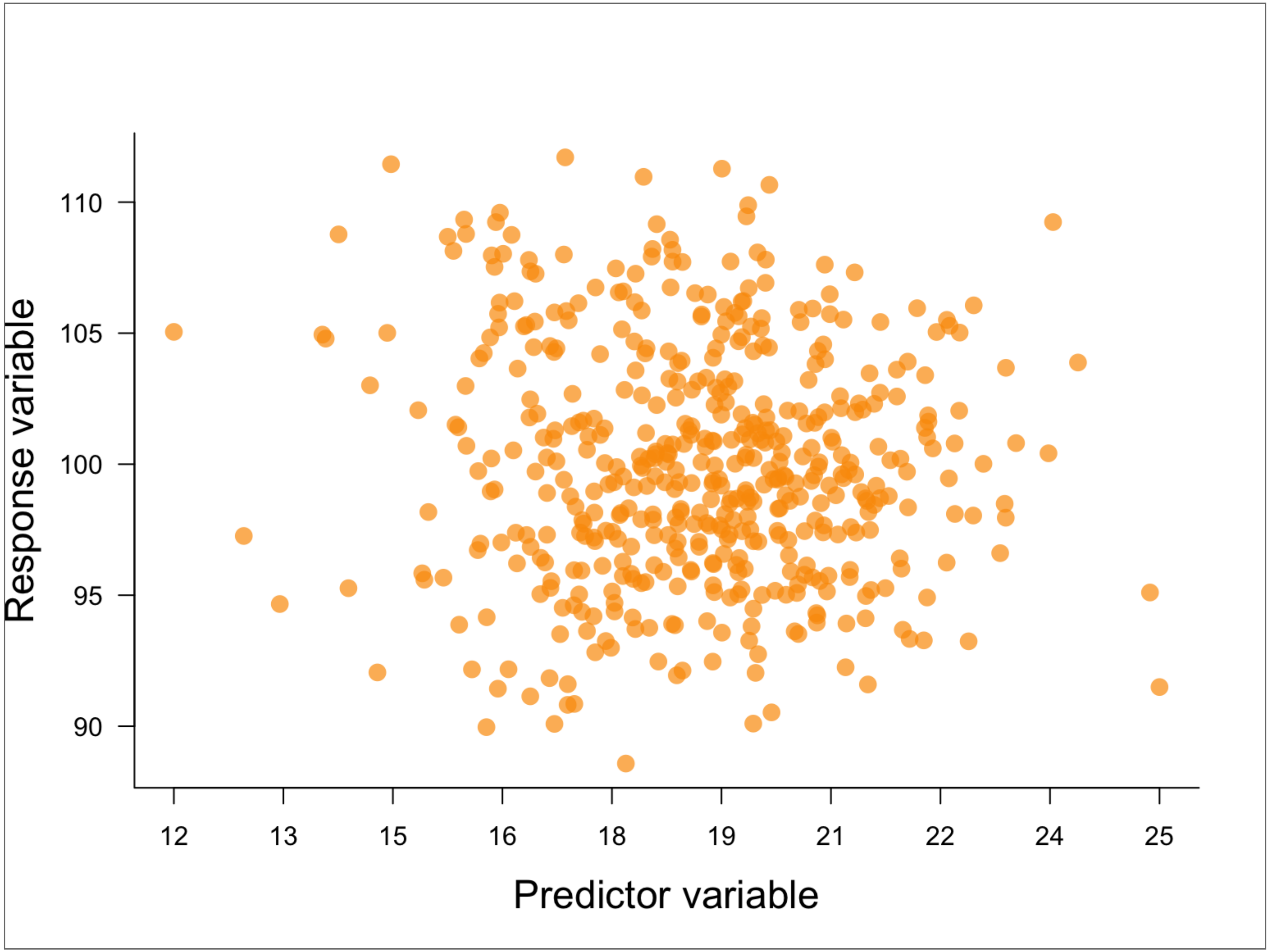
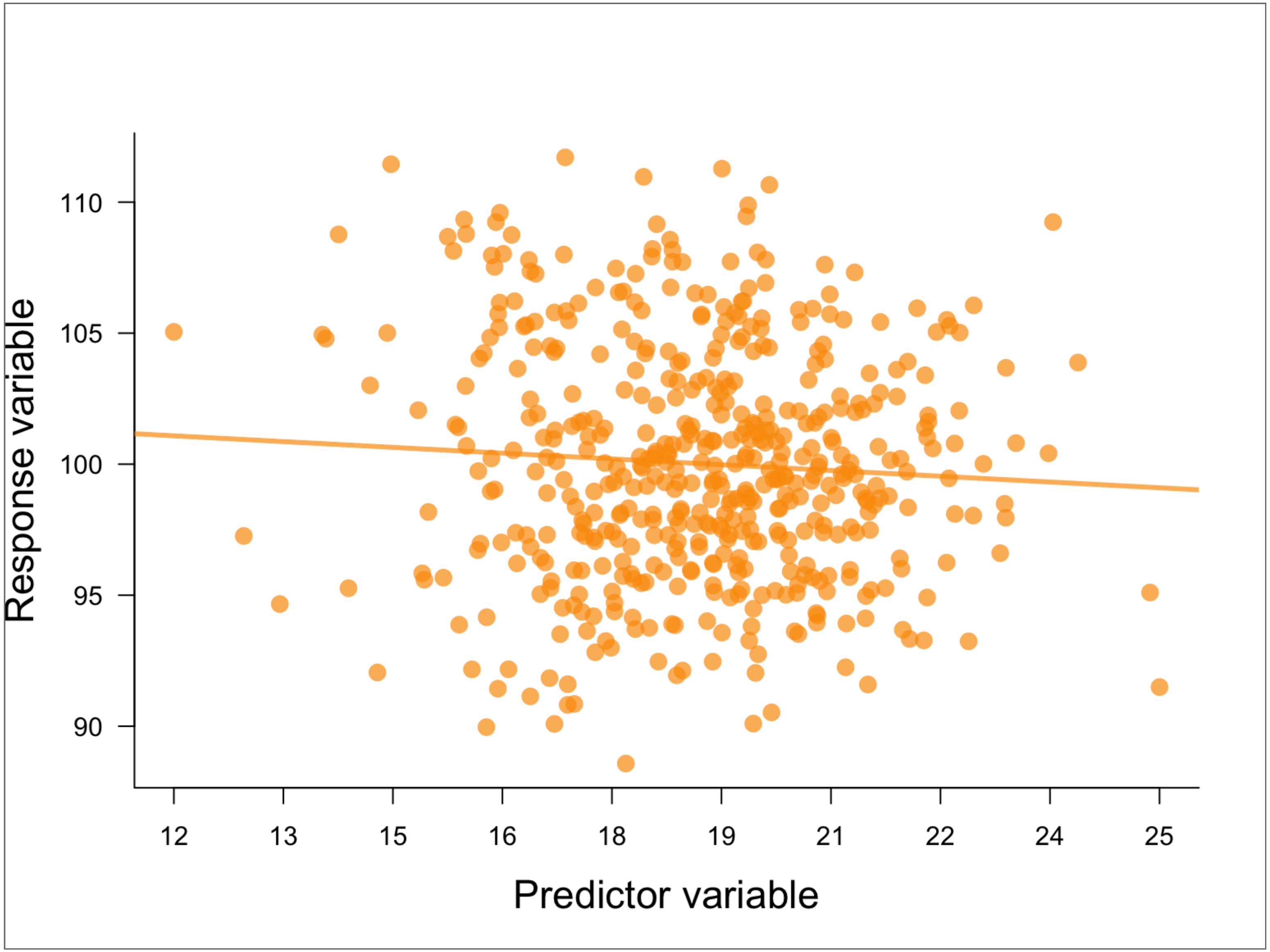
MIXED EFFECTS MODELS

2 NOVEMBER 2018

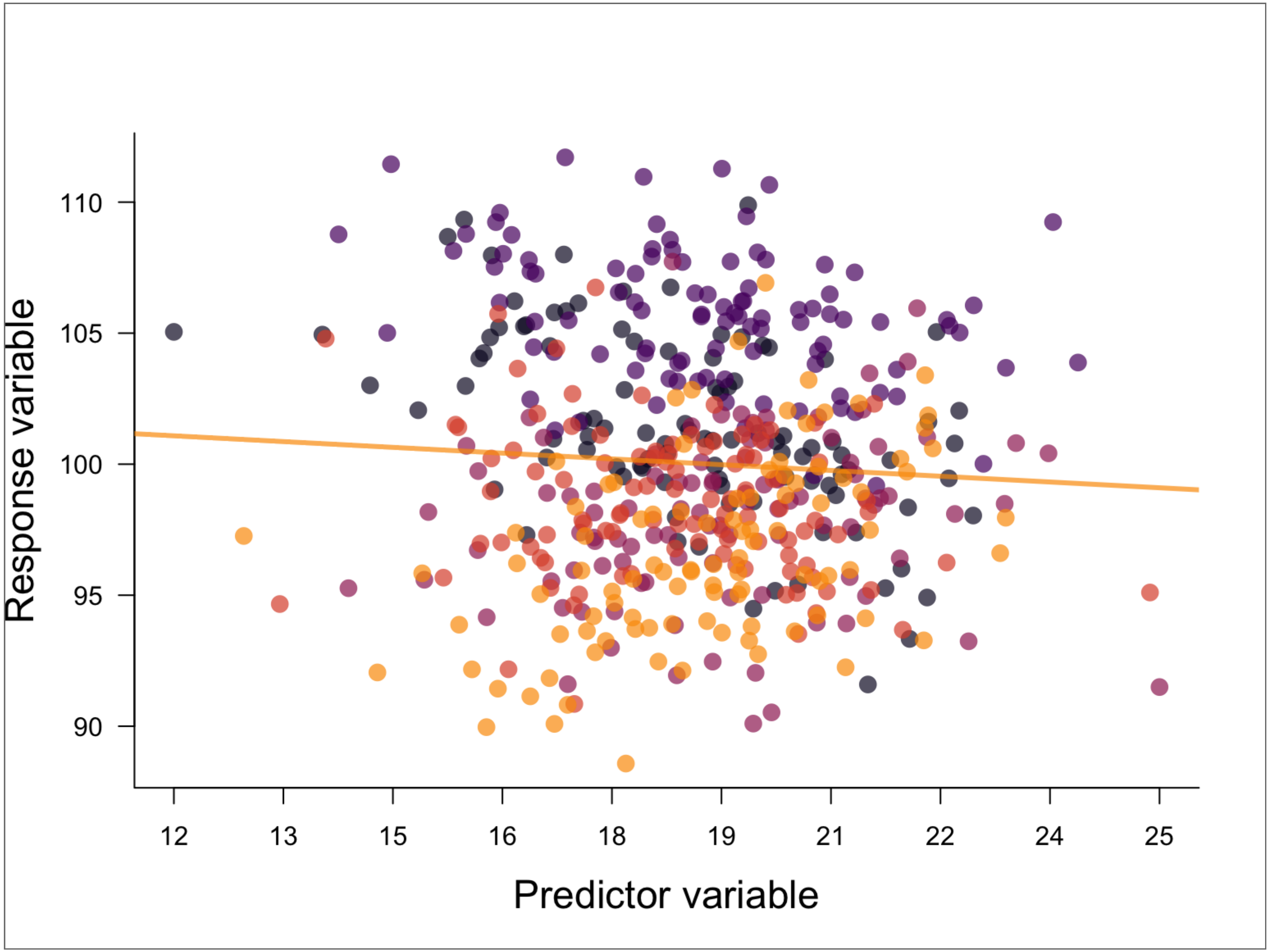
AN EXAMPLE


Response variable
110
105
100
95
90
12
13
15
16
18
19
21
22
24
25
Predictor variable

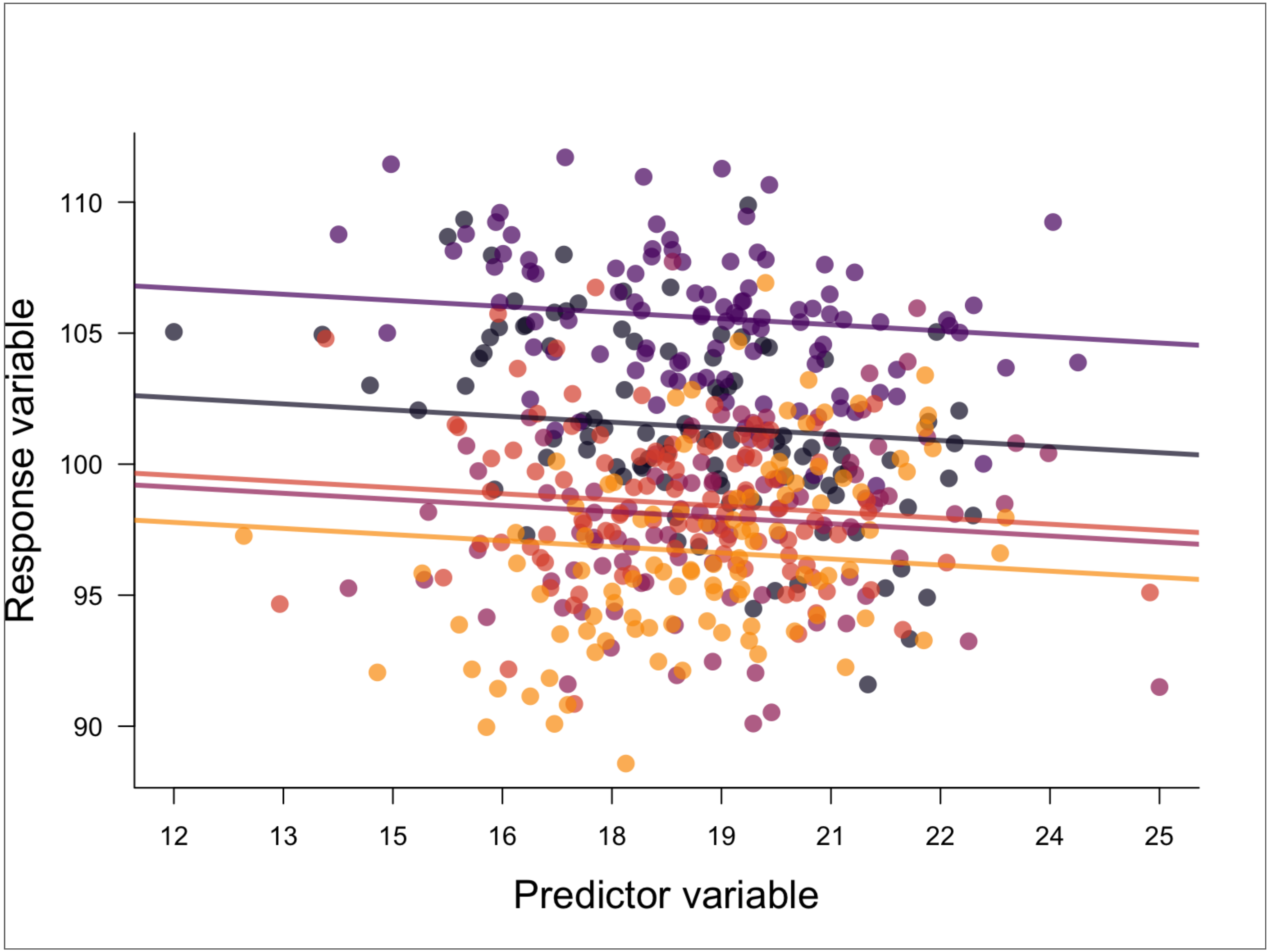
SINGLE REGRESSION LINE



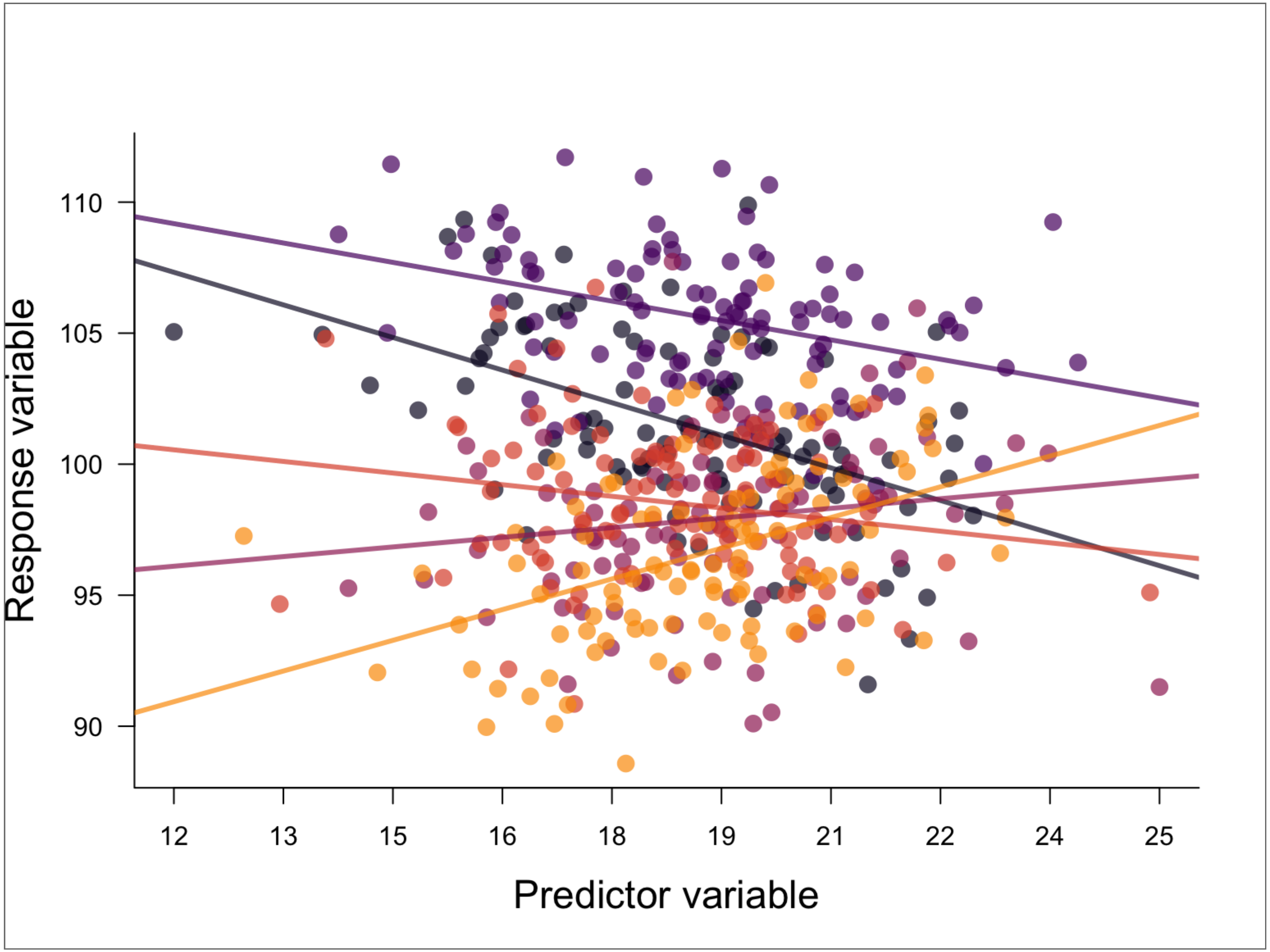
KNOWN SOURCES OF VARIATION



VARIATION IN INTERCEPTS



VARIATION IN INTERCEPTS AND SLOPES



VARIATION IN PARAMETERS

- in some cases, variation is directly of interest
- in other cases, it's a nuisance
  - can break independence assumptions
  - can introduce extra noise
- mixed models can help

MIXED MODELS

- mix of *fixed* and *random* effects
- these terms are not consistently defined
- in this context, really only matters for factors (categorical variables)

FIXED EFFECTS

- these are what we've used in general linear models
  - intercepts
  - slopes
  - interactions
- my definition: categories are independent

RANDOM EFFECTS

- parameters differ among categories but categories aren’t fully independent
- some definitions:
  - random if we haven’t sampled the entire population
  - random if we “don’t care” about the factor
  - random if there is some form of shrinkage

RANDOM EFFECTS

- examples: repeated measures, spatial blocks
  - can be a really good way to account for non-independent observations

- caveat: `lme4` methods often require >5 levels for random effects models to work

- be pragmatic (and check model fit!)

MIXED MODELS

- assumptions: much the same as a general linear model
- residuals are independent
- normally distributed residuals
- constant variance of residuals

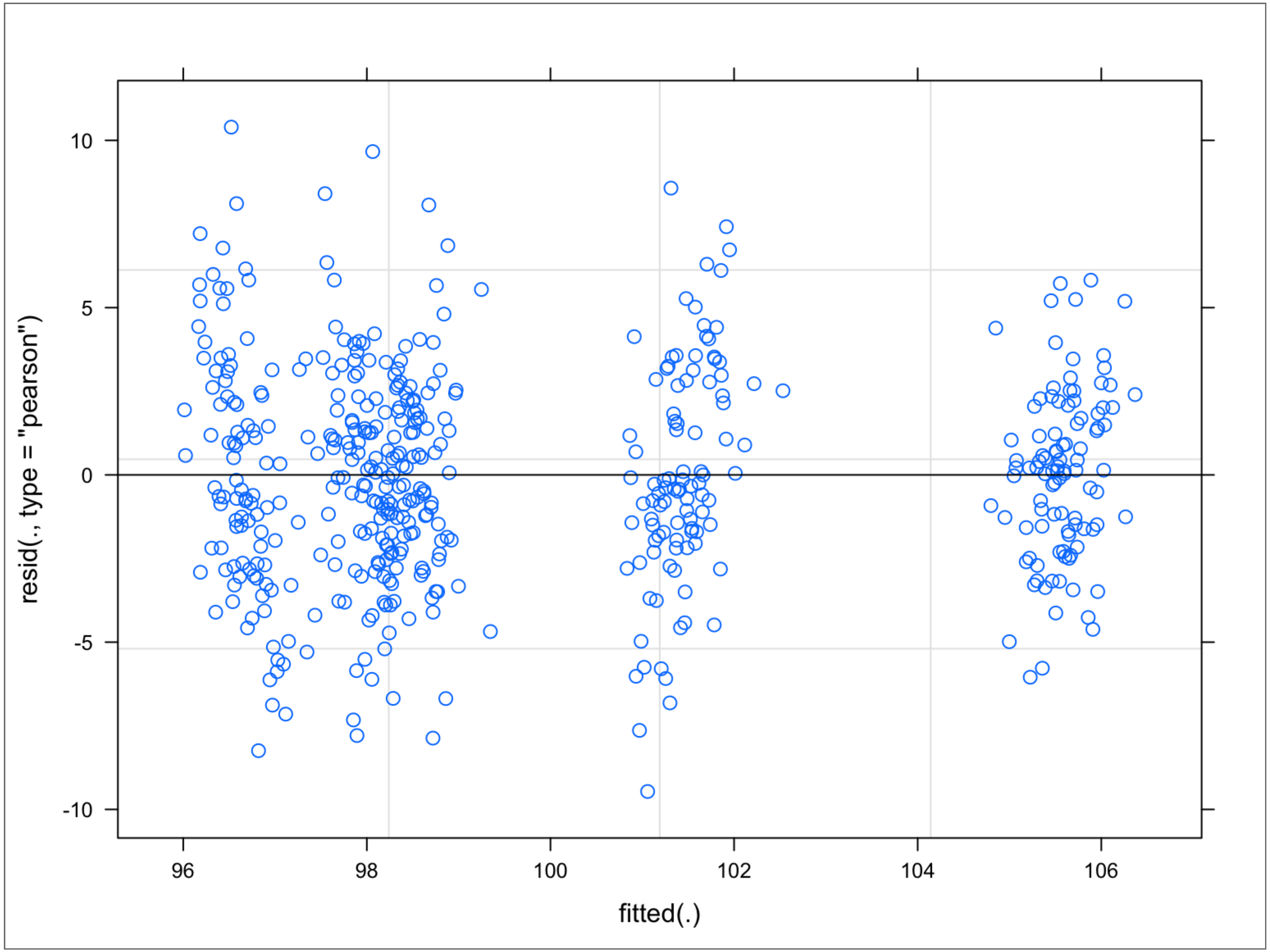
MIXED MODELS IN R

- use a formula interface to define models

MIXED MODELS IN R

```
# load lme4 package
library(lme4)

# fit model with single intercept and slope
mod_lm <- lm(response ~ predictor)

# fit model with random intercepts
mod_int <- lmer(response ~ predictor + (1 | block))

# fit model with random intercepts and slopes
mod_slope <- lmer(response ~ predictor + (1 + predictor | block))

# fit model with nested random intercepts
mod_slope <- lmer(response ~ predictor + (1 + predictor | block / nested_block))
```

PLOT FUNCTION



SUMMARY FUNCTION

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: y ~ x + (1 | z)
##
## REML criterion at convergence: 2593.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.9891 -0.6603 -0.0267  0.6876  3.2835
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  z        (Intercept) 12.82    3.580
##  Residual             10.03    3.166
## Number of obs: 500, groups:  z, 5
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept) 100.0136     1.6074  62.222
## x            -0.3119     0.1450  -2.151
##
## Correlation of Fixed Effects:
##   (Intr)
## x 0.004
```

MIXED MODELS IN R

```
# print the fixed effects
fixef(mod_int)
```

```
## (Intercept)           x
## 100.0135534  -0.3118507
```

```
# print the random effects
ranef(mod_int)
```

```
## $z
##   (Intercept)
## 1    1.387692
## 2    5.573009
## 3   -2.021700
## 4   -1.574244
## 5   -3.364757
```

MIXED MODELS IN R

```
# print the fixed effects
fixef(mod_slope)
```

```
## (Intercept)           x
##  99.9562825  -0.2379978
```

```
# print the random effects
ranef(mod_slope)
```

```
## $z
##   (Intercept)          x
## 1    1.317356 -1.4299245
## 2    5.629131 -0.7534335
## 3   -2.060671  0.7314980
## 4   -1.563628 -0.3570966
## 5   -3.322187  1.8089566
```

INTERPRETING RANDOM EFFECTS

- in short: don't!
- if you care about it, it might be better as a fixed effect
- however, can still look at "variance components"
  - technical term: variance partitioning

- `VarCorr(mod)` is useful for this (but so is `summary(mod)`)

MODEL ASSESSMENT AND MODEL SELECTION

- many different approaches (see Worksheet 1)
- start by assessing model fit
- but also need to assess model fit *for purpose*
- which model is “best”?
- my approach: often decide on random effects *a priori* and don’t “select” these