Statistical inference in practice

PSCI 2301: Quantitative Political Science II

Prof. Brenton Kenkel

brenton.kenkel@gmail.com

Vanderbilt University

February 5, 2025

Recap

Last time — theoretical foundations of statistical inference

1. Law of large numbers (LLN)
    - Sample mean converges to population mean
    - Key question for inference: How much data do we have?
2. Central limit theorem (CLT)
    - Sample mean is approximately normally distributed across samples
    - Lets us calculate “margin of error” given sample size
    - …or sample size we’d need for a given margin of error
3. Hypothesis testing: Leverage the CLT to set up tests with a known “failure” rate in case null hypothesis is true

Today's agenda

1. Inference for regression coefficients
    - Quick review of regression model
    - Influences on magnitude of standard errors
2. Calculations in R
    - Estimating treatment effects with regression
    - Calculating standard errors
    - Confidence intervals
3. Standard errors when observations aren't independent
    - Problem situation: above-predicted Y_i $\rightsquigarrow$ above-predicted Y_j
    - Why this leads to having less data than we think
    - Clustered standard errors in R

Inference for treatment effects

A quick regression recap

The bivariate linear regression model:

$$\mathbb{E}[Y_i \mid X_i = x] = \alpha + \beta x$$

α is the intercept, β is the slope

Regression with binary X

Suppose X_i is binary, so every $X_i = 0$ or $X_i = 1$.

Slope now equals difference of means:

$$\beta = \mathbb{E}[Y_i \mid X_i = 1] - \mathbb{E}[Y_i \mid X_i = 0]$$

Intercept now equals average in the 0 group:

$$\alpha = \mathbb{E}[Y_i \mid X_i = 0]$$

The regression slope and the difference of means

```
df_ggl <-
  read_csv("http://hdl.handle.net/10079/d3669799-4537-411e-b175-d9e837324c35") |>
  mutate(y = if_else(voted == "Yes", 1, 0))
```

```
## Difference of means
mean(df_ggl$y[df_ggl$treatment == "Neighbors"]) -
  mean(df_ggl$y[df_ggl$treatment == "Control"])
```

```
[1] 0.08130991
```

```
## Regression coefficients
df_ggl <- df_ggl |>
    mutate(treat = case_when(
           treatment == "Neighbors" ~ 1,
           treatment == "Control" ~ 0,
           TRUE ~ NA
         ))
lm(y ~ treat, data = df_ggl)
```

```
Call:
lm(formula = y ~ treat, data = df_ggl)

Coefficients:
(Intercept)        treat
    0.29664      0.08131
```

The standard error of the regression slope

Slope estimate formula:

$$\hat{\beta} = \frac{\mathbf{cov}[X_i, Y_i]}{\mathbf{var}[X_i]}$$

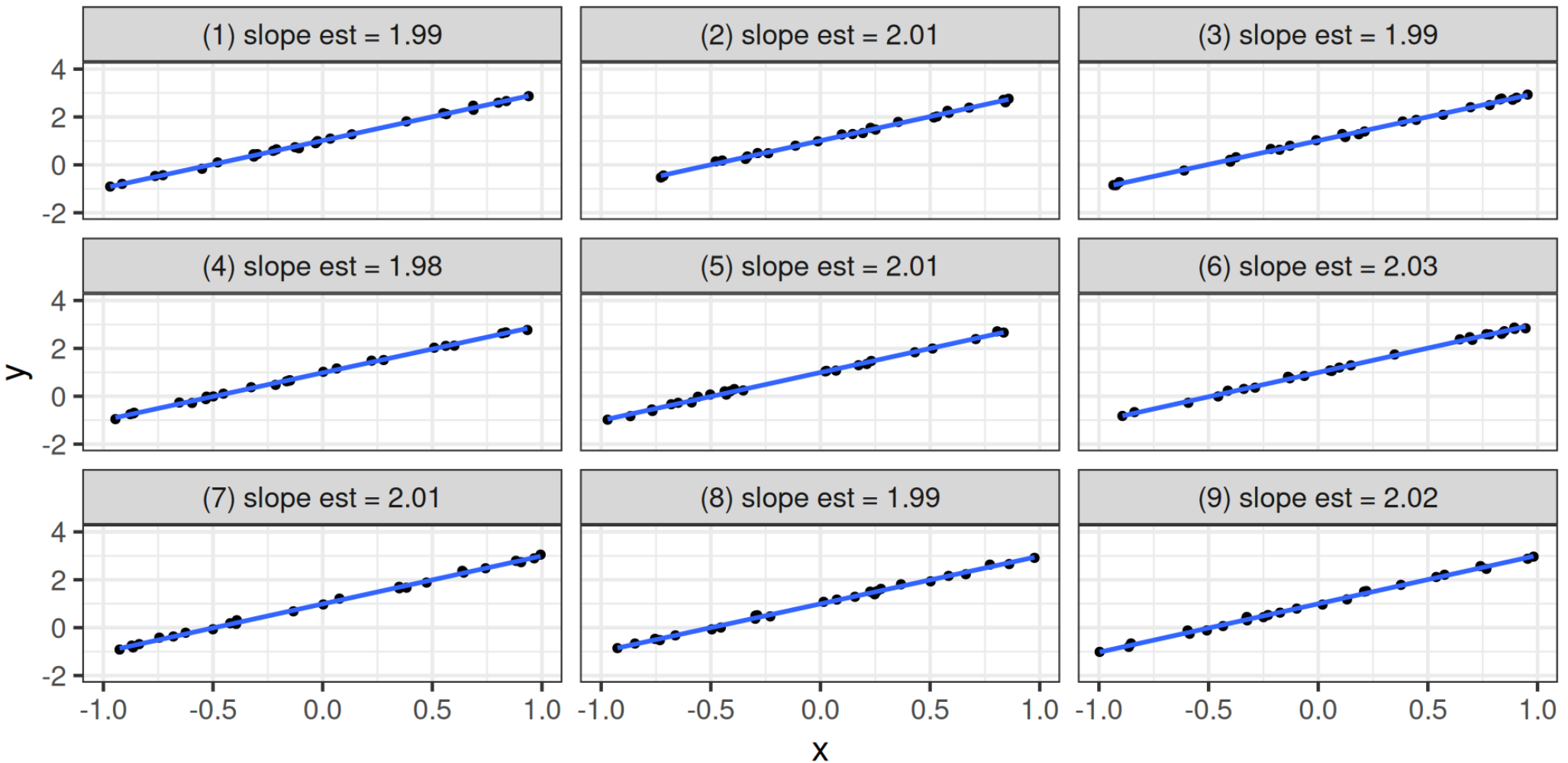
Standard error — extent of variation in estimate across samples of size N:

