Metrics Review

EC 421, Set 2

Edward Rubin
10 January 2019

Prologue

R showcase

New this week

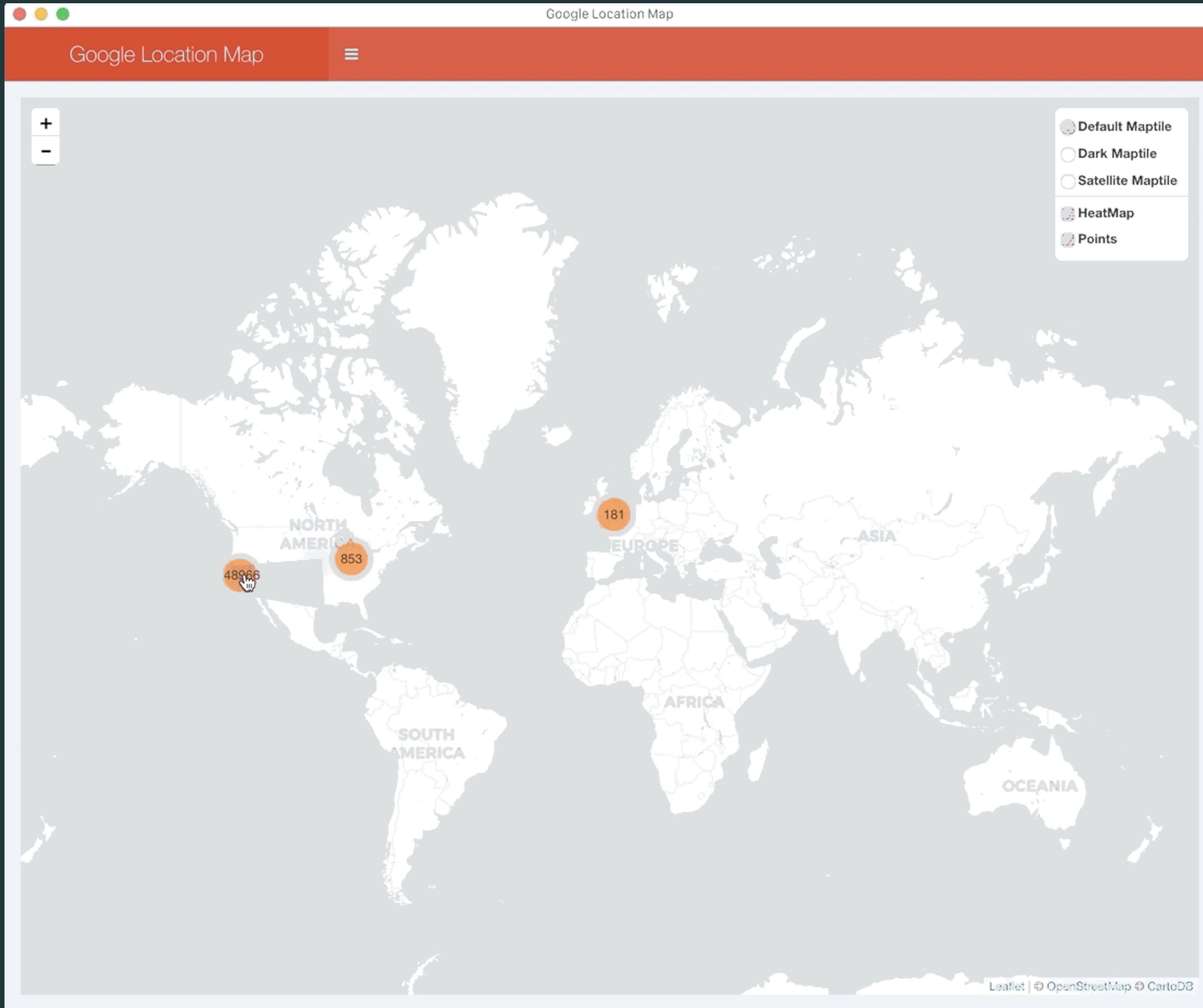
Because part of this course is about learning and implementing R, I'm going to share some interesting/amazing/fun applications of R.

Culture of Insight website

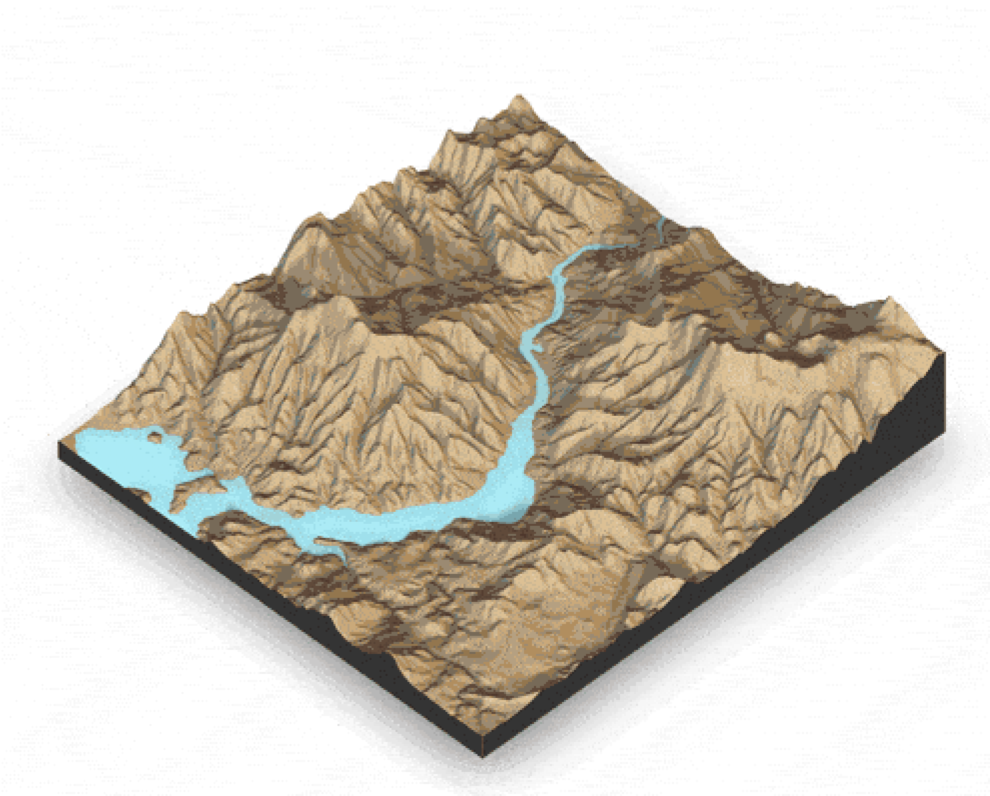
- R-based web application
- Maps your location data (as tracked by Google)
- Great example of R's ability to extend beyond traditional statistical programming
- (Visualization matters.)

The `rayshader` package

- Creates really cool shaded maps (easily!)
- What else does one need?



Google Location Map
Google Location Map
Default Maptile
Dark Maptile
Satellite Maptile
HeatMap
Points
NORTH AMERICA
EUROPE
ASIA
AFRICA
SOUTH AMERICA
OCEANIA
48966
853
181
Leaflet | © OpenStreetMap © CartoDB



The `rayshader` package.

Last Time

Follow Up

`R` is available at **all academic workstations at UO**.

Last Time

Follow Up

R is available at **all academic workstations at UO**.

Motivation

In our last set of slides, we

1. discussed the **motivation** for studying econometrics (metrics)
2. **introduced R**—why we use it, what it can do
3. **started reviewing** material from your previous classes

These notes continue the review, building the foundation for some new topics (soon).

Review

Population *vs.* sample

Models and notation

We write our (simple) population model

$$y_i = \beta_0 + \beta_1 x_i + u_i$$

and our sample-based estimated regression model as

$$y_i = \hat{\beta}_0 + \hat{\beta}_1 x_i + e_i$$

An estimated regession model produces estimates for each observation:

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$$

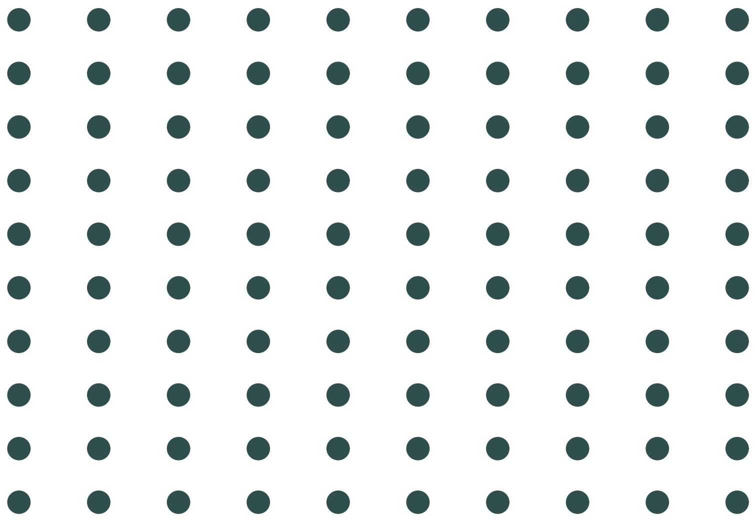
which gives us the *best-fit* line through our dataset.

Population *vs.* sample

Question: Why are we so worked up about the distinction between *population* and *sample*?

Population *vs.* sample

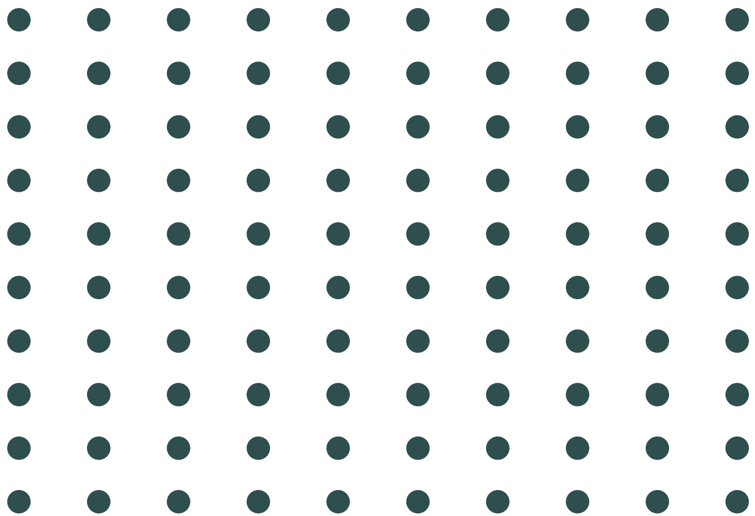
Question: Why are we so worked up about the distinction between *population* and *sample*?



