meaningful for inference

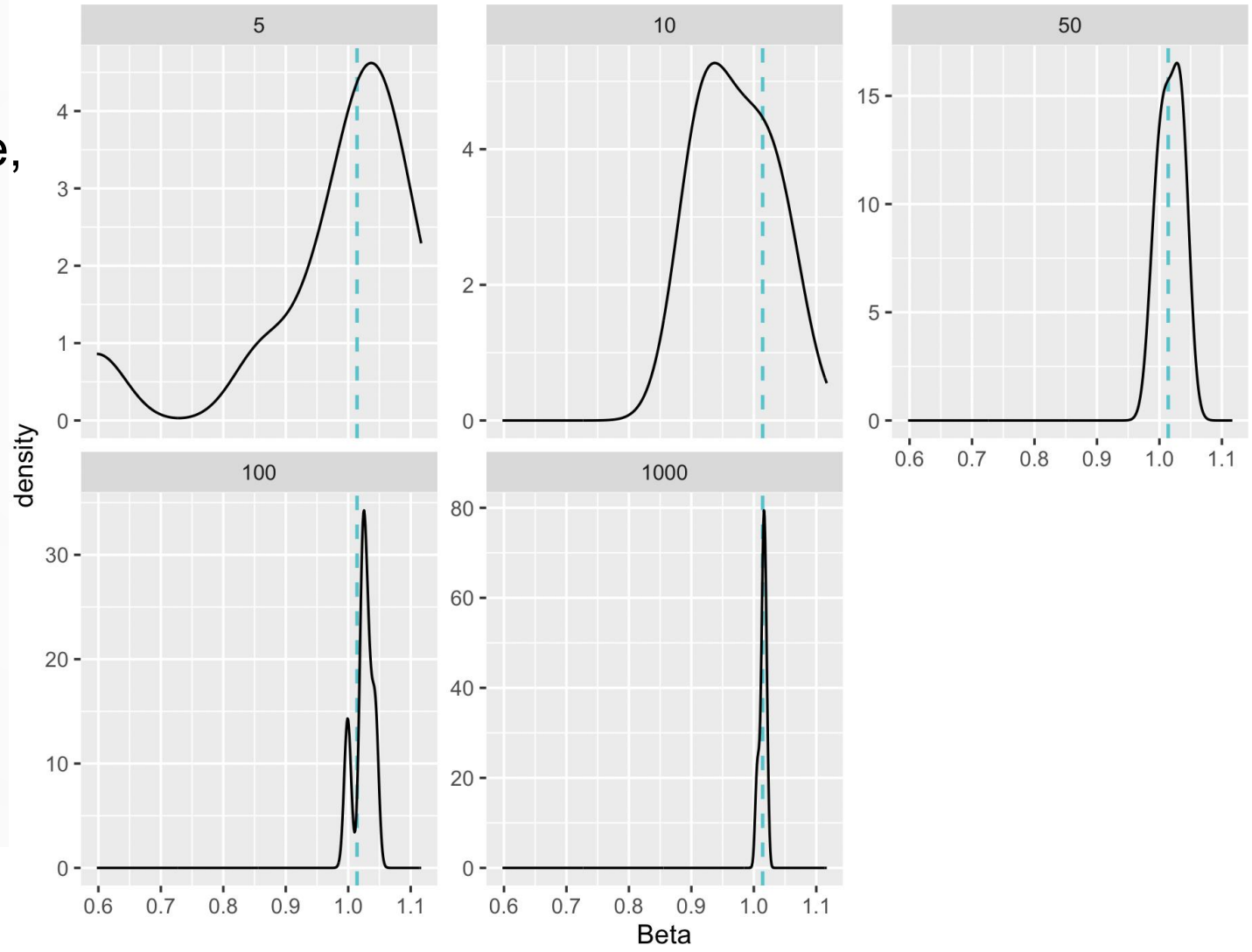
COMMON MISTAKES: SMALL SAMPLES

- Small sample -> less confidence, more likely wrong
  - p-values less meaningful for inference



COMMON MISTAKES: SMALL SAMPLES

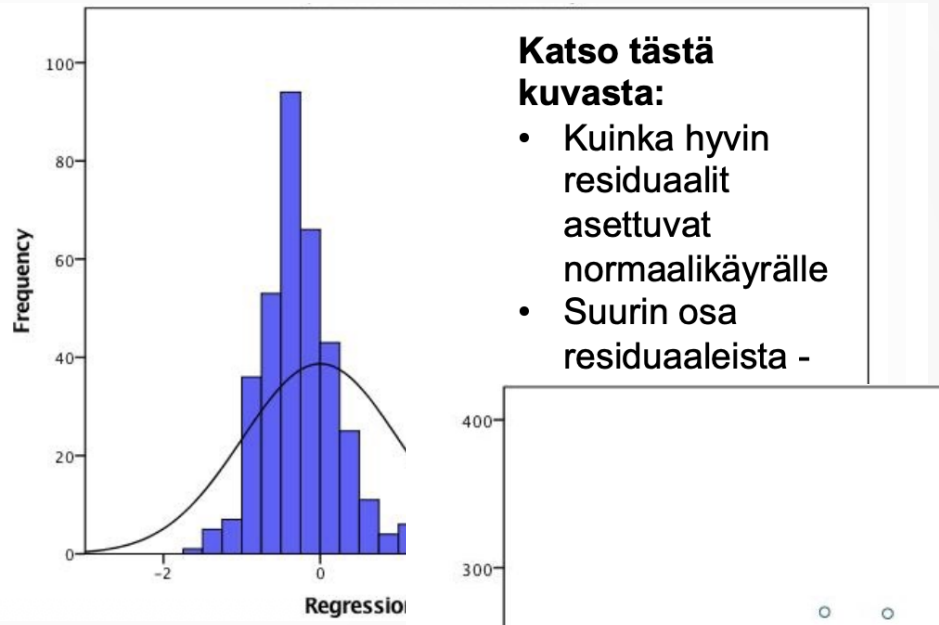
- Small sample -> less confidence, more likely wrong
  - p-values less meaningful for inference
- (See the code for plotting each sampled fit)



COMMON MISTAKES: VIOLATED ASSUMPTIONS

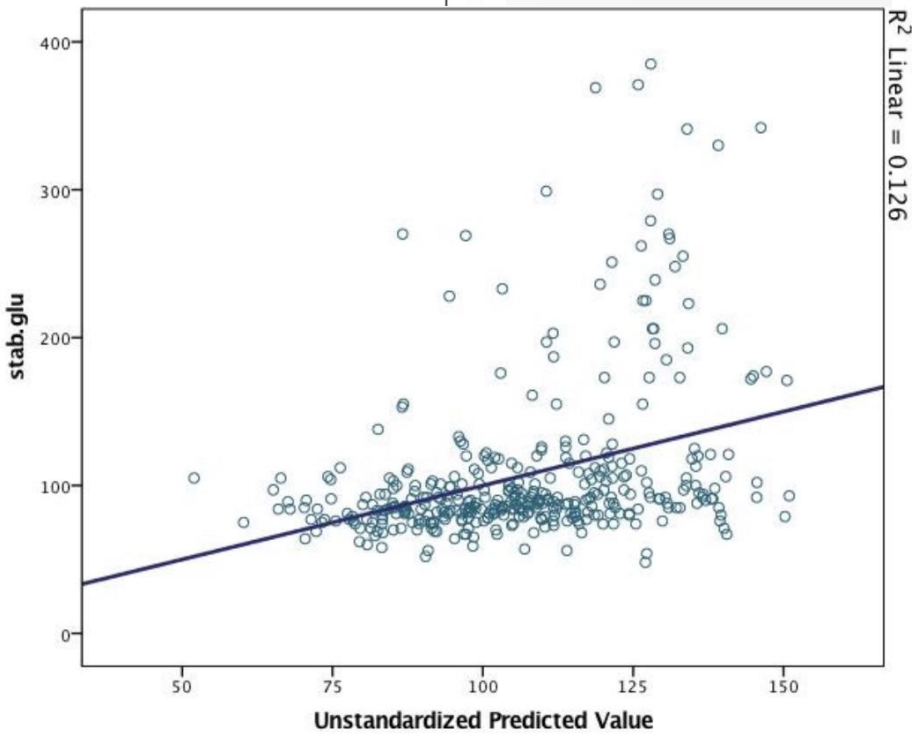
- Residuals not normally distributed
  - Wrong model – somehow…
  - Predictors? Distribution?
  - Might not be solvable….

- Residuals not homoskedastic
  - which looks like this (for example)
  - Missing predictor? Variance function? Distributions? Process?



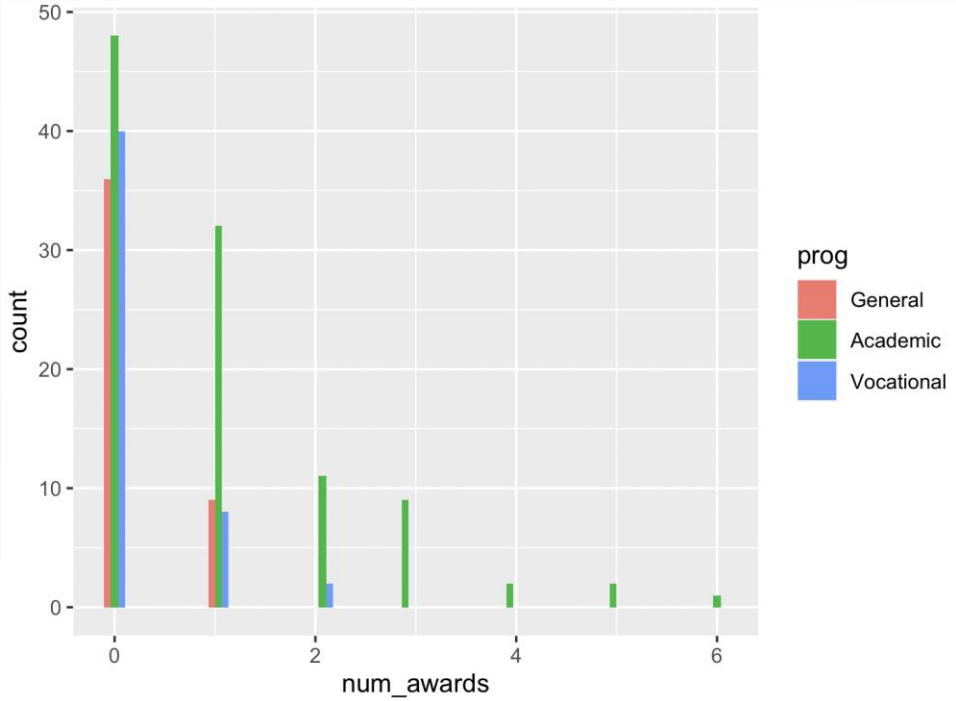
Katso tästä kuvasta:

- Kuinka hyvin residuaalit asettuvat normaalikäyrälle
- Suurin osa residuaaleista -



EXPANDING THE FRAMEWORK: GLM

- Not all data works in a classic linear model
- Depends on underlying data-generating process as well as statistical fit
  - E.g. count data – might feel continuous (hence normal) but usually behaves quite differently



EXPANDING THE FRAMEWORK: GLM

- What if assume normal continuous data?