$$\mathbf{sd}[\hat{\beta}] = \frac{\mathbf{sd}[Y_i - \hat{\alpha} - \hat{\beta} X_i]}{\sqrt{N}\,\mathbf{sd}[X_i]} = \frac{1}{\sqrt{N}} \cdot \frac{\textbf{residual std deviation}}{\textbf{std deviation of ind var}}$$

- More "unpredictable" outcome $\rightsquigarrow$ bigger residuals $\rightsquigarrow$ higher std error
- Greater sample size $\rightsquigarrow$ lower std error
- More variation in independent variable $\rightsquigarrow$ lower std error
  - $\rightarrow$ for a binary treatment, most accurate results if assignment is 50-50

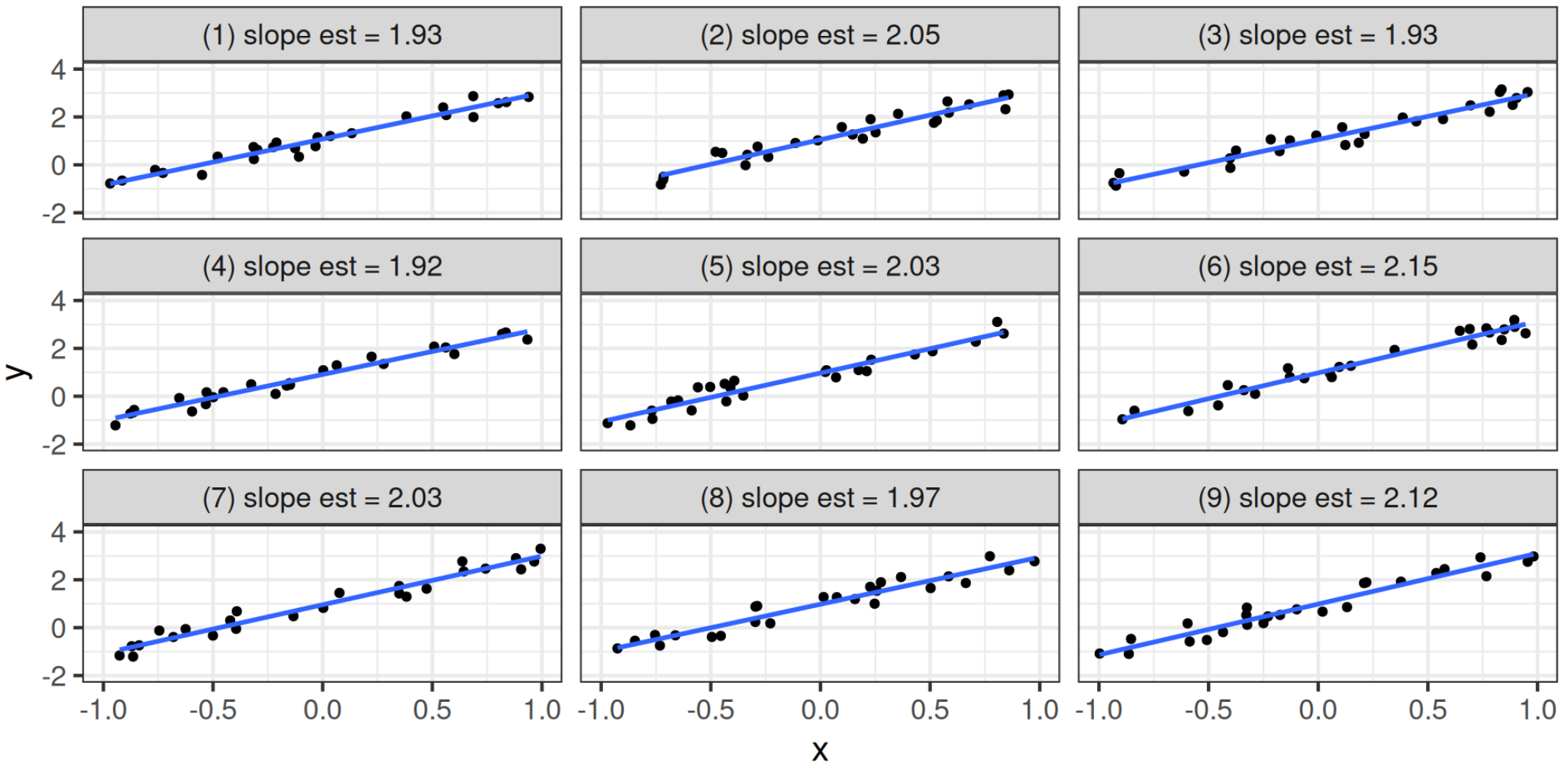
Residual variance increases standard error

Data simulated from regression equation $Y_i = 1 + 2X_i + \epsilon_i$



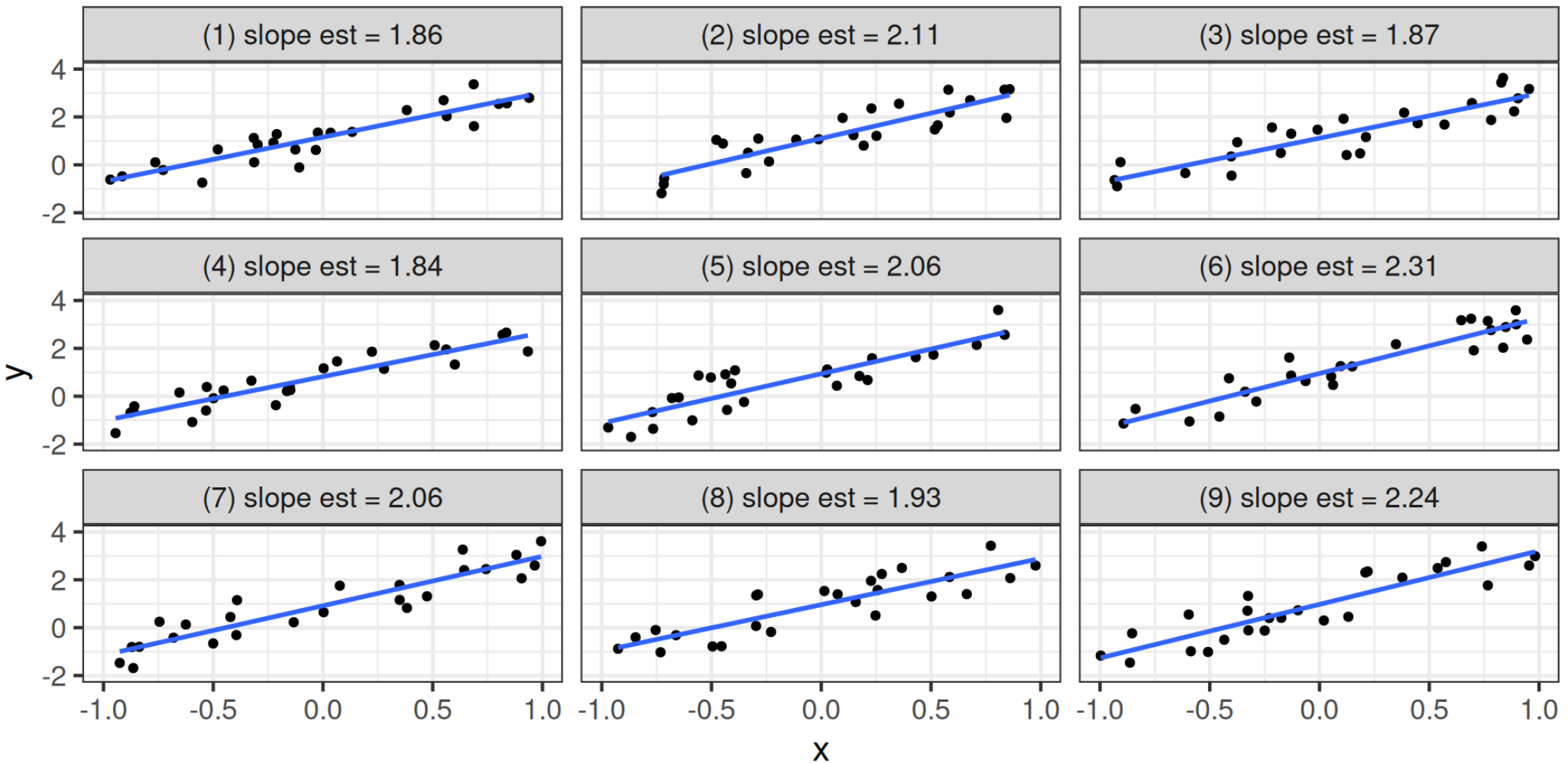
Residual variance increases standard error

Data simulated from regression equation $Y_i = 1 + 2X_i + \epsilon_i$



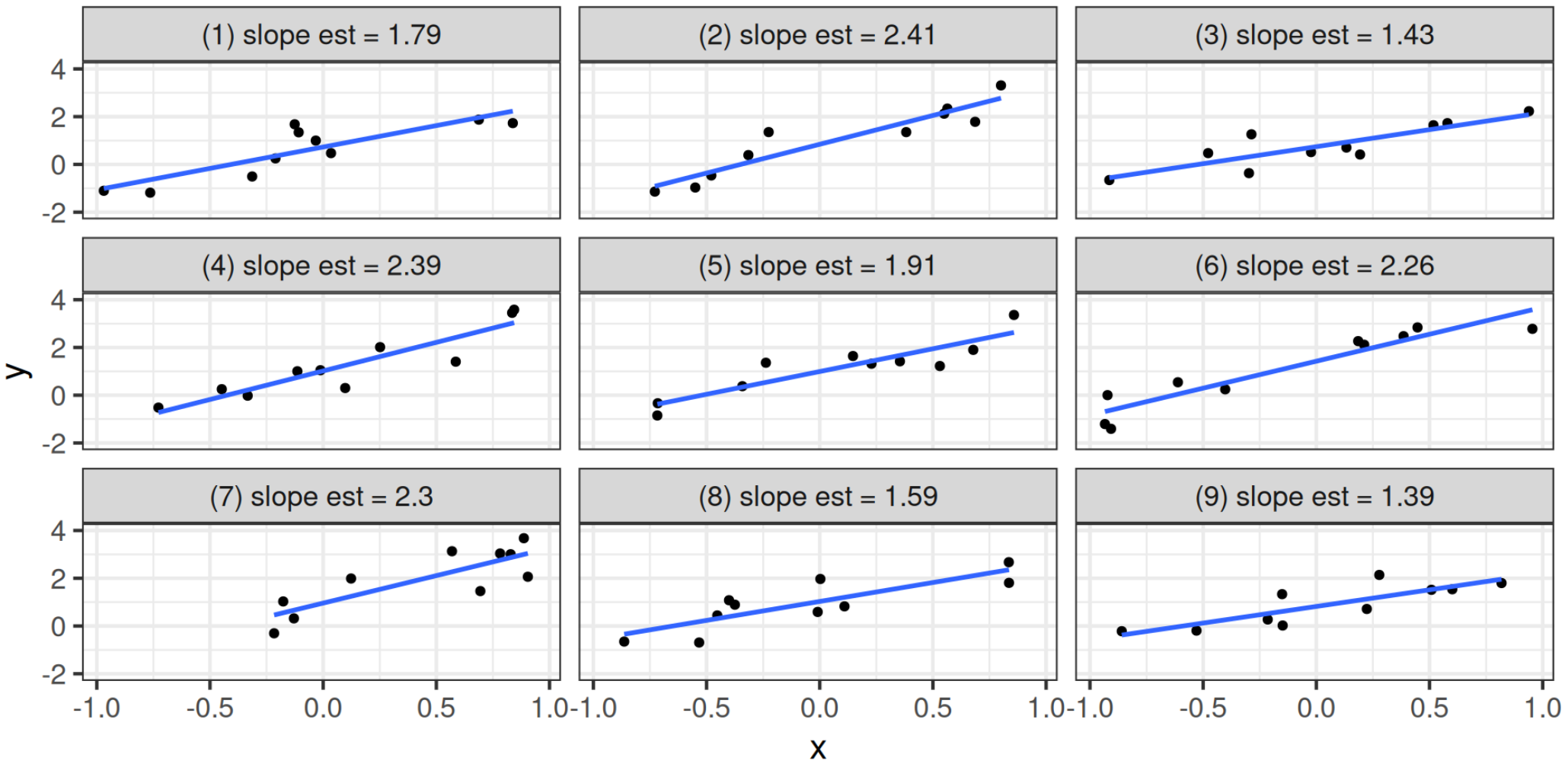
Residual variance increases standard error

Data simulated from regression equation $Y_i = 1 + 2X_i + \epsilon_i$



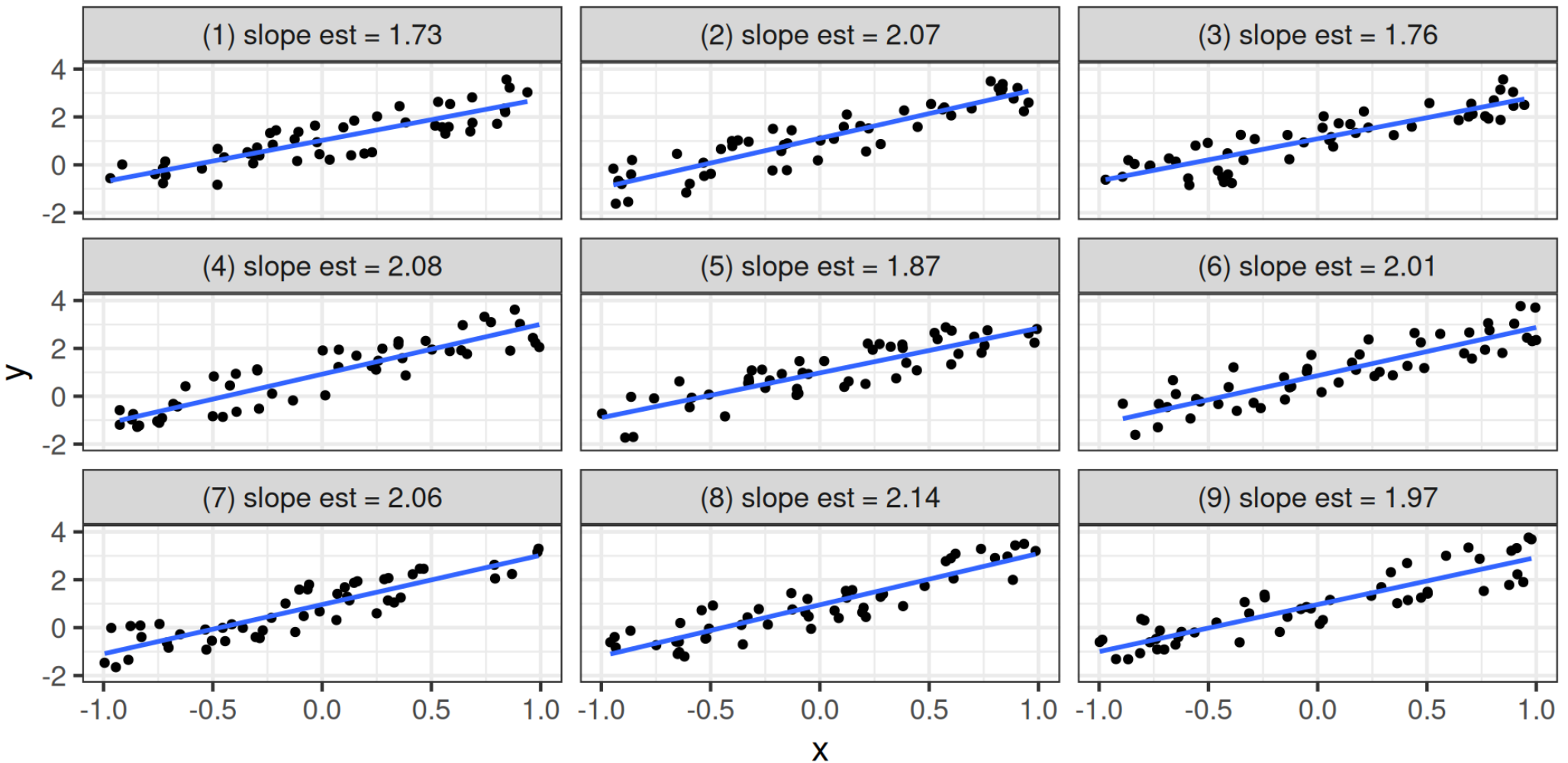
Sample size decreases standard error

Data simulated from regression equation $Y_i = 1 + 2X_i + \epsilon_i$



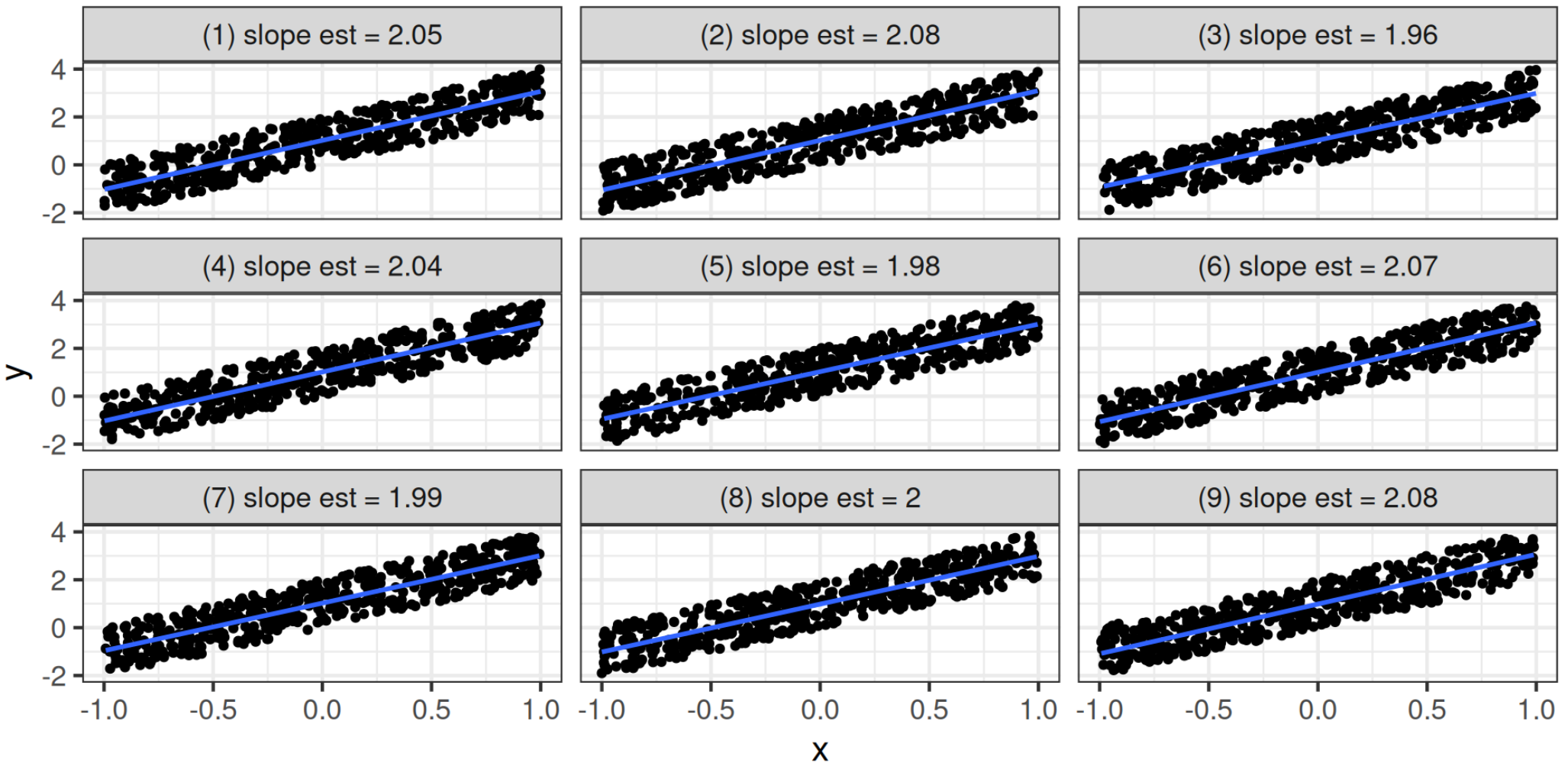
Sample size decreases standard error

Data simulated from regression equation $Y_i = 1 + 2X_i + \epsilon_i$



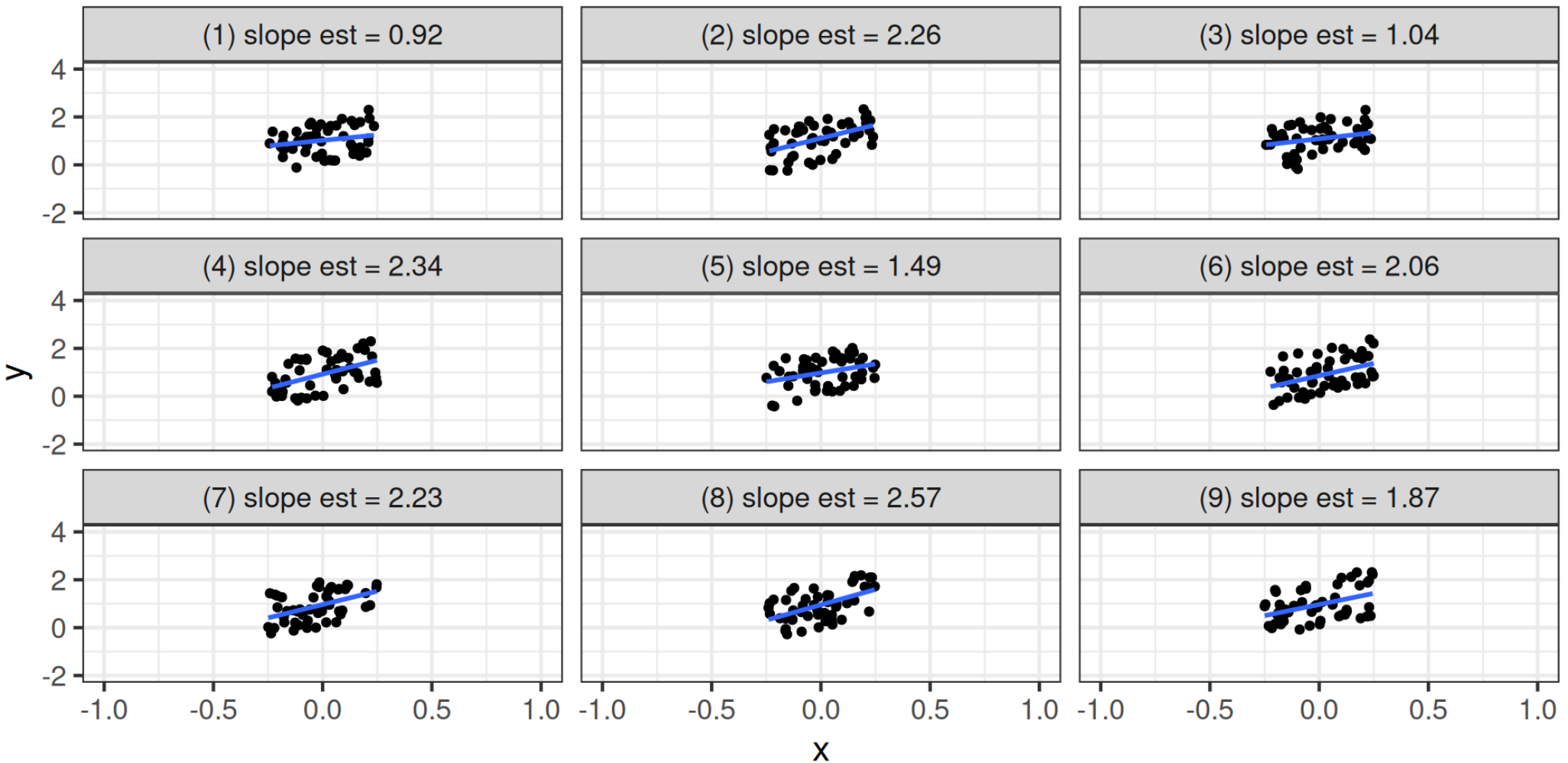
Sample size decreases standard error

Data simulated from regression equation $Y_i = 1 + 2X_i + \epsilon_i$



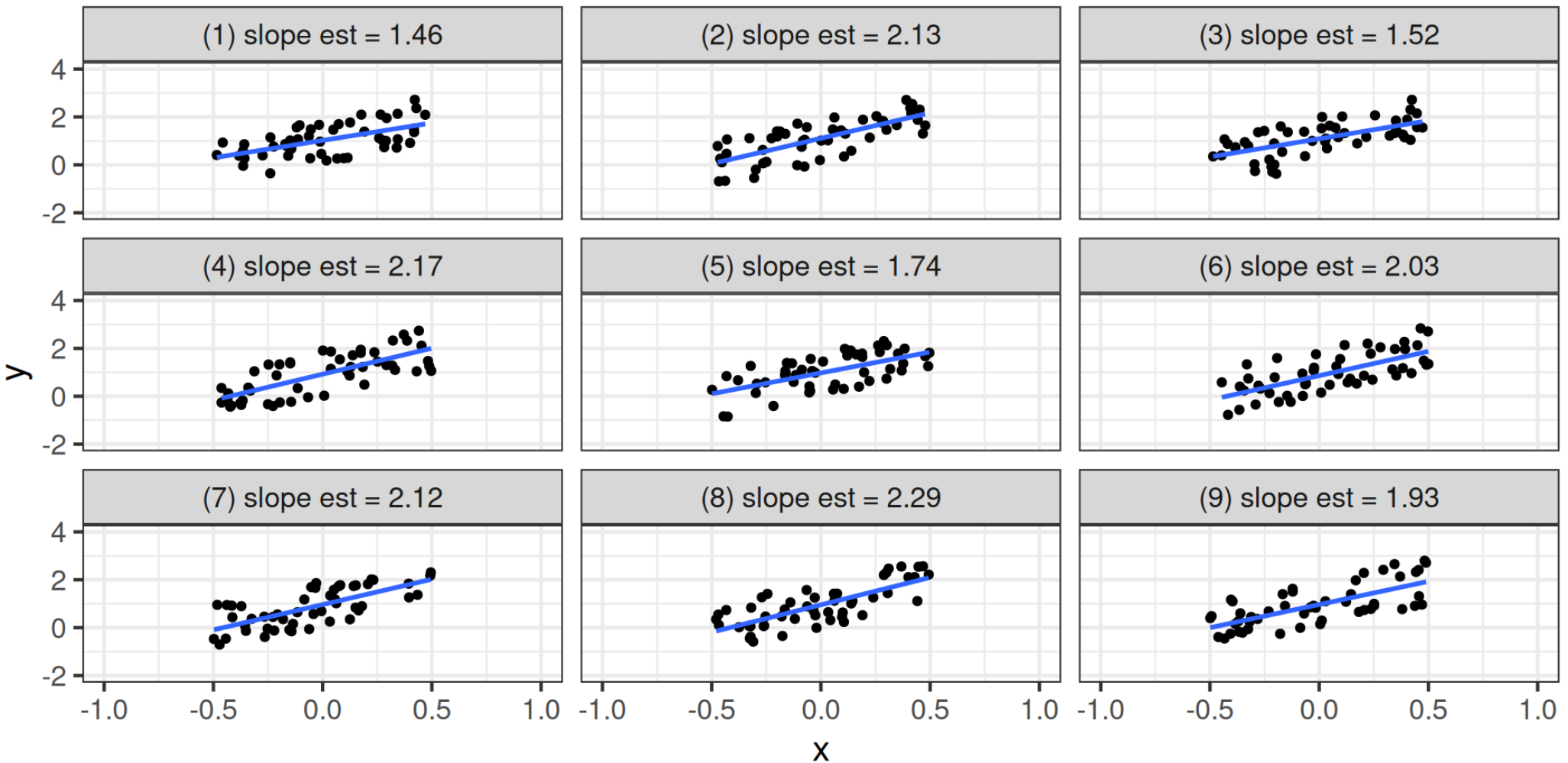
X variation decreases standard error

Data simulated from regression equation $Y_i = 1 + 2X_i + \epsilon_i$



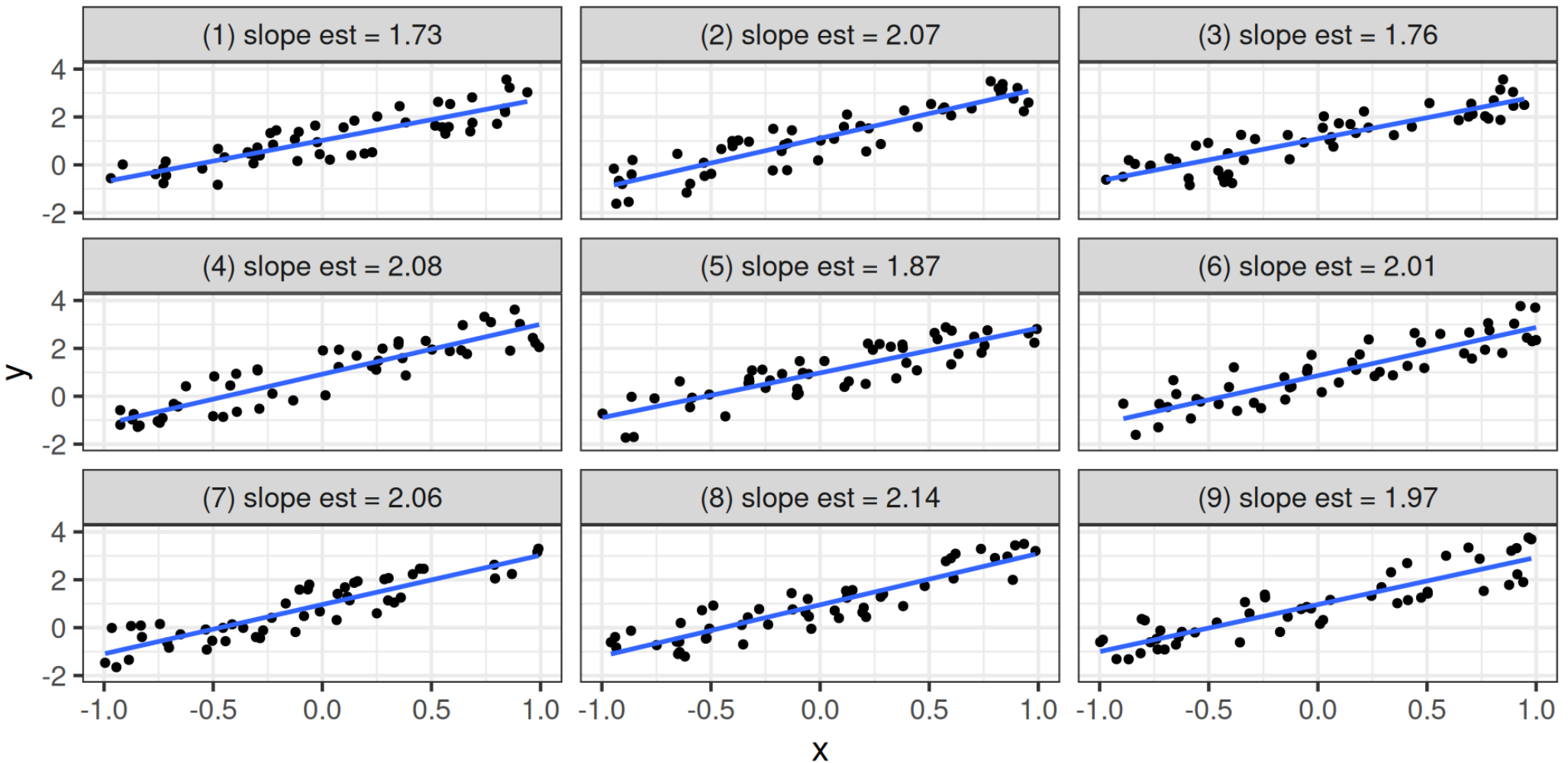
X variation decreases standard error

Data simulated from regression equation $Y_i = 1 + 2X_i + \epsilon_i$



X variation decreases standard error

Data simulated from regression equation $Y_i = 1 + 2X_i + \epsilon_i$



Calculating standard errors

If observations are independent ...

→ i.e., Y_i above regression line doesn't predict whether Y_j is above or below

... can use the `lm()` default standard errors:

```
fit_neighbors <- lm(y ~ treat, data = df_ggl)
summary(fit_neighbors)
```

```
Call:
lm(formula = y ~ treat, data = df_ggl)

Residuals:
    Min      1Q  Median      3Q     Max
-0.3780 -0.2966 -0.2966  0.6220  0.7034

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.296638   0.001055  281.05   <2e-16 ***
treat       0.081310   0.002587   31.43   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.4616 on 229442 degrees of freedom
```

Calculating standard errors

To extract as a data frame:

```
library("broom")
tidy(fit_neighbors)
```

```
# A tibble: 2 × 5
  term        estimate std.error statistic   p.value
  <chr>          <dbl>     <dbl>     <dbl>     <dbl>
1 (Intercept)   0.297    0.00106     281.  0
2 treat         0.0813   0.00259      31.4 2.04e-216
```

- test statistic: coefficient divided by standard error
- p value: probability of seeing a slope this large in a sample of this size if population slope were 0
    - → Will be 0.05 (5%) or less if |test statistic| ≥ 2

Calculating confidence intervals

For 95% of samples, the confidence interval $\hat{\beta} \pm 2\,\mathrm{se}[\hat{\beta}]$ contains the population slope

```
tidy(fit_neighbors) |>
  mutate(lower = estimate - 2 * std.error, upper = estimate + 2 * std.error)
```

```
# A tibble: 2 × 7
  term        estimate std.error statistic   p.value  lower  upper
  <chr>          <dbl>     <dbl>     <dbl>     <dbl>  <dbl>  <dbl>
1 (Intercept)   0.297    0.00106     281.  0          0.295  0.299
2 treat         0.0813   0.00259      31.4 2.04e-216 0.0761 0.0865
```

For confidence level of z%, use `qnorm(0.5 * (1 + z))` in place of 2:

```
## 99% confidence
qnorm(0.5 * (1 + 0.99))
```

```
[1] 2.575829
```

Inference with multiple treatment groups

```r
fit_all <- lm(y ~ treatment, data = df_ggl)
tidy(fit_all)
```

```
# A tibble: 5 × 5
  term               estimate std.error statistic  p.value
  <chr>                 <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)          0.315     0.00237    132.   0
2 treatmentControl    -0.0179    0.00260     -6.88 5.85e-12
3 treatmentHawthorne   0.00784   0.00336      2.33 1.96e- 2
4 treatmentNeighbors   0.0634    0.00336     18.9  1.64e-79
5 treatmentSelf        0.0306    0.00336      9.12 7.64e-20
```

- One group will always be omitted from the regression
  - → Here, it's the "Civic" treatment
- Intercept = average outcome among omitted group
- Each slope = difference relative to omitted group
  - → e.g., Neighbors increases turnout by 6.34% compared to Civic
- p value = probability of difference this big if 0 in population

Inference with multiple treatment groups

By default, R omits the first category in alphabetical order. To change that...

```
## Use fct_relevel() to tell R which category to put first
df_ggl <- df_ggl |>
  mutate(treatment = fct_relevel(treatment, "Hawthorne"))

## Rerun regression on modified data
fit_all_vs_hawthorne <- lm(y ~ treatment, data = df_ggl)
tidy(fit_all_vs_hawthorne)
```

```
# A tibble: 5 × 5
  term                estimate std.error statistic  p.value
  <chr>                  <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)          0.322     0.00237    136.   0
2 treatmentCivic Duty -0.00784   0.00336     -2.33 1.96e- 2
3 treatmentControl    -0.0257    0.00260     -9.90 4.37e-23
4 treatmentNeighbors   0.0556    0.00336     16.6  1.69e-61
5 treatmentSelf        0.0228    0.00336      6.78 1.17e-11
```

A general warning

Statistically significant regression $\neq$ Existence of causal effect

Whether you can interpret a regression coefficient as a causal effect depends on the design of your study, not whether the coefficient is statistically significant.

- Causal inference comes from random treatment assignment
  - → ...or a close-enough approximation thereof
- SEs/significance just tell you how precise your measurement is
- These are largely a measure of sample size
  - → With a big enough sample, significance virtually guaranteed

Dealing with spillovers

Samples can be smaller than they appear

Imagine study A, with $N = 1000$ data points

- Whether each person $i = 1, \ldots, 1000$ voted in the 2024 general election

...versus study B, with $N = 3000$ data points

- Whether each person $i = 1, \ldots, 1000$ voted for president in 2024
- Whether each person $i = 1, \ldots, 1000$ voted for US Senate in 2024
- Whether each person $i = 1, \ldots, 1000$ voted for US House in 2024

Does study B really have 3x as much data as study A?

Independence and true sample size

The issue with study B:

- Voting in one race highly predictive of voting in others in same election
- i's presidential turnout isn't independent of i's Senate/House turnout

How do we correct for this?

- For each observation i, identify:
  - → Which observations' outcomes may be partly predicted by knowing Y_i
  - → Which observations (hopefully many more) are not predicted by Y_i
- Instruct our statistical software to take these clusters into account
  - → Will generally result in larger standard errors and p-values
  - → Greater within-cluster correlations ⇝ bigger corrections needed

Clusters in the turnout study

Potential independence violation in GGL: voters in same household

One household member votes ⇝ Others likelier-than-average to vote

```
## Within actual households
df_ggl |>
  filter(hh_size == 2, treatment == "Control") |>
  group_by(hh_id) |>
  summarize(n_votes = sum(y)) |>
  count(n_votes) |>
  mutate(prop = n / sum(n))
```

```
# A tibble: 3 × 3
  n_votes     n  prop
    <dbl> <int> <dbl>
1       0 36415 0.612
2       1 10137 0.170
3       2 12959 0.218
```

```
## Shuffling voters into fake households
df_ggl |>
  filter(hh_size == 2, treatment == "Control") |>
  mutate(fake_hh_id = sample(hh_id)) |>
  group_by(fake_hh_id) |>
  summarize(n_votes = sum(y)) |>
  count(n_votes) |>
  mutate(prop = n / sum(n))
```

```
# A tibble: 3 × 3
  n_votes     n   prop
    <dbl> <int>  <dbl>
1       0 28872 0.485
2       1 25223 0.424
3       2  5416 0.0910
```

Correcting standard errors for clustering

- Install + load the `fixest` package
- Use `feols()` in place of `lm()` ▶ Original results
- All else same as before, now just accounting for non-independence

```
library("fixest")
fit_all_clus <- feols(y ~ treatment,
                      vcov = ~ hh_id,
                      data = df_ggl)
tidy(fit_all_clus)
```

```
# A tibble: 5 × 5
  term                estimate std.error statistic  p.value
  <chr>                  <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)           0.322     0.00298    108.   0
2 treatmentCivic Duty -0.00784   0.00420     -1.86 6.22e- 2
3 treatmentControl    -0.0257    0.00326     -7.90 2.81e-15
4 treatmentNeighbors   0.0556    0.00431     12.9  4.25e-38
5 treatmentSelf        0.0228    0.00425      5.36 8.45e- 8
```

Wrapping up

What we did today

1. Influences on standard errors of the treatment effect estimate
    - More unpredictable outcome $\rightsquigarrow$ Higher SEs
    - More variation in treatment values $\rightsquigarrow$ Lower SEs
    - Larger sample size $\rightsquigarrow$ Lower SEs
2. Other important inferential statistics
    - Confidence interval: Will contain population value in specified proportion of samples
    - p-value: Chance of seeing this big an effect in sample if none in population
3. Calculating inferential statistics in R
    - `tidy()` to extract table of estimates, SEs, p-values
    - Confidence interval: estimate $\pm$ 2se
    - `feols()` when there is clustering of observations

To do for next week

Next week's topic — Matching

1. Read research paper "MPs for Sale?" by Eggers and Hainmueller
2. Read *Mastering 'Metrics*, chapter 2, pages 47–55
3. Read "Matching Methods for Causal Inference" by Stuart
4. Work on Problem Set 2 due Tues 2/11