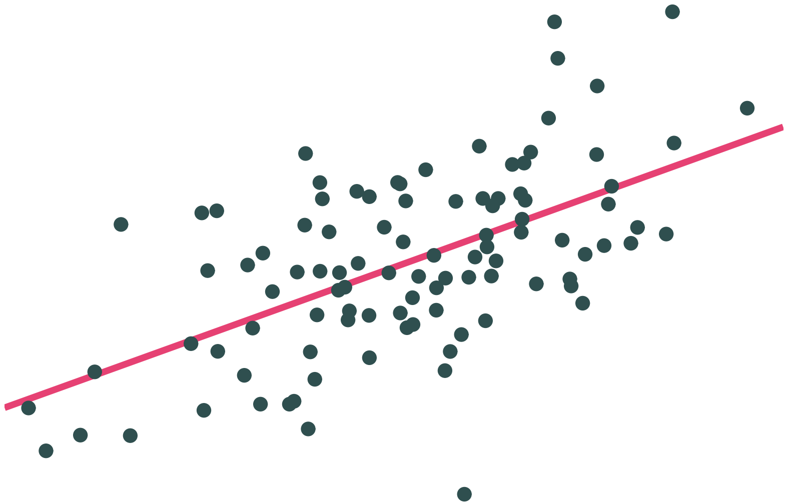
Population

Population *vs.* sample

Question: Why are we so worked up about the distinction between *population* and *sample*?



Population



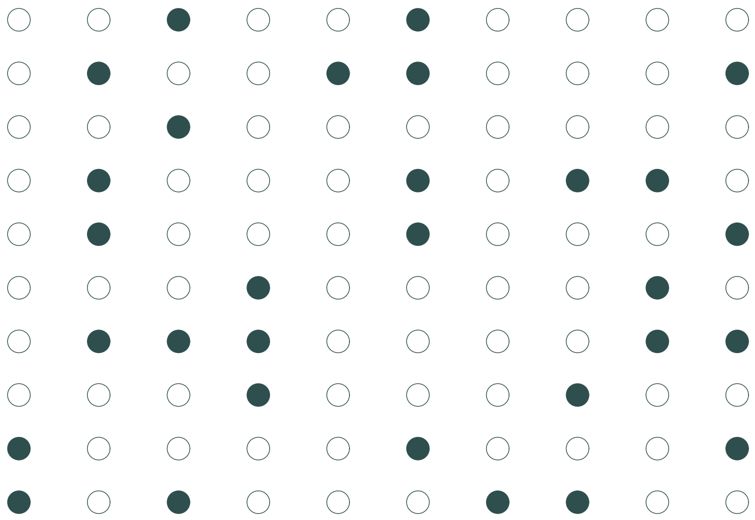
Population relationship

$$y_i = 2.53 + 0.57 x_i + u_i$$

$$y_i = \beta_0 + \beta_1 x_i + u_i$$

Population *vs.* sample

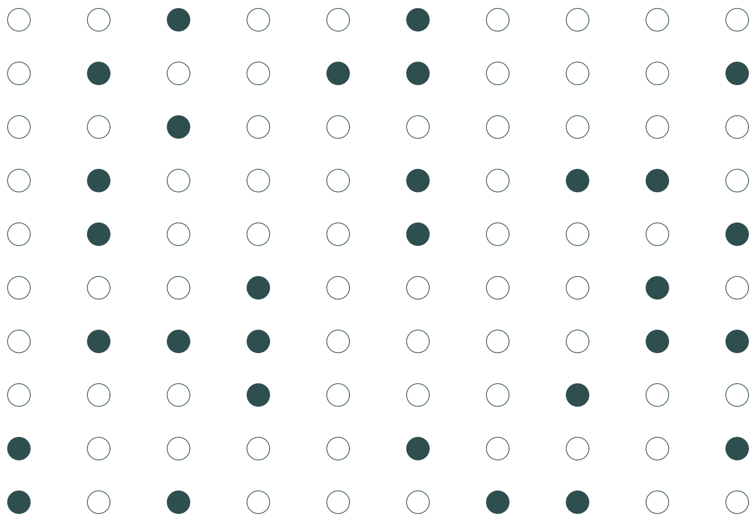
Question: Why are we so worked up about the distinction between *population* and *sample*?



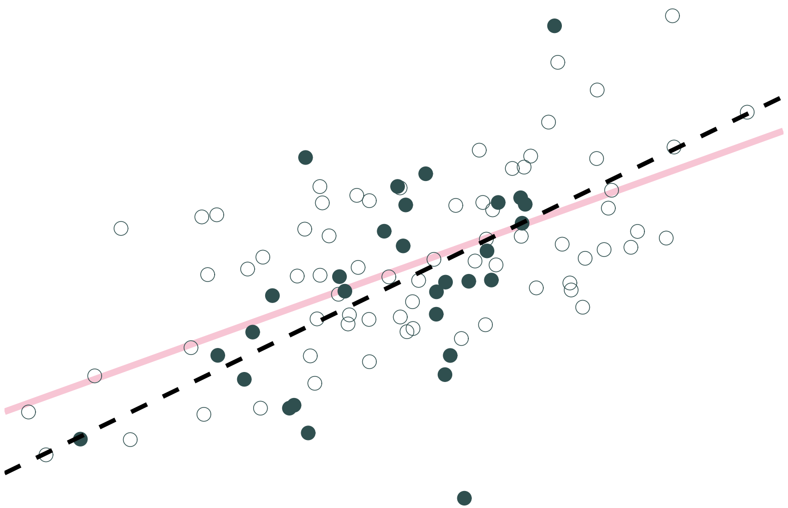
Sample 1: 30 random individuals

Population *vs.* sample

Question: Why are we so worked up about the distinction between *population* and *sample*?



Sample 1: 30 random individuals



Population relationship

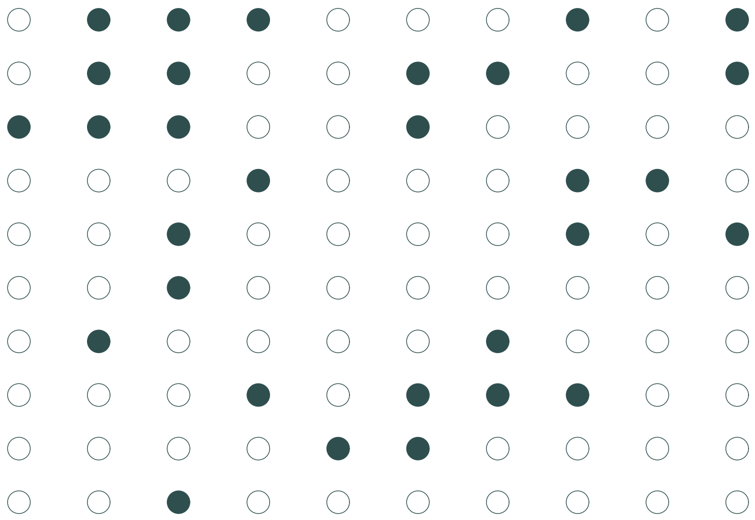
$$y_i = 2.53 + 0.57 x_i + u_i$$

Sample relationship

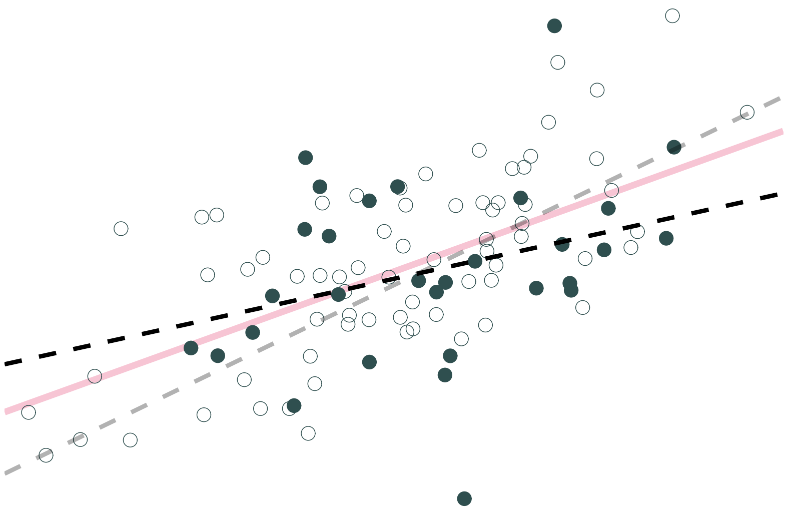
$$\hat{y}_i = 1.36 + 0.76 x_i$$

Population *vs.* sample

Question: Why are we so worked up about the distinction between *population* and *sample*?



Sample 2: 30 random individuals



Population relationship

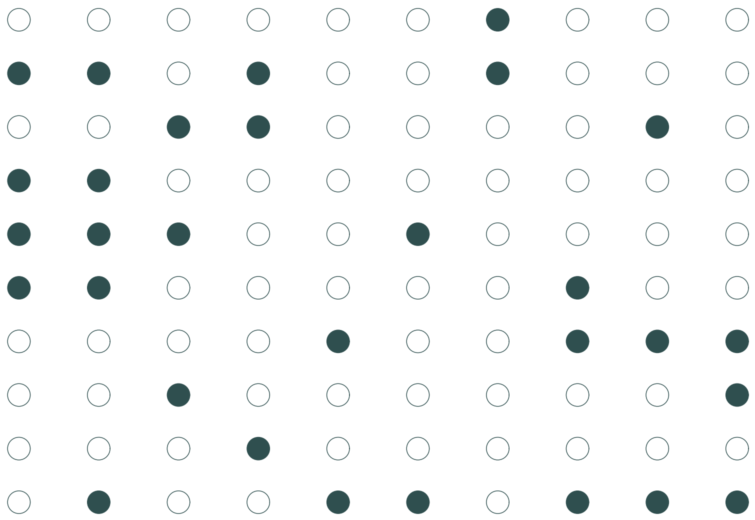
$$y_i = 2.53 + 0.57 x_i + u_i$$

Sample relationship

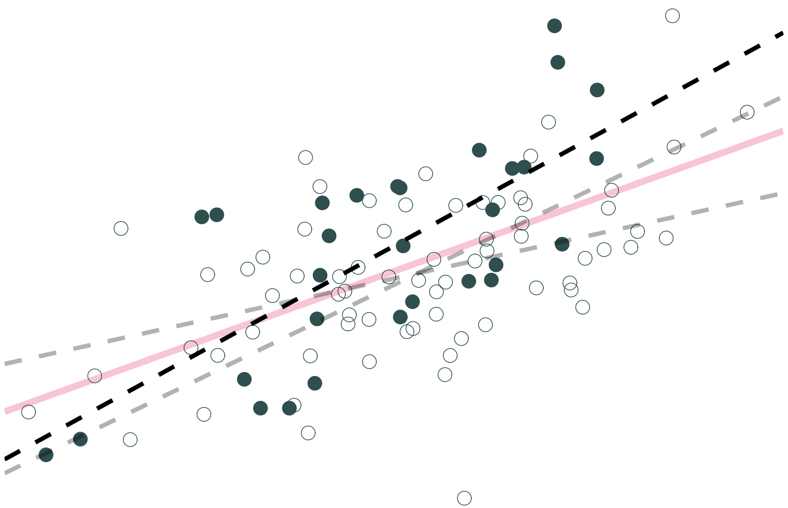
$$\hat{y}_i = 3.53 + 0.34 x_i$$

Population *vs.* sample

Question: Why are we so worked up about the distinction between *population* and *sample*?