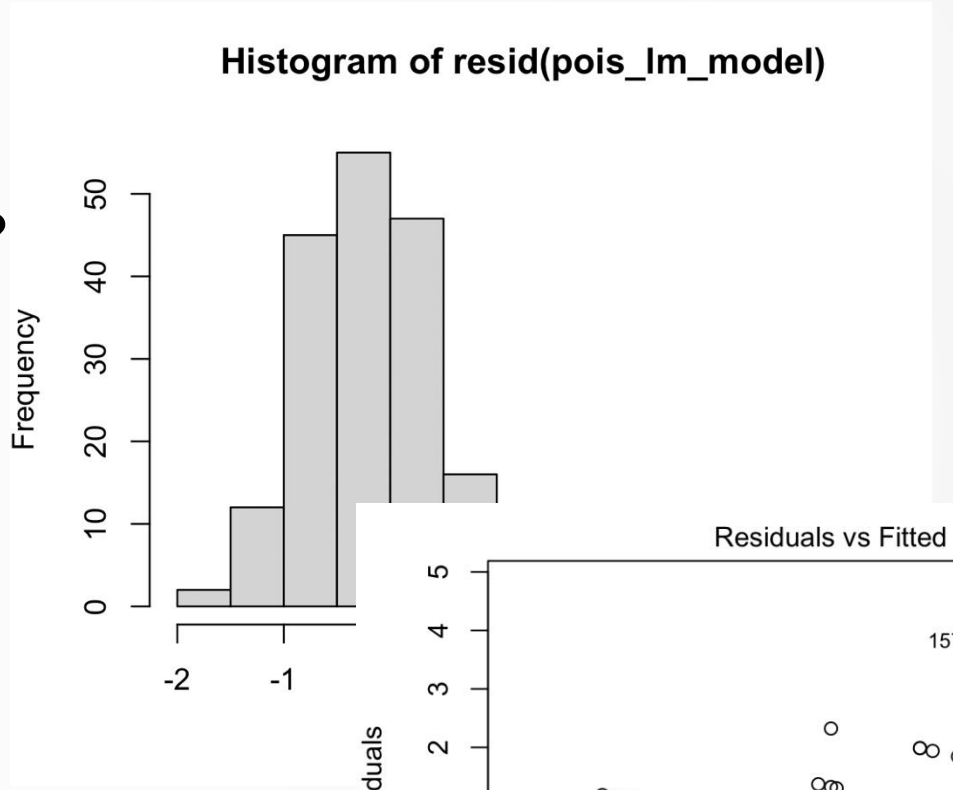
```
Call:
lm(formula = num_awards ~ prog + math, data = pois_data)

Residuals:
    Min      1Q  Median      3Q     Max
-1.7311 -0.5618 -0.1537  0.2851  4.4126

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    -2.195504   0.411417  -5.336 2.60e-07 ***
progAcademic    0.478613   0.168956   2.833   0.0051 **
progVocational  0.212506   0.187433   1.134   0.2583
math            0.047889   0.007773   6.161 4.03e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9019 on 196 degrees of freedom
Multiple R-squared:  0.2773, Adjusted R-squared:  0.2662
F-statistic: 25.07 on 3 and 196 DF,  p-value: 9.016e-14
```

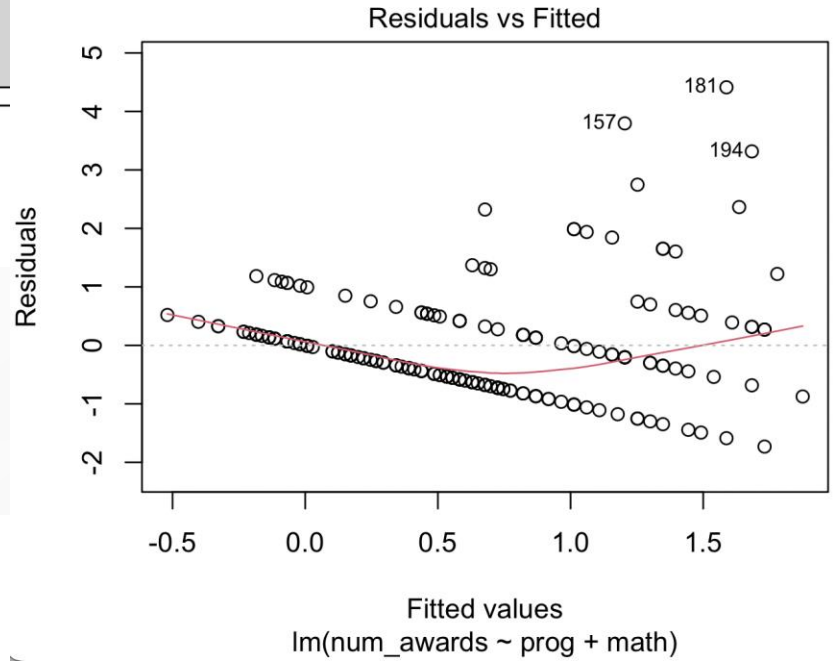
- Hooray, significant results, time to publish!
- Actually, best check the model fit first.
- (Not to mention that significant results are NOT what makes a good paper)

EXPANDING THE FRAMEWORK: GLM

- What if assume normal continuous data?

```
Call:
lm(formula = num_awards ~ prog + math, data = pois_data)

Residuals:
    Min      1Q  Median      3Q     Max
-1.7311 -0.5618 -0.1537  0.2851  4.4126

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    -2.195504   0.411417  -5.336 2.60e-07 ***
progAcademic    0.478613   0.168956   2.833   0.0051 **
progVocational  0.212506   0.187433   1.134   0.2583
math            0.047889   0.007773   6.161 4.03e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9019 on 196 degrees of freedom
Multiple R-squared:  0.2773, Adjusted R-squared:  0.2662
F-statistic: 25.07 on 3 and 196 DF,  p-value: 9.016e-14
```





EXPANDING THE FRAMEWORK: GLM

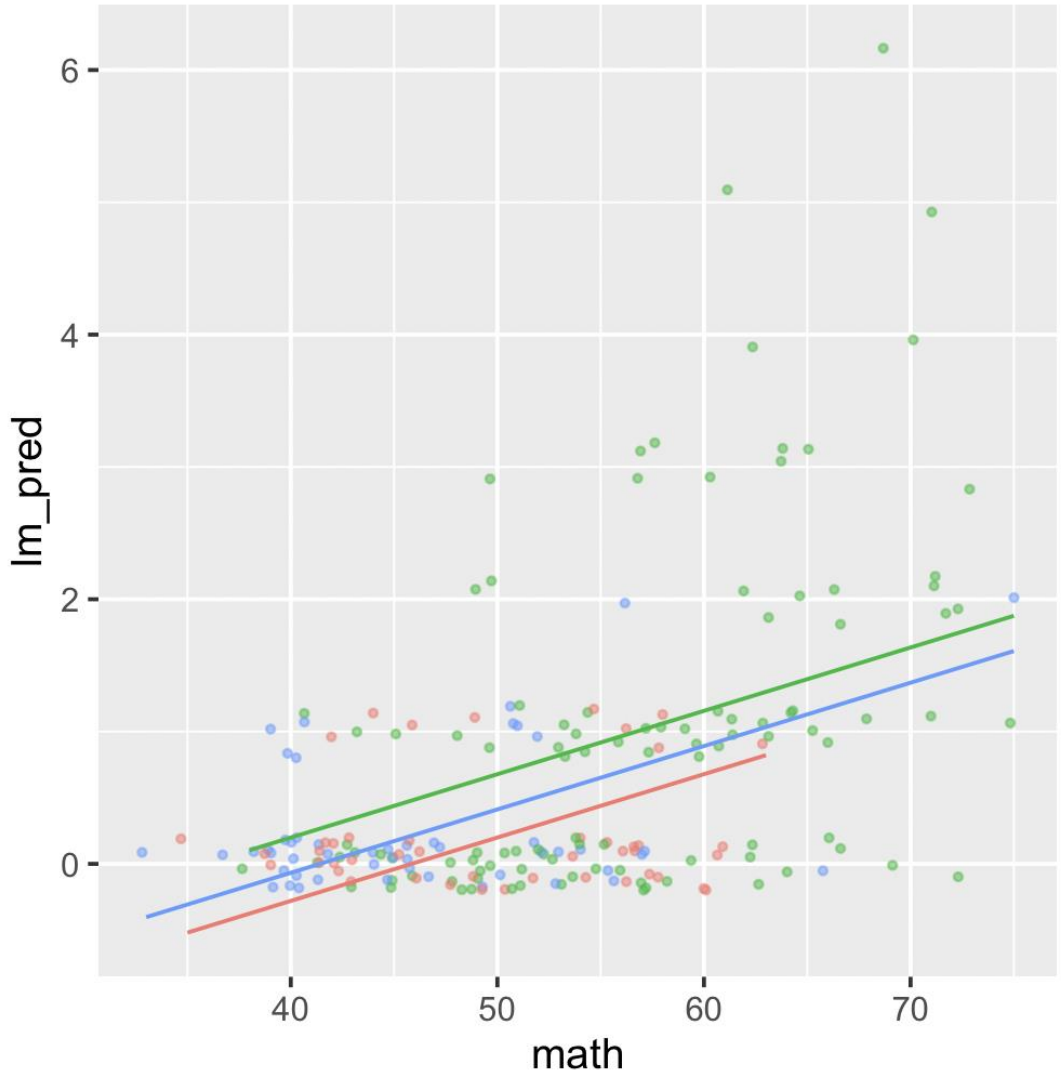
- Model is not capturing the data well!
- Generalized Linear Model
  - link function -> linearise relationships with non-linear response
- Better model for count data: Poisson GLM

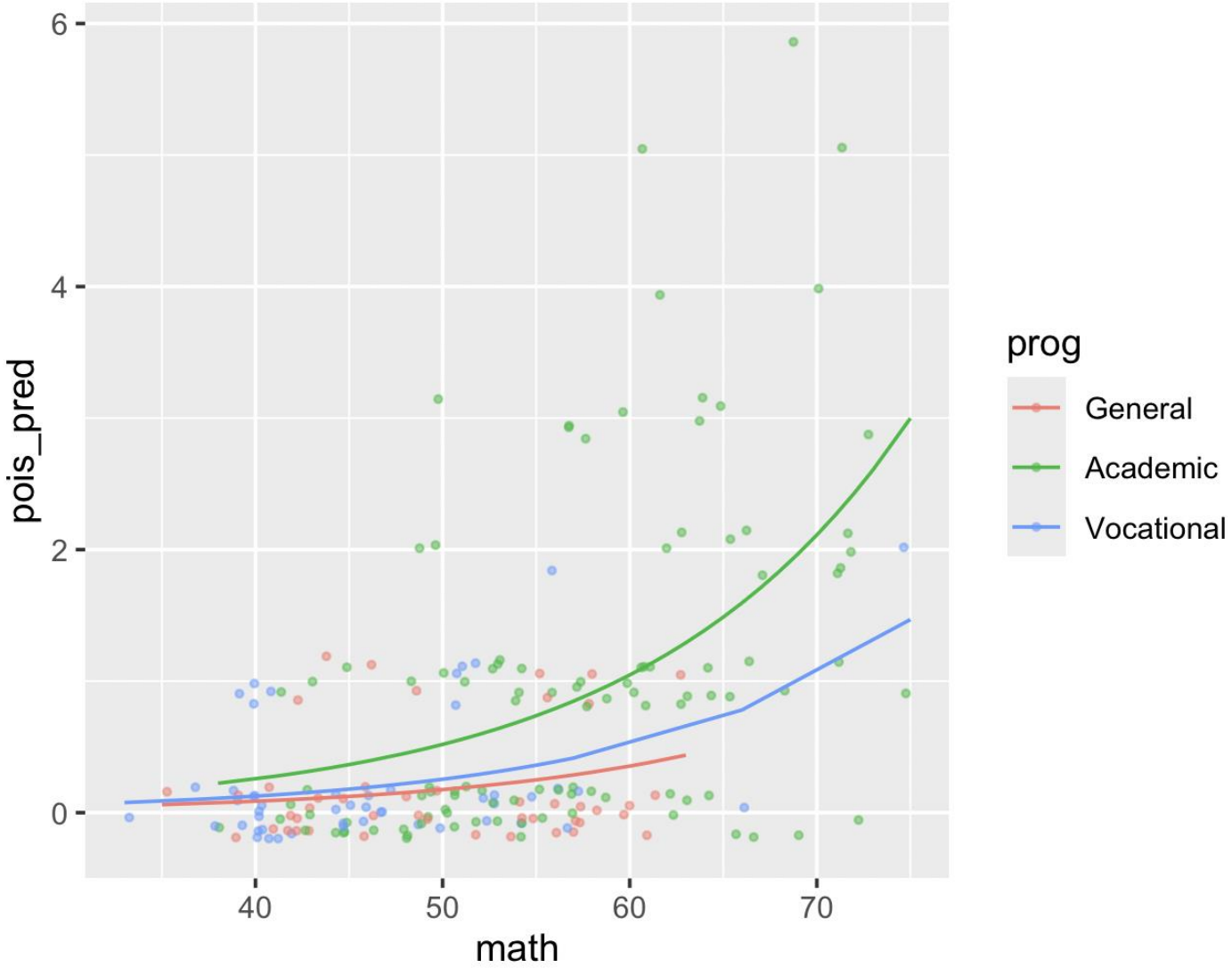
EXPANDING THE FRAMEWORK: GLM

```
Call:
glm(formula = num_awards ~ prog + math, family = "poisson", data =
pois_data)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  -5.24712    0.65845  -7.969 1.60e-15 ***
progAcademic  1.08386    0.35825   3.025  0.00248 **
progVocational  0.36981    0.44107   0.838  0.40179
math          0.07015    0.01060   6.619 3.63e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 287.67  on 199  degrees of freedom
Residual deviance: 189.45  on 196  degrees of freedom
AIC: 373.5

Number of Fisher Scoring iterations: 6
```

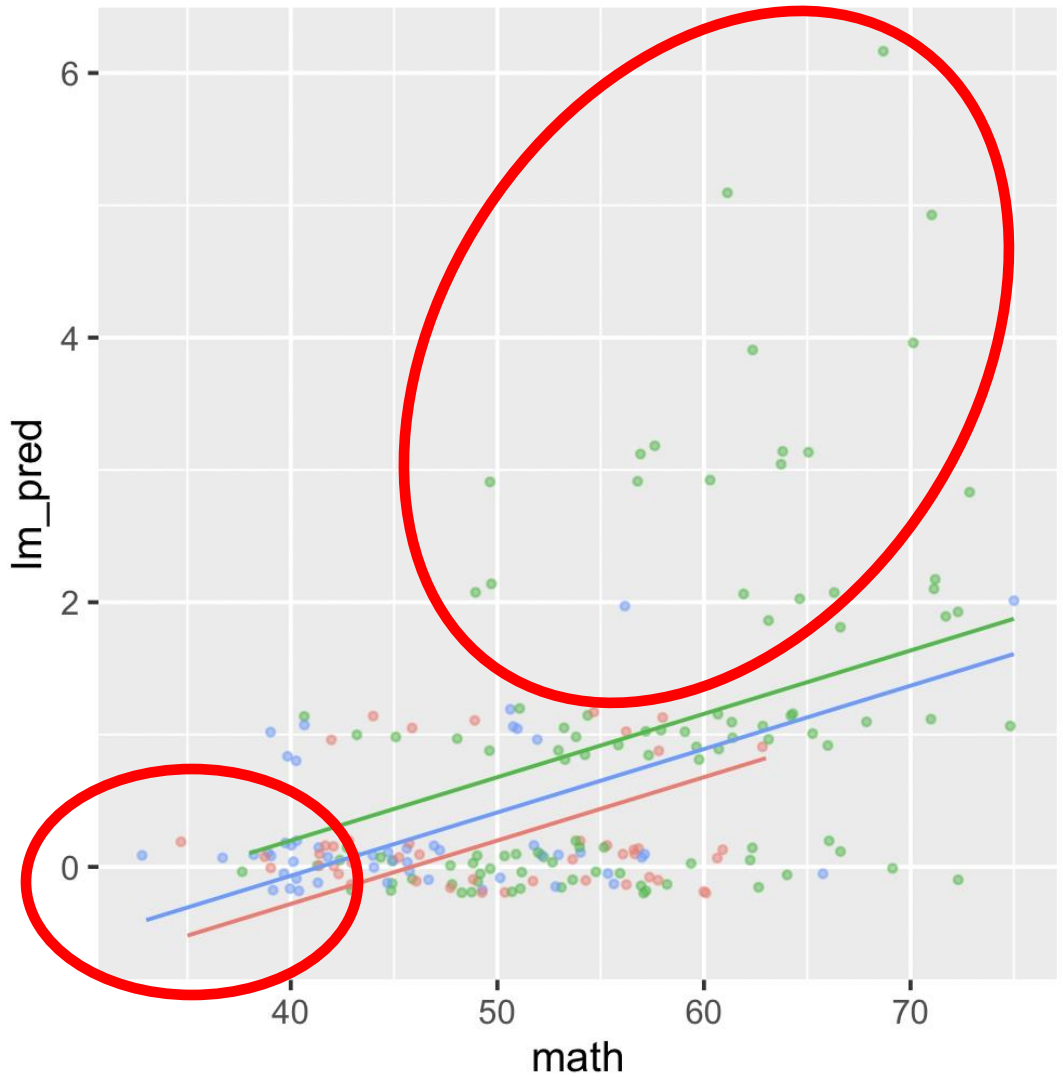
```
Call:
lm(formula = num_awards ~ prog + math, data = pois_data)

Residuals:
    Min      1Q  Median      3Q     Max
-1.7311 -0.5618 -0.1537  0.2851  4.4126

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.195504   0.411417  -5.336 2.60e-07 ***
progAcademic  0.478613   0.168956   2.833   0.0051 **
progVocational  0.212506   0.187433   1.134   0.2583
math          0.047889   0.007773   6.161 4.03e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9019 on 196 degrees of freedom
Multiple R-squared:  0.2773, Adjusted R-squared:  0.2662
F-statistic: 25.07 on 3 and 196 DF,  p-value: 9.016e-14
```

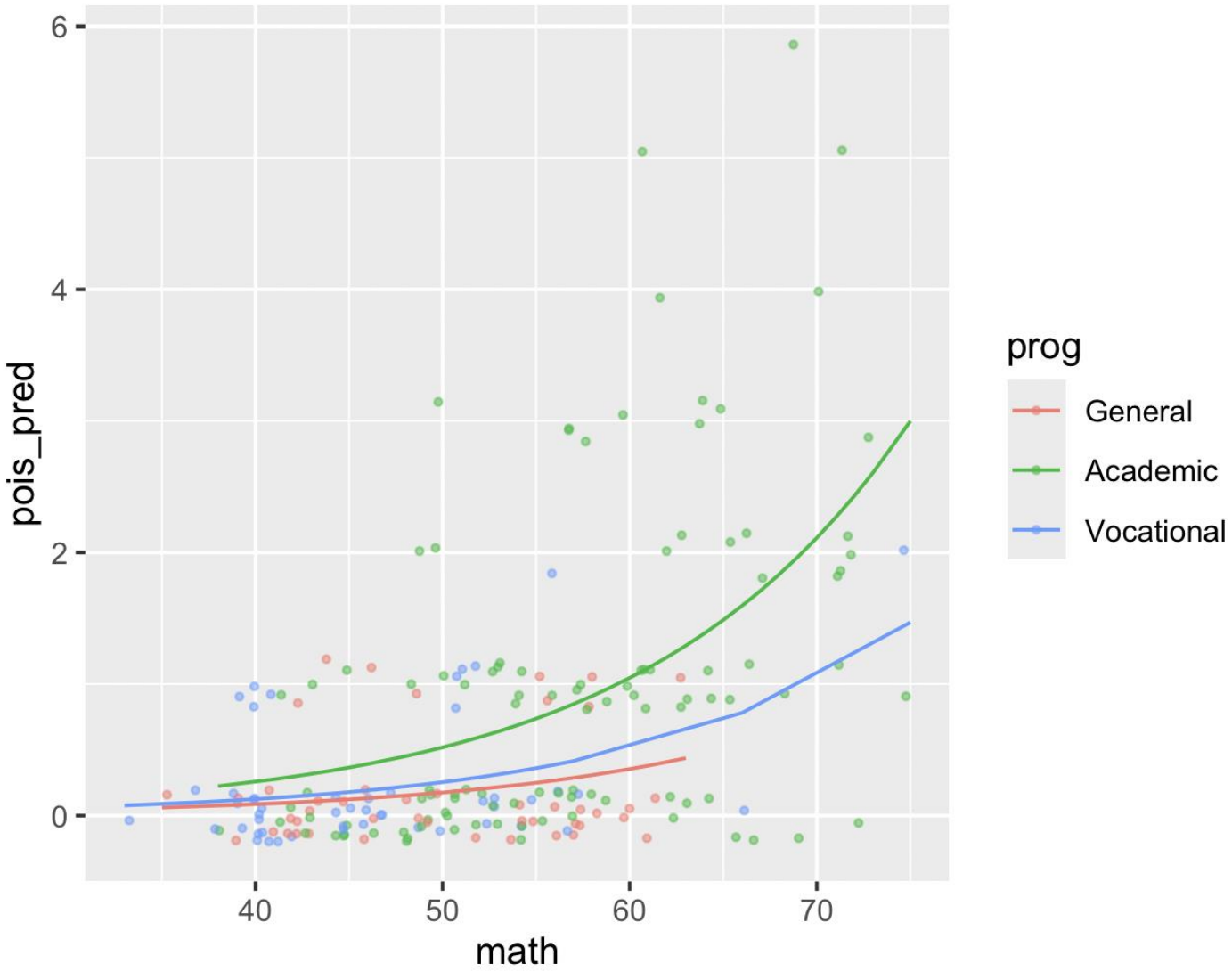
EXPANDING THE FRAMEWORK: GLM





EXPANDING THE FRAMEWORK: GLM

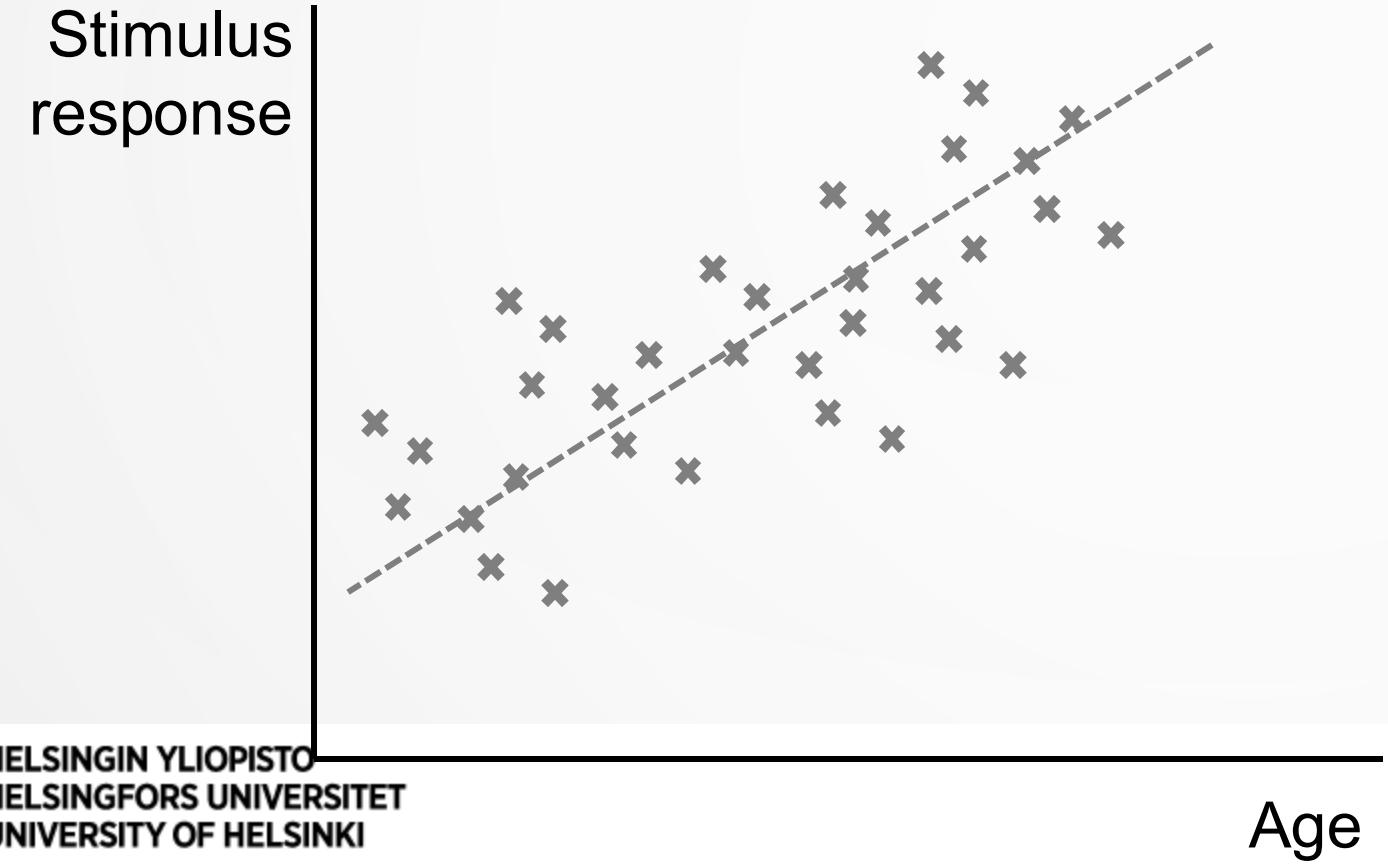




EXPANDING THE FRAMEWORK: GLM

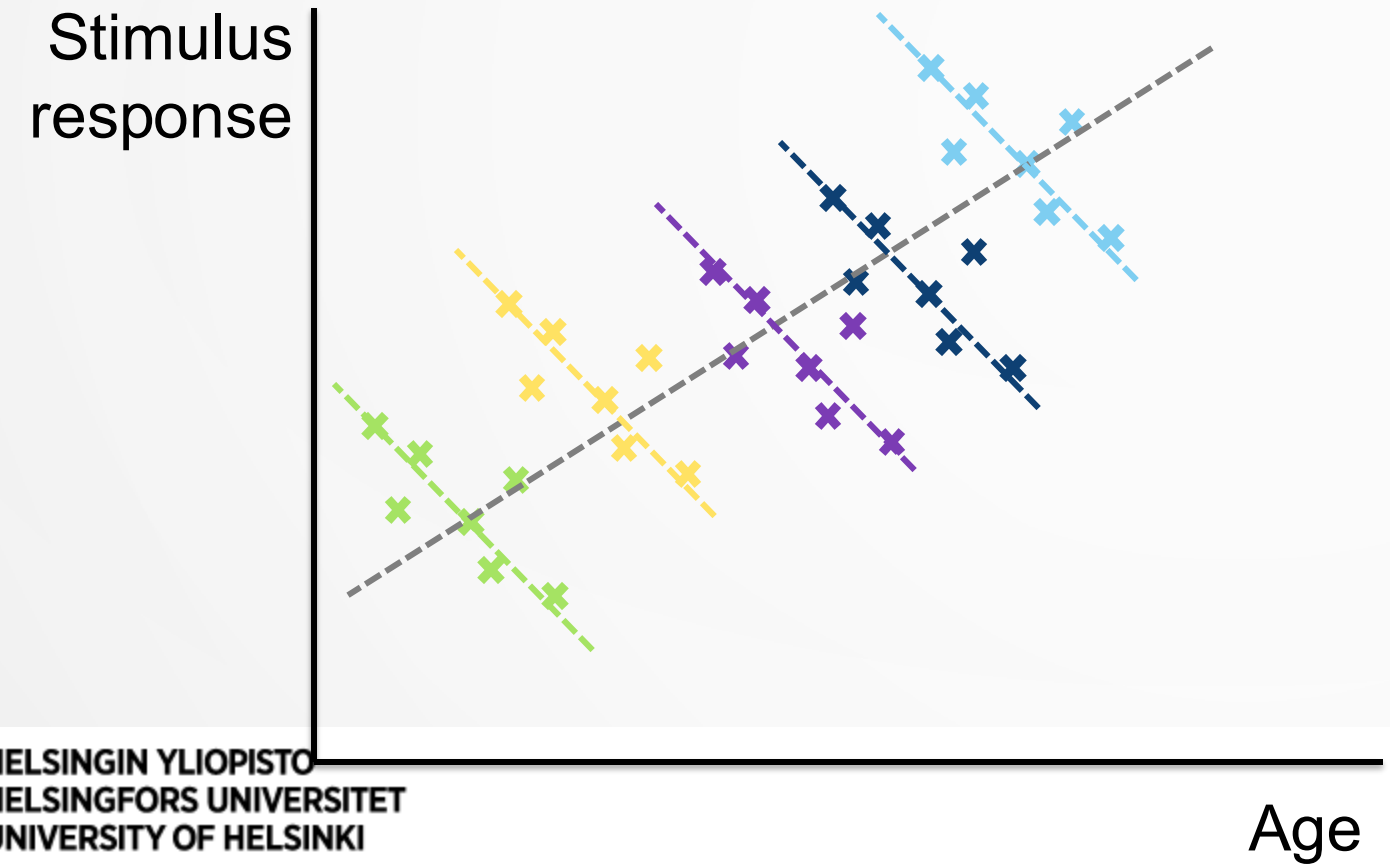
- N.B. Interpretation gets less straightforward!
  - Estimates in output table are on the scale of the linear relationship – for Poisson, this is log (binary logistic is a logit, etc.)
  - ”The change in log(y) for every unit increase in x” – fundamentally non-linear w.r.t raw response variable so not very meaningful to just exp(coef(model))
- Other generalizations exist: binomial (logistic), negative binomial (for more heavily skewed counts), beta (dispersion depends on a modelled variable), etc.
  - can be very tricky to figure out which model is best fit for your data
  - all models are wrong but some are useful! Your job is to make sure it’s useful enough

EXPANDING THE FRAMEWORK: LMM



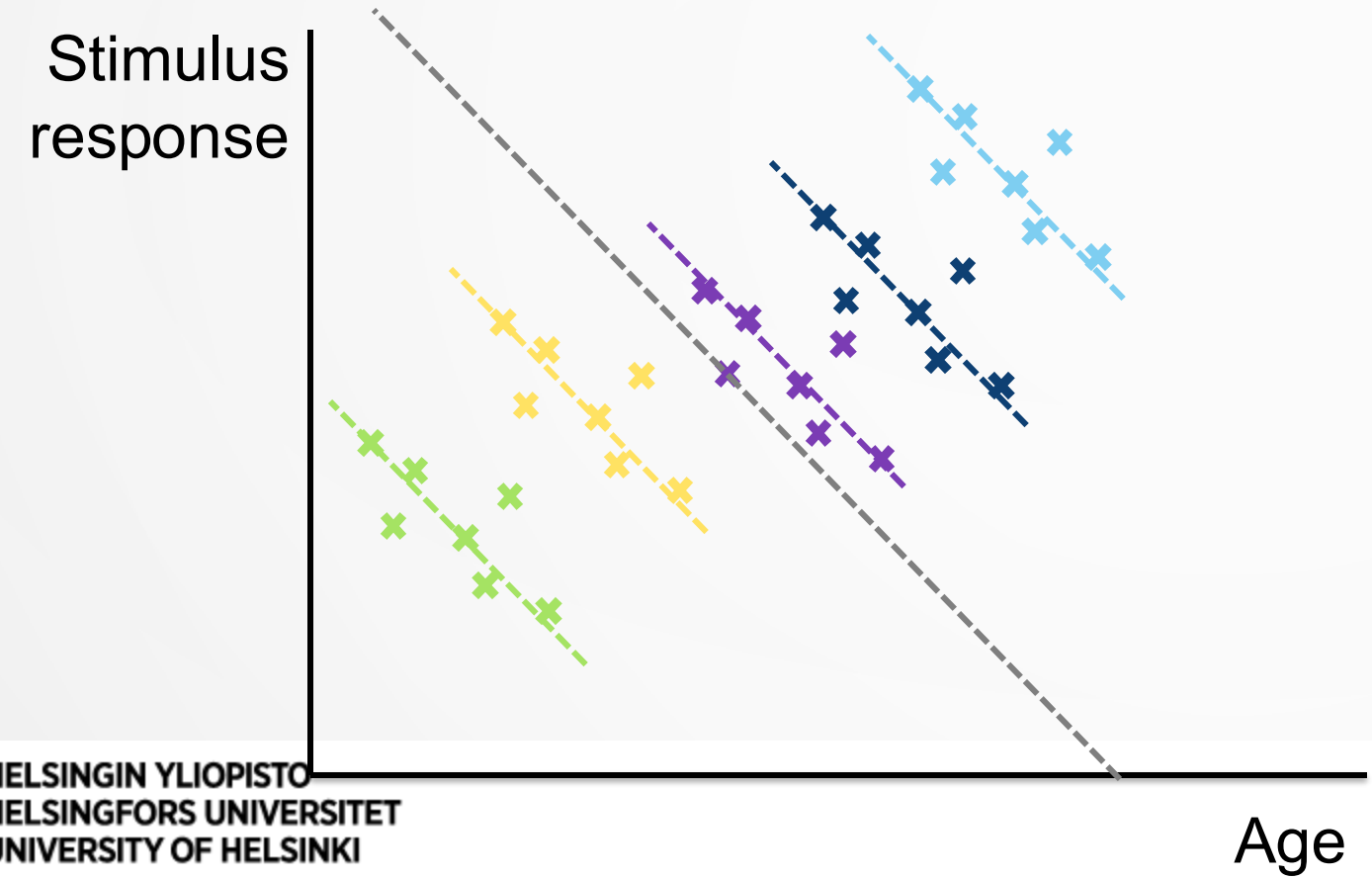
- (Also another common mistake)
- How to account for non-independence
  - clustering in data
  - repeated measures
  - pseudoreplication

EXPANDING THE FRAMEWORK: LMM



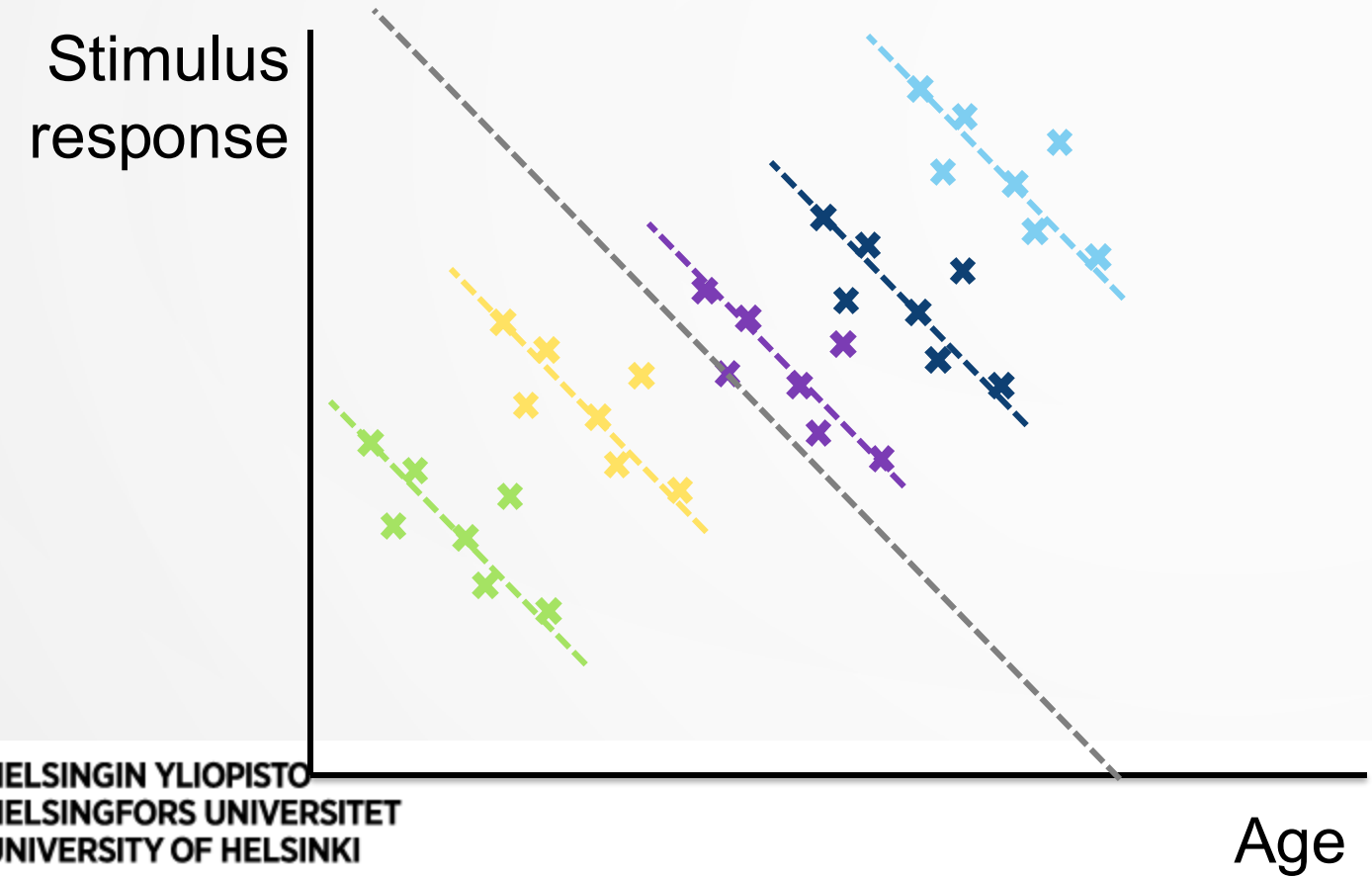
- Assuming independence, strong positive relationship
- Mixed effects models: fit an error structure that allows for associations between datapoints
  - Random effects
  - GLMMs also exist!

EXPANDING THE FRAMEWORK: LMM



- True relationship more complex
  - within individual negative -> aging as individual process decreases response
  - between-individual slightly positive -> at population level, older individuals respond more strongly

EXPANDING THE FRAMEWORK: LMM



- True relationship more complex
  - within individual negative -> aging as individual process decreases response
  - between-individual slightly positive -> at population level, older individuals respond more strongly
- Great tutorial/resource: https://ourcodingclub.github.io/tutorials/mixed-models/

SUMMARY

- R is your friend
- Investment now will pay off many times c.f. clunking around with SPSS
  - Learn R early in your career!
- For both R and statistics: lifelong learning experience, you never ”know enough”
- Need stats advice? Come to us before you collect the data!
- Coding with genAI: can be useful to get started, BUT need a good level of R fluency to judge what the code is actually doing (often unnecessarily complicated)
  - Best avoided until you have some confidence in your own abilities
  - Learning is quicker when you do it all yourself

FURTHER RESOURCES

- PhD (or independent) course: Statistical Methods for Medical Research by Matti Pirinen
- Book: Learn R as a language – Pedro Aphalo
- Bookdown/ebook: https://argoshare.is.ed.ac.uk/healthyr_book/

- Stackoverflow
- https://stats.oarc.ucla.edu/