Sample 3: 30 random individuals



Population relationship

$$y_i = 2.53 + 0.57 x_i + u_i$$

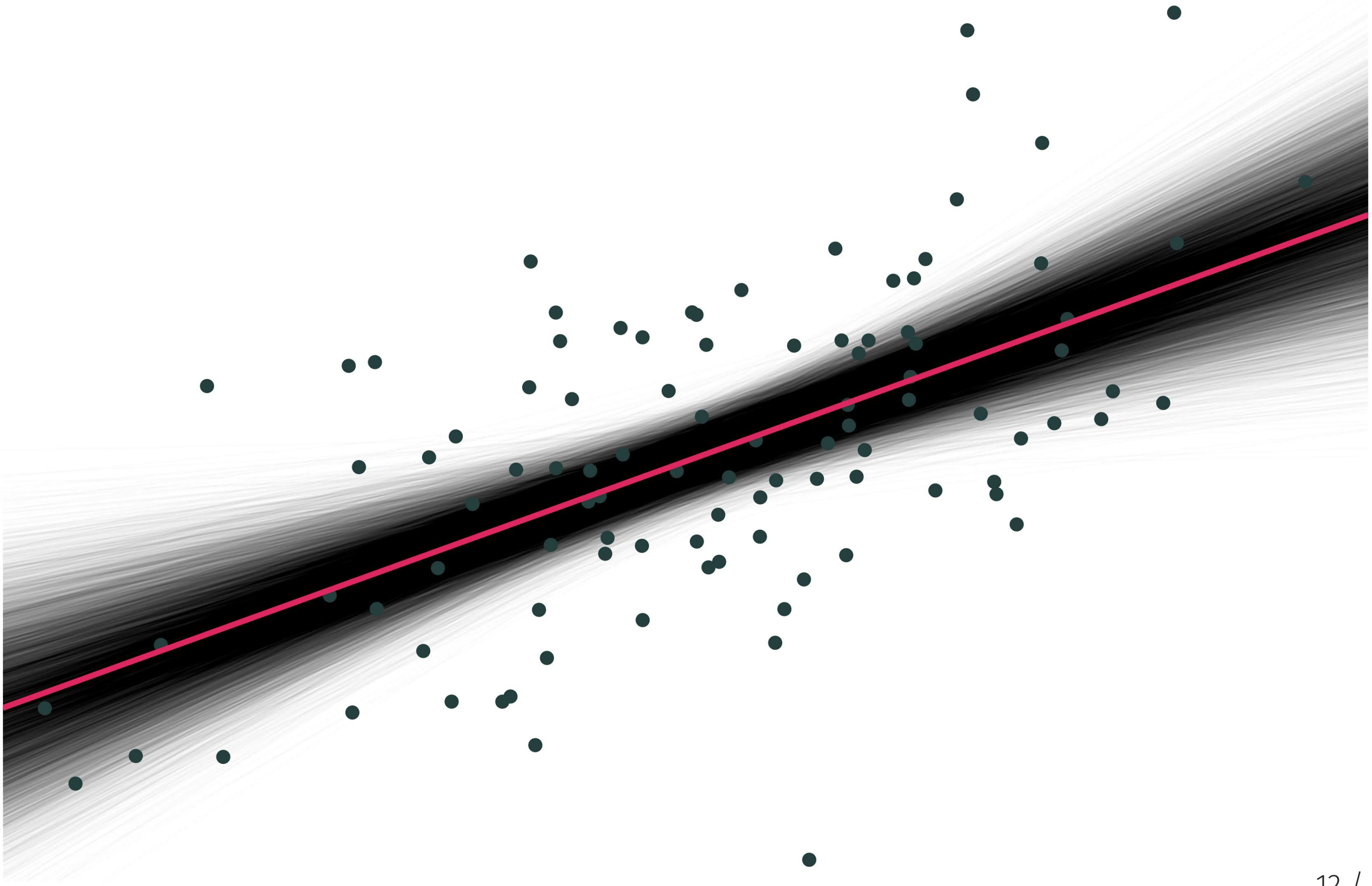
Sample relationship

$$\hat{y}_i = 1.44 + 0.86 x_i$$

Let's repeat this **10,000 times**.

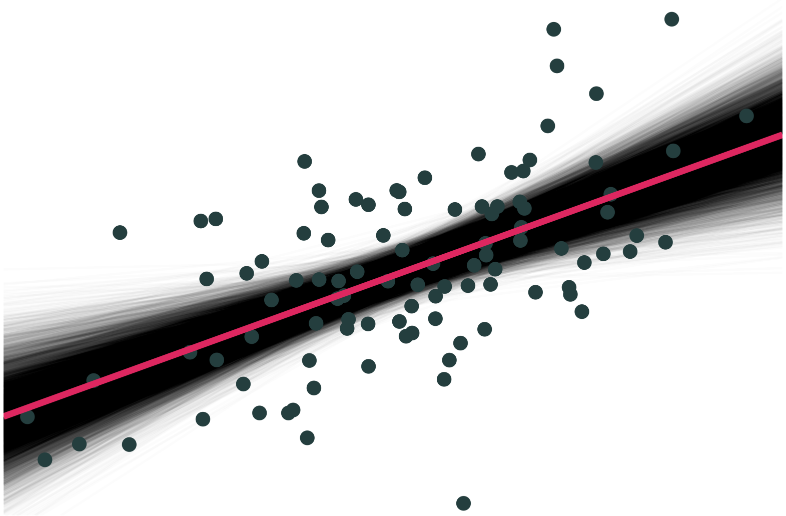
(This exercise is called a (Monte Carlo) simulation.)

Population *vs.* sample



Population *vs.* sample

Question: Why are we so worked up about the distinction between *population* and *sample*?

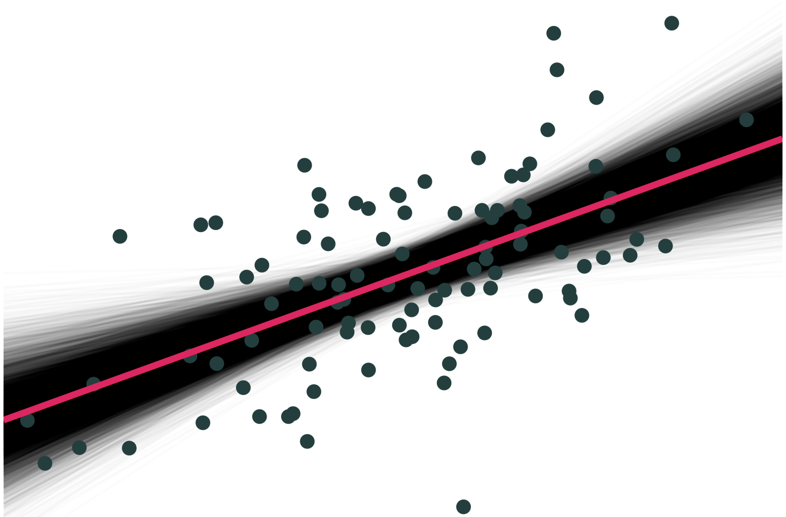


What do you notice?

- On **average**, our regression lines match the population line very nicely.
- However, **individual lines** (samples) can really miss the mark.
- Differences between individual samples and the population lead to **uncertainty** for the econometrician.

Population *vs.* sample

Question: Why are we so worked up about the distinction between *population* and *sample*?



What do you notice?

- On **average**, our regression lines match the population line very nicely.
- However, **individual lines** (samples) can really miss the mark.
- Differences between individual samples and the population lead to **uncertainty** for the econometrician.

Answer: Uncertainty matters.

$\hat{\beta}$ itself is a random variable—dependent upon the random sample. When we take a sample and run a regression, we don't know if it's a 'good' sample ($\hat{\beta}$ is close to β) or a 'bad sample' (our sample differs greatly from the population).

Population *vs.* sample

Uncertainty

Keeping track of this uncertainty will be a key concept throughout our class.

- Estimating standard errors for our estimates.
- Testing hypotheses.
- Correcting for heteroskedasticity and autocorrelation.

Population *vs.* sample

Uncertainty

Keeping track of this uncertainty will be a key concept throughout our class.

- Estimating standard errors for our estimates.
- Testing hypotheses.
- Correcting for heteroskedasticity and autocorrelation.

But first, let's remind ourselves of how we get these (uncertain) regression estimates.

Linear regression

The estimator

We can estimate a regression line in `R` (`lm(y ~ x, my_data)`) and `Stata` (`reg y x`). But where do these estimates come from?

A few slides back:

> $$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$$
>
> which gives us the *best-fit* line through our dataset.

But what do we mean by "best-fit line"?

Being the "best"

Question: What do we mean by *best-fit line*?

Answers:

- In general,[†] *best-fit line* means the line that minimizes the sum of squared errors (SSE):

$$\text{SSE} = \sum_{i=1}^{n} e_i^2$$

where

$$e_i = y_i - \hat{y}_i$$

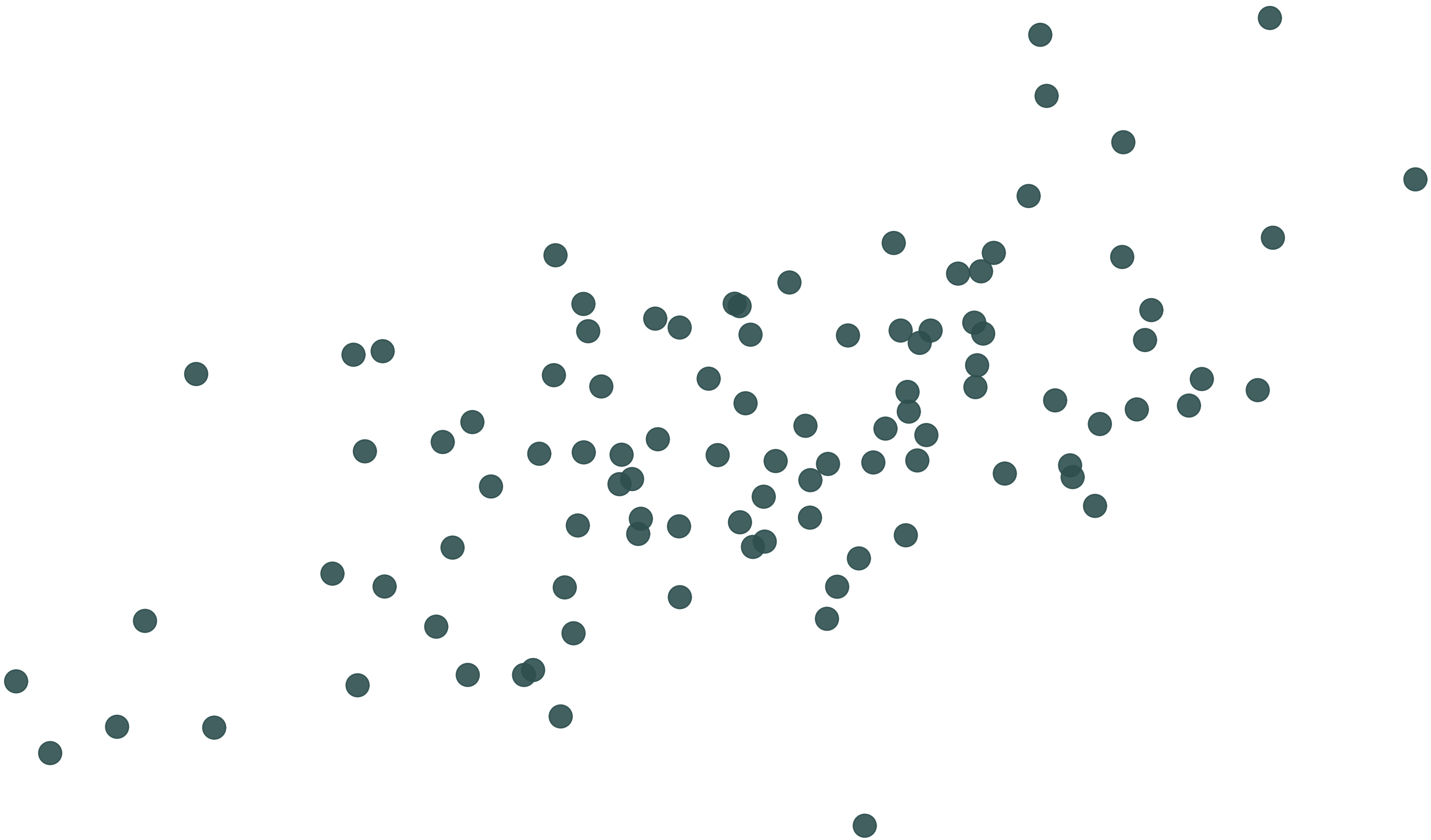
- Ordinary **least squares** (**OLS**) minimizes the sum of the squared errors.
- Based upon a set of (mostly palatable) assumptions, OLS
  - Is unbiased (and consistent)
  - Is the *best*[††] linear unbiased estimator (BLUE)

[†]: *In general* here means *generally in econometrics*. It's possible to have other definitions (common in machine learning).

[††]: In the case of BLUE, *best* means minimum variance.

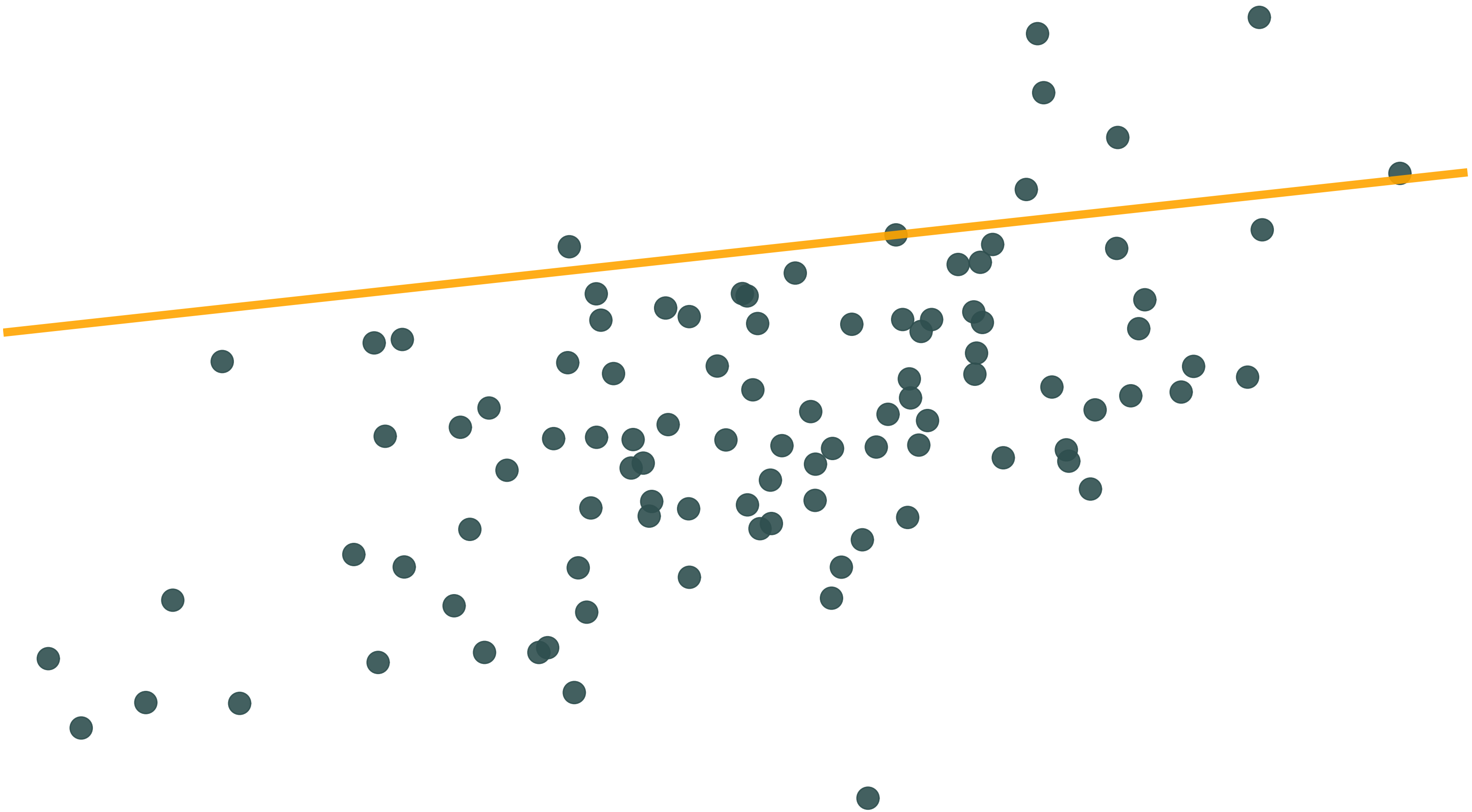
OLS *vs.* other lines/estimators

Let's consider the dataset we previously generated.



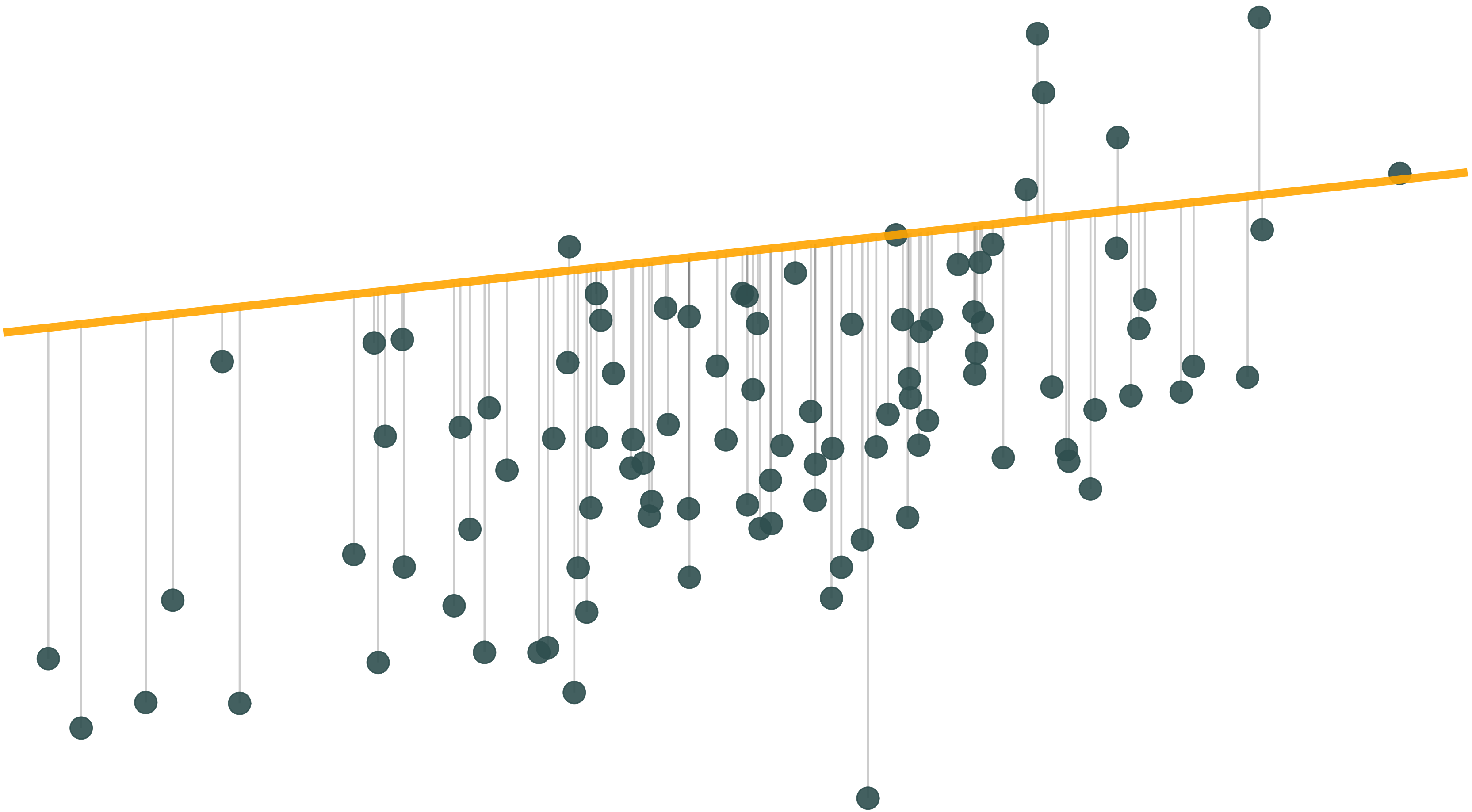
OLS *vs.* other lines/estimators

For any line (*i.e.*, $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$) that we draw



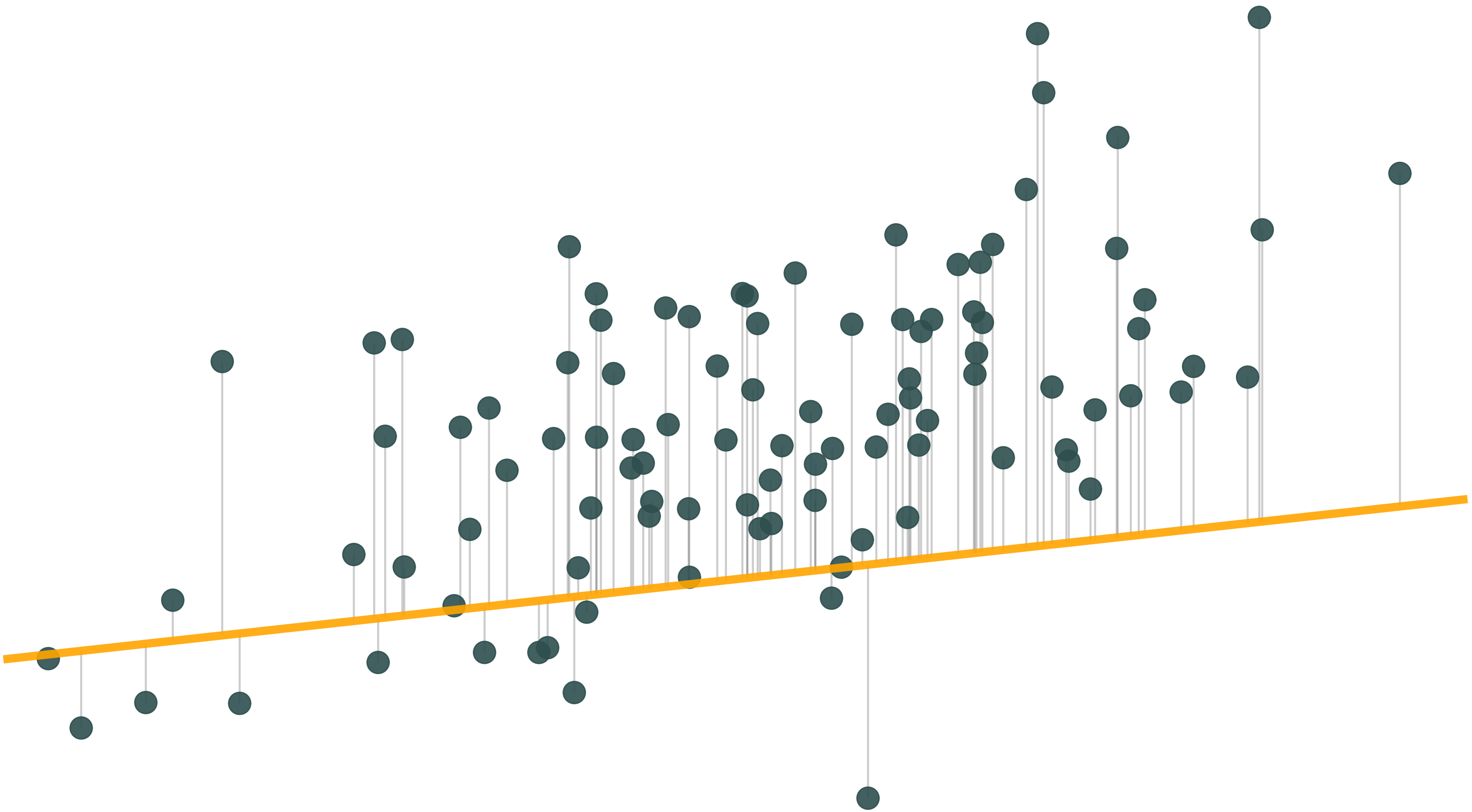
OLS *vs.* other lines/estimators

For any line (*i.e.*, $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$) that we draw, we can calculate the errors: $e_i = y_i - \hat{y}_i$



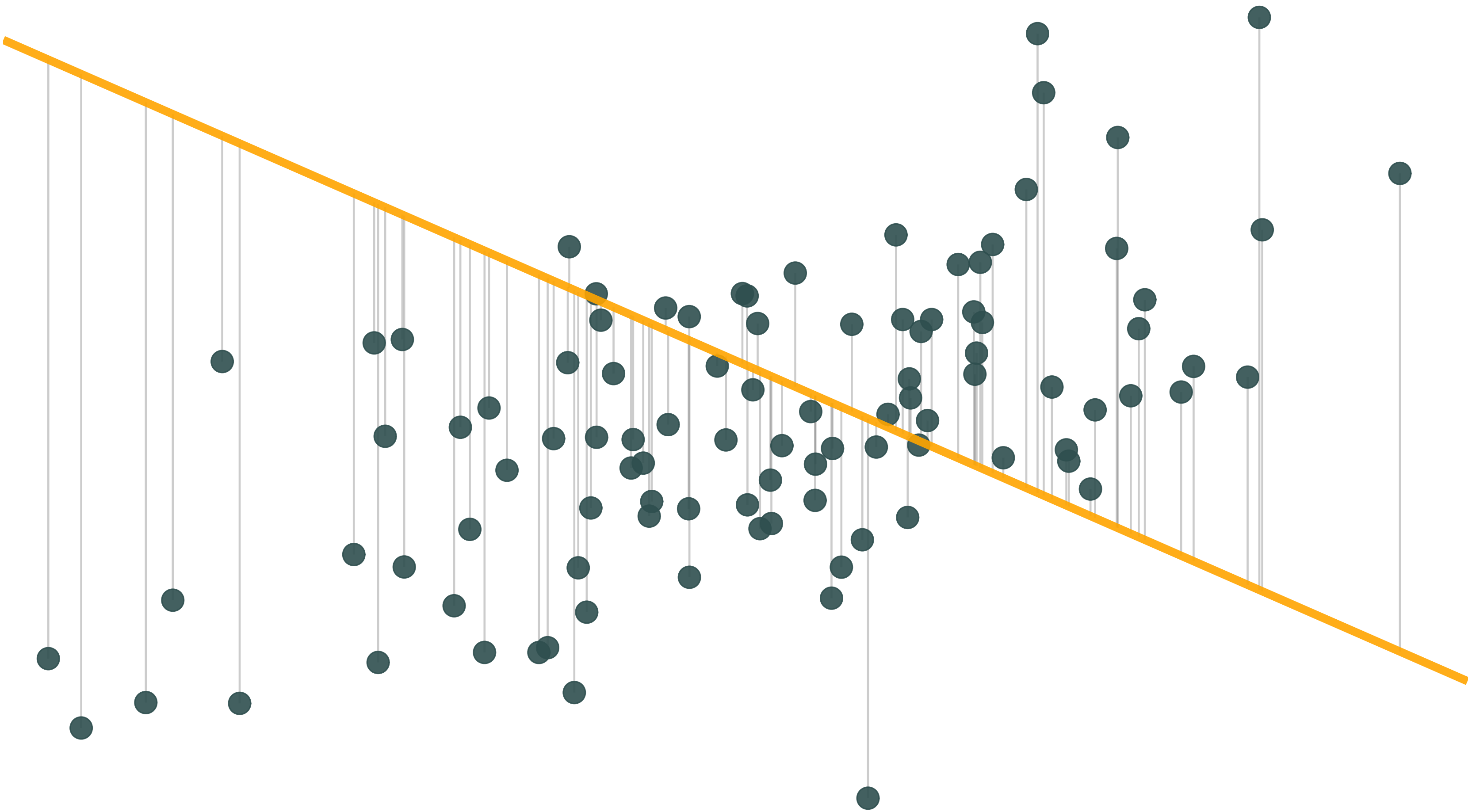
OLS *vs.* other lines/estimators

For any line (*i.e.*, $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$) that we draw, we can calculate the errors: $e_i = y_i - \hat{y}_i$



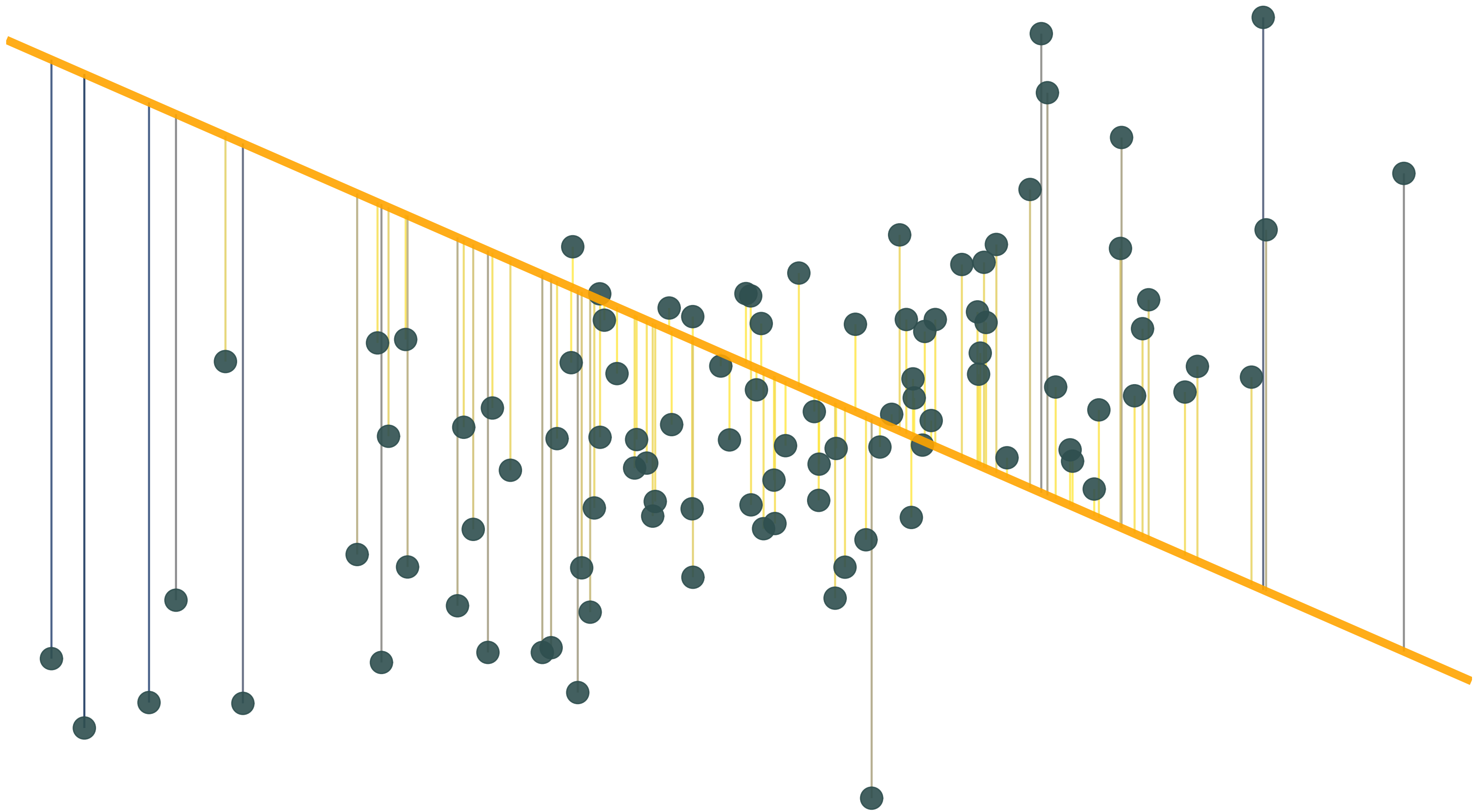
OLS *vs.* other lines/estimators

For any line (*i.e.*, $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$) that we draw, we can calculate the errors: $e_i = y_i - \hat{y}_i$



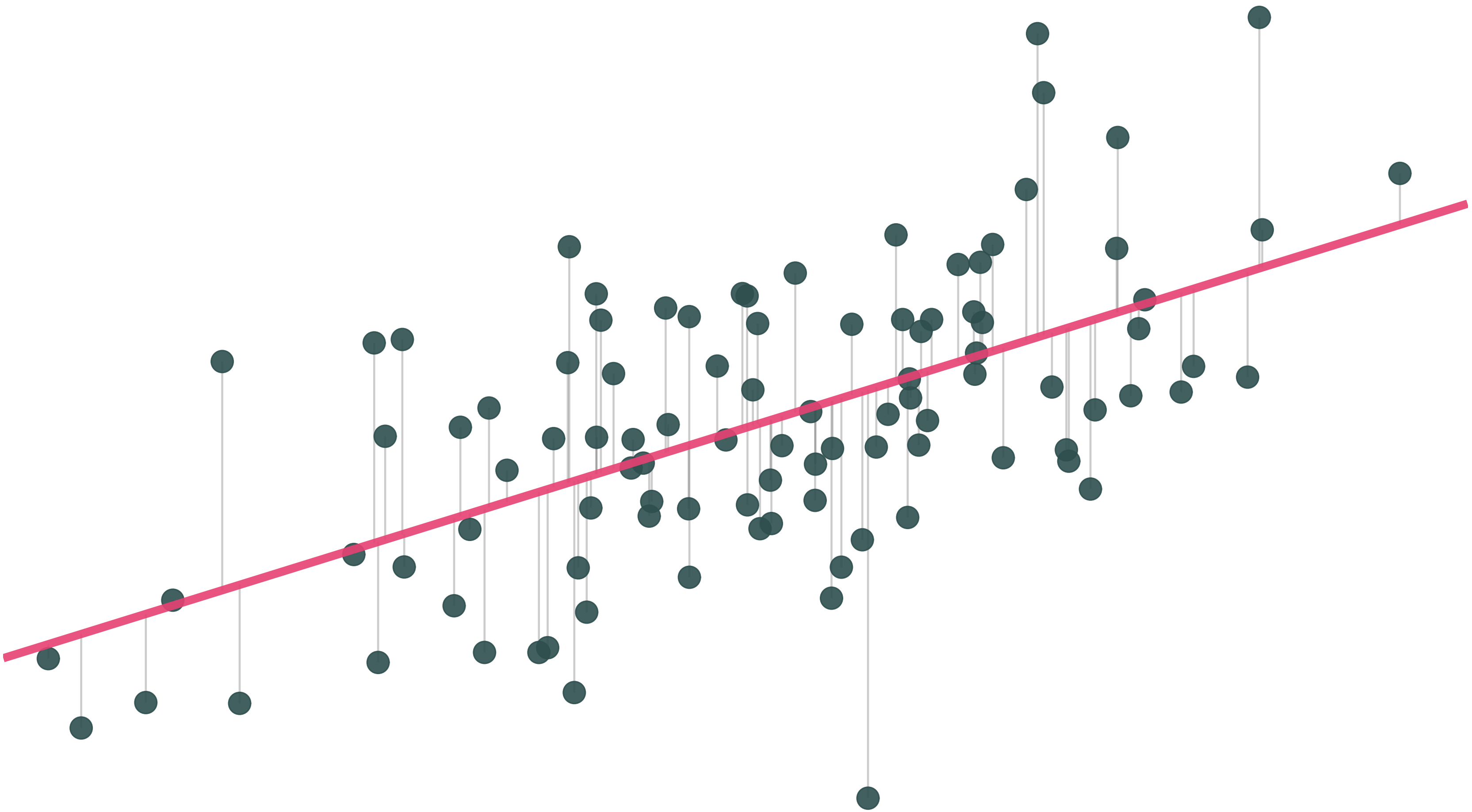
OLS *vs.* other lines/estimators

Because SSE squares the errors (*i.e.*, $\sum e_i^2$), big errors are penalized more than small ones.



OLS *vs.* other lines/estimators

The OLS estimate is the combination of $\hat{\beta}_0$ and $\hat{\beta}_1$ that minimize SSE.



OLS

Formally

In simple linear regression, the OLS estimator comes from choosing the $\hat{\beta}_0$ and $\hat{\beta}_1$ that minimize the sum of squared errors (SSE), *i.e.*,

$$\min_{\hat{\beta}_0,\, \hat{\beta}_1} \text{SSE}$$

Formally

In simple linear regression, the OLS estimator comes from choosing the $\hat{\beta}_0$ and $\hat{\beta}_1$ that minimize the sum of squared errors (SSE), *i.e.*,

$$\min_{\hat{\beta}_0,\, \hat{\beta}_1} \mathrm{SSE}$$

But we already know

$$\mathrm{SSE} = \sum_i e_i^2$$

Now we use the definitions of e_i and $\hat{y}$ (plus and some algebra)

$$\begin{aligned} e_i^2 &= (y_i - \hat{y}_i)^2 = \left(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i\right)^2 \\ &= y_i^2 - 2y_i\hat{\beta}_0 - 2y_i\hat{\beta}_1 x_i + \hat{\beta}_0^2 + 2\hat{\beta}_0\hat{\beta}_1 x_i + \hat{\beta}_1^2 x_i^2 \end{aligned}$$

Formally

In simple linear regression, the OLS estimator comes from choosing the $\hat{\beta}_0$ and $\hat{\beta}_1$ that minimize the sum of squared errors (SSE), *i.e.*,

$$\min_{\hat{\beta}_0,\, \hat{\beta}_1} \text{SSE}$$

But we already know

$$\text{SSE} = \sum_i e_i^2$$

Now we use the definitions of e_i and $\hat{y}$ (plus and some algebra)

$$\begin{aligned} e_i^2 &= \left(y_i - \hat{y}_i \right)^2 = \left(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i \right)^2 \\ &= y_i^2 - 2 y_i \hat{\beta}_0 - 2 y_i \hat{\beta}_1 x_i + \hat{\beta}_0^2 + 2 \hat{\beta}_0 \hat{\beta}_1 x_i + \hat{\beta}_1^2 x_i^2 \end{aligned}$$

Recall: Minimizing a multivariate function requires (**1**) first derivatives equal zero (the *first-order conditions*) and (**2**) second-order conditions on concavity.

OLS

Formally

We're getting close. We need to **minimize SSE**, and we've just shown how SSE relates to our sample (our data, *i.e.*, x and y) and our estimates (*i.e.*, $\hat{\beta}_0$ and $\hat{\beta}_1$).

$$\text{SSE} = \sum_i e_i^2 = \sum_i \left(y_i^2 - 2 y_i \hat{\beta}_0 - 2 y_i \hat{\beta}_1 x_i + \hat{\beta}_0^2 + 2 \hat{\beta}_0 \hat{\beta}_1 x_i + \hat{\beta}_1^2 x_i^2 \right)$$

For the first-order conditions of minimization, we now take the first derivates[†] of SSE with respect to $\hat{\beta}_0$ and $\hat{\beta}_1$.

$$\begin{aligned} \frac{\partial \text{SSE}}{\partial \hat{\beta}_0} &= \sum_i \left(2\hat{\beta}_0 + 2\hat{\beta}_1 x_i - 2 y_i \right) = 2n\hat{\beta}_0 + 2\hat{\beta}_1 \sum_i x_i - 2 \sum_i y_i \\ &= 2n\hat{\beta}_0 + 2n\hat{\beta}_1 \overline{x} - 2n\overline{y} \end{aligned}$$

where $\overline{x} = \frac{\sum x_i}{n}$ and $\overline{y} = \frac{\sum y_i}{n}$ give the sample means of x and y (sample size n).

[†]: I'll leave the second-order conditions for you...

OLS

Formally

The first-order conditions state that the derivatives are equal to zero, so:

$$\frac{\partial \text{SSE}}{\partial \hat{\beta}_0} = 2n\hat{\beta}_0 + 2n\hat{\beta}_1\overline{x} - 2n\overline{y} = 0$$

which implies

$$\hat{\beta}_0 = \overline{y} - \hat{\beta}_1\overline{x}$$

Now for $\hat{\beta}_1$.

OLS

Formally

Take the derivative of SSE with respect to $\hat{\beta}_1$

$$\begin{aligned}\frac{\partial \text{SSE}}{\partial \hat{\beta}_1} &= \sum_i \left(2\hat{\beta}_0 x_i + 2\hat{\beta}_1 x_i^2 - 2y_i x_i\right) = 2\hat{\beta}_0 \sum_i x_i + 2\hat{\beta}_1 \sum_i x_i^2 - 2\sum_i y_i x_i \\ &= 2n\hat{\beta}_0 \overline{x} + 2\hat{\beta}_1 \sum_i x_i^2 - 2\sum_i y_i x_i\end{aligned}$$

set it equal to zero (first-order conditions, again)

$$\frac{\partial \text{SSE}}{\partial \hat{\beta}_1} = 2n\hat{\beta}_0 \overline{x} + 2\hat{\beta}_1 \sum_i x_i^2 - 2\sum_i y_i x_i = 0$$

and substitute in our relationship for $\hat{\beta}_0$, *i.e.*, $\hat{\beta}_0 = \overline{y} - \hat{\beta}_1 \overline{x}$. Thus,

$$2n\left(\overline{y} - \hat{\beta}_1 \overline{x}\right)\overline{x} + 2\hat{\beta}_1 \sum_i x_i^2 - 2\sum_i y_i x_i = 0$$

Formally

Continuing from the last slide

$$2n\left(\overline{y} - \hat{\beta}_1\overline{x}\right)\overline{x} + 2\hat{\beta}_1\sum_i x_i^2 - 2\sum_i y_i x_i = 0$$

we multiply out

$$2n\overline{y}\,\overline{x} - 2n\hat{\beta}_1\overline{x}^2 + 2\hat{\beta}_1\sum_i x_i^2 - 2\sum_i y_i x_i = 0$$

$$\implies 2\hat{\beta}_1\left(\sum_i x_i^2 - n\overline{x}^2\right) = 2\sum_i y_i x_i - 2n\overline{y}\,\overline{x}$$

$$\implies \hat{\beta}_1 = \frac{\sum_i y_i x_i - 2n\overline{y}\,\overline{x}}{\sum_i x_i^2 - n\overline{x}^2} = \frac{\sum_i (x_i - \overline{x})(y_i - \overline{y})}{\sum_i (x_i - \overline{x})^2}$$

OLS

Formally

Done!

We now have (lovely) OLS estimators for the slope

$$\hat{\beta}_1 = \frac{\sum_i (x_i - \overline{x})(y_i - \overline{y})}{\sum_i (x_i - \overline{x})^2}$$

and the intercept

$$\hat{\beta}_0 = \overline{y} - \hat{\beta}_1 \overline{x}$$

Plus, you know what the *least squares* part of ordinary least squares means. ☺

OLS

Formally

Done!

We now have (lovely) OLS estimators for the slope

$$\hat{\beta}_1 = \frac{\sum_i (x_i - \overline{x})(y_i - \overline{y})}{\sum_i (x_i - \overline{x})^2}$$

and the intercept

$$\hat{\beta}_0 = \overline{y} - \hat{\beta}_1 \overline{x}$$

Plus, you know what the *least squares* part of ordinary least squares means. ☺

We now turn to the assumptions and (implied) properties of OLS.

OLS: Assumptions and properties

OLS: Assumptions and properties

Properties

Question: What properties might we care about for an estimator?

Properties

Question: What properties might we care about for an estimator?

Refresher: Density functions

Recall that we use **probability density functions** (PDFs) to describe the probability a **continuous random variable** takes on a range of values. (The total area = 1.)

These PDFs characterize probability distributions, and the most common/famous/popular distributions get names (*e.g.*, normal, *t*, Gamma).

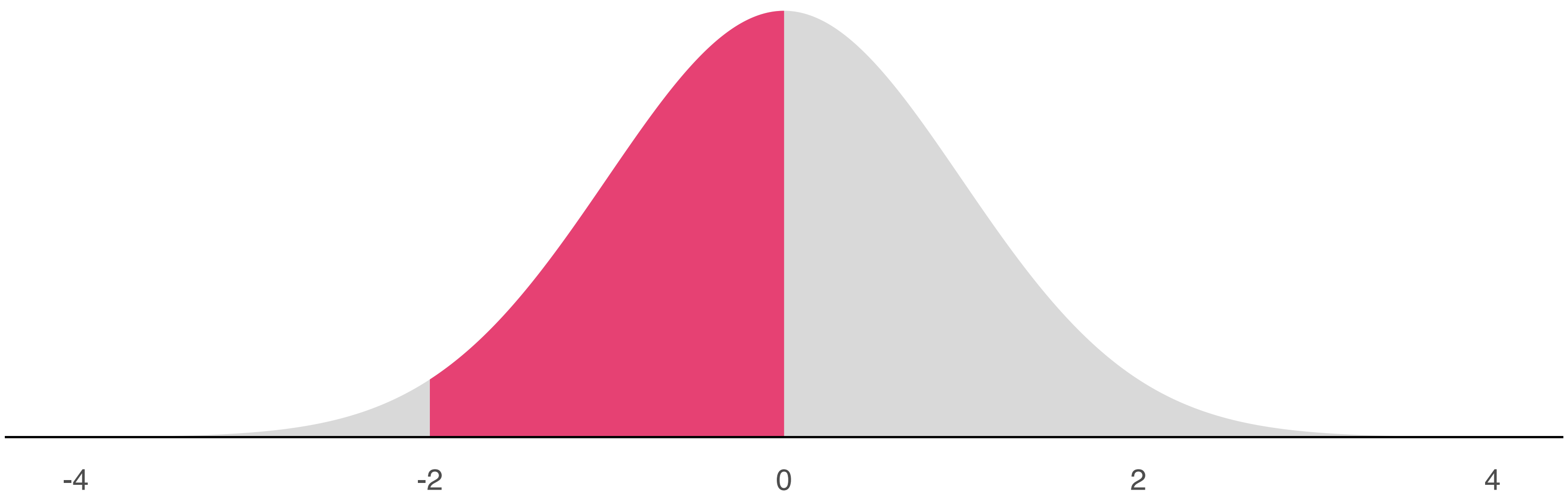
OLS: Assumptions and properties

Properties

Question: What properties might we care about for an estimator?

Refresher: Density functions

The probability a standard normal random variable takes on a value between -2 and 0: $\mathrm{P}(-2 \leq X \leq 0) = 0.48$

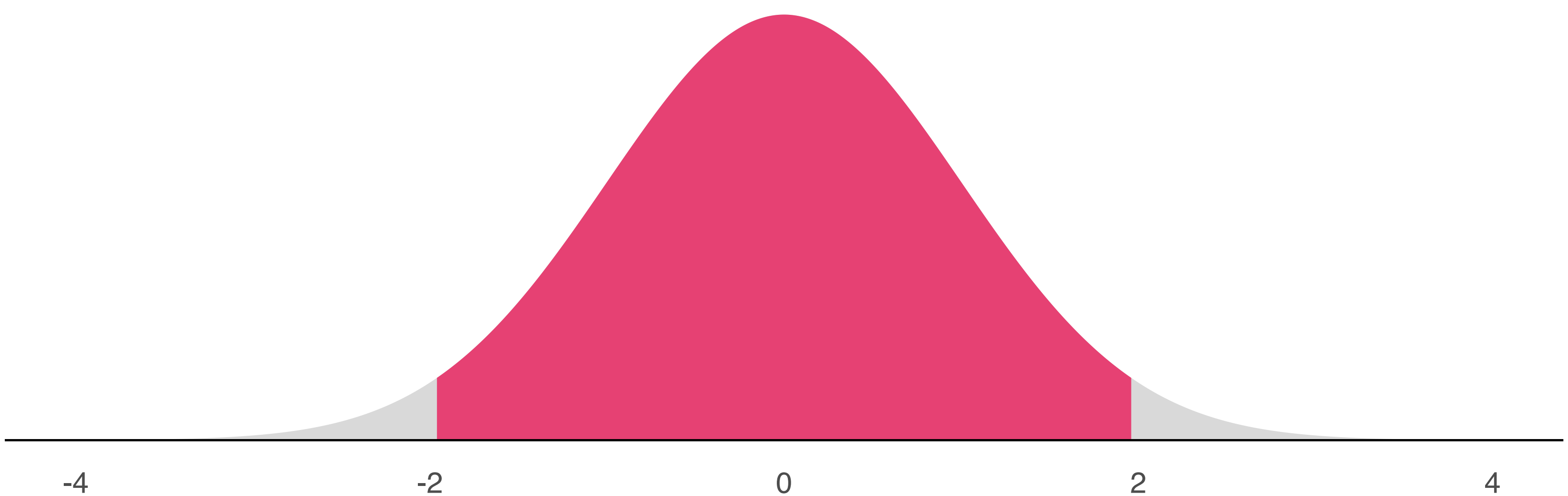


Properties

Question: What properties might we care about for an estimator?

Refresher: Density functions

The probability a standard normal random variable takes on a value between -1.96 and 1.96:
$\mathrm{P}(-1.96 \leq X \leq 1.96) = 0.95$



OLS: Assumptions and properties

Properties

Question: What properties might we care about for an estimator?

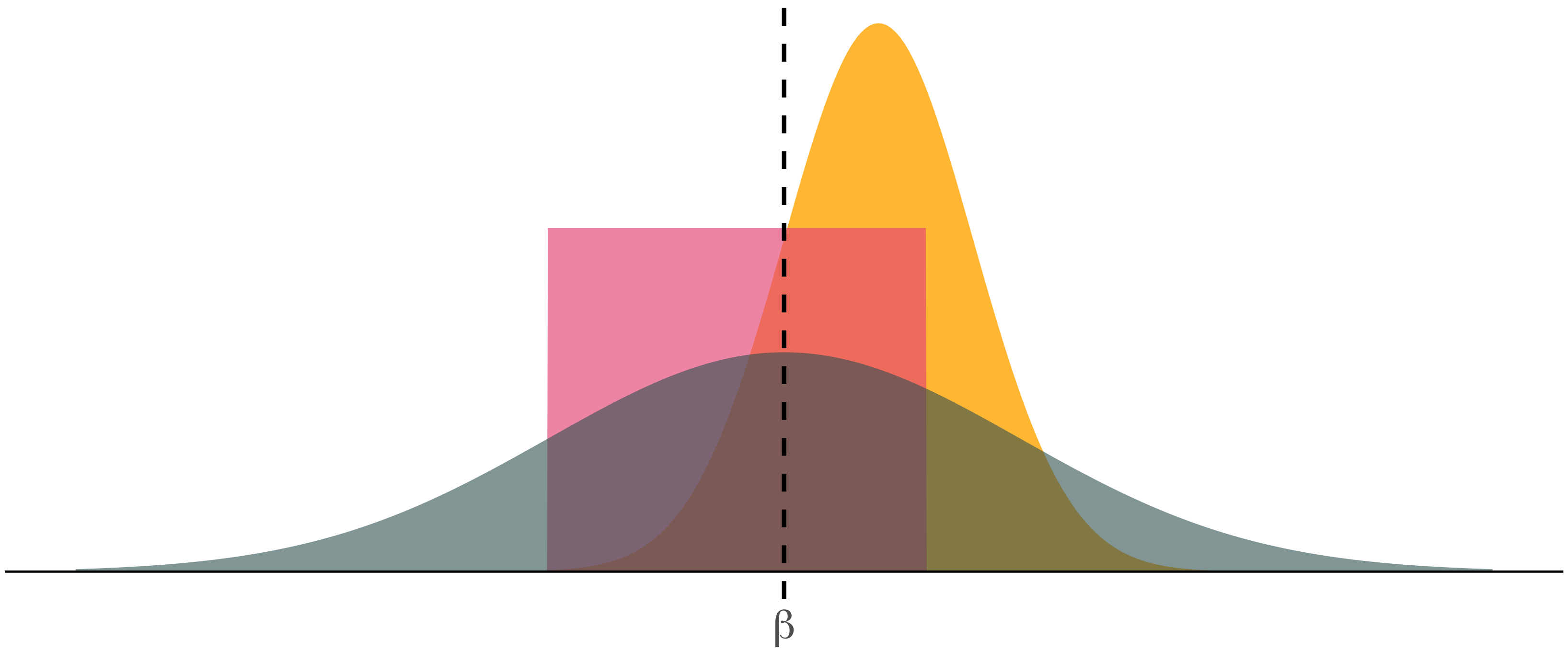
Refresher: Density functions

The probability a standard normal random variable takes on a value beyond 2:
$\mathrm{P}(X > 2) = 0.023$

Properties

Question: What properties might we care about for an estimator?

Imagine we are trying to estimate an unknown parameter β, and we know the distributions of three competing estimators. Which one would we want? How would we decide?



OLS: Assumptions and properties

Properties

Question: What properties might we care about for an estimator?

Answer one: Bias.

On average (after *many* samples), does the estimator tend toward the correct value?

More formally: Does the mean of estimator's distribution equal the parameter it estimates?

$$\mathop{\text{Bias}}_{\beta}\left(\hat{\beta}\right) = \boldsymbol{E}\left[\hat{\beta}\right] - \beta$$

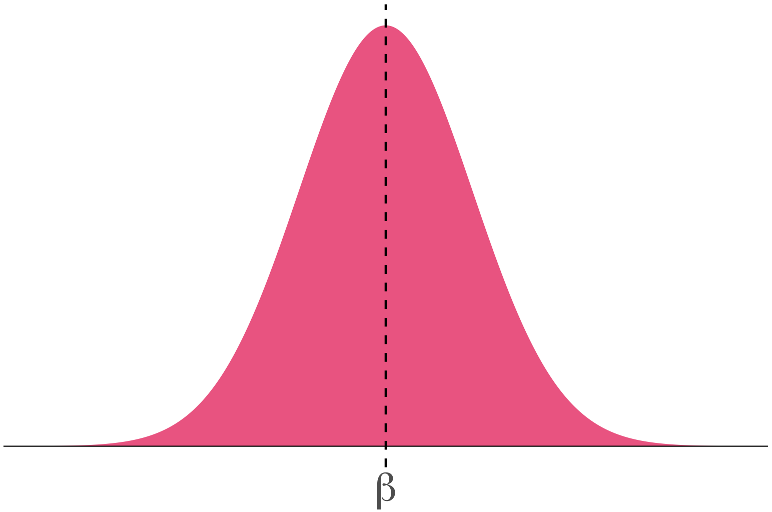
OLS: Assumptions and properties

Properties

Question: What properties might we care about for an estimator?

Answer one: Bias.

Unbiased estimator: $\boldsymbol{E}\left[\hat{\beta}\right] = \beta$



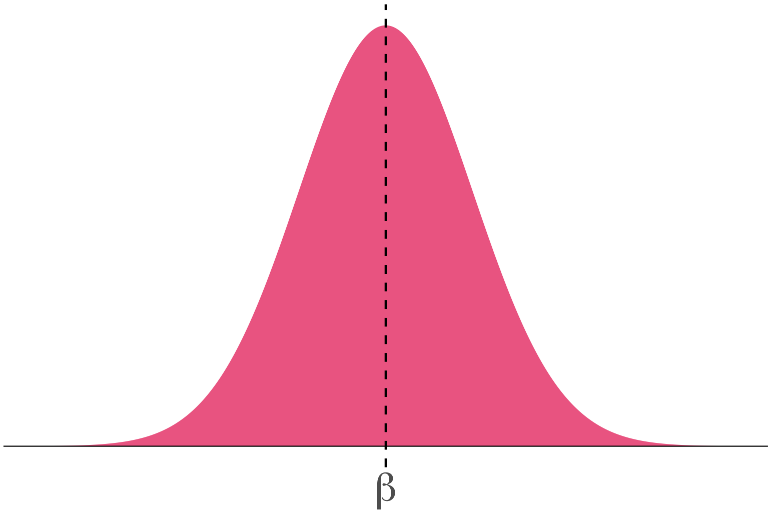
OLS: Assumptions and properties

Properties

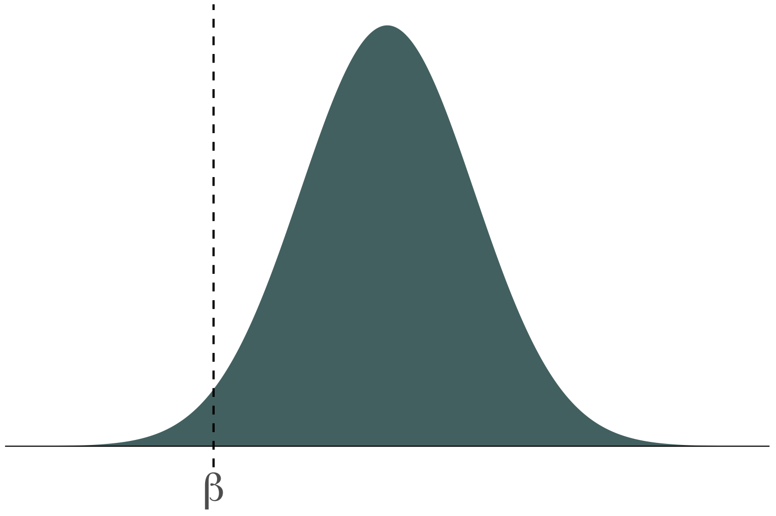
Question: What properties might we care about for an estimator?

Answer one: Bias.

Unbiased estimator: $\boldsymbol{E}\left[\hat{\beta}\right] = \beta$



Biased estimator: $\boldsymbol{E}\left[\hat{\beta}\right] \neq \beta$



OLS: Assumptions and properties

Properties

Question: What properties might we care about for an estimator?

Answer two: Variance.

The central tendencies (means) of competing distributions are not the only things that matter. We also care about the **variance** of an estimator.

$$\mathrm{Var}\left(\hat{\beta}\right) = \boldsymbol{E}\left[\left(\hat{\beta} - \boldsymbol{E}\left[\hat{\beta}\right]\right)^2\right]$$

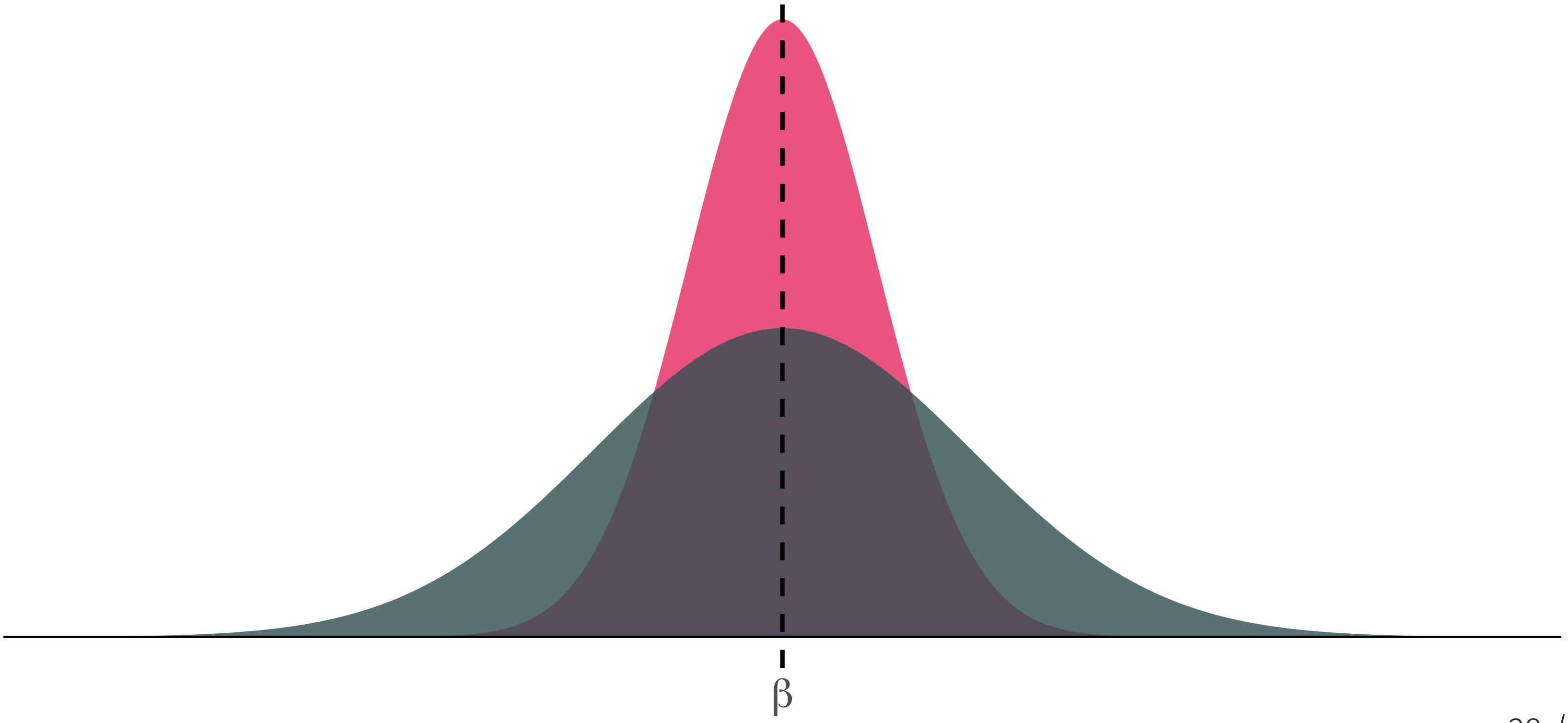
Lower variance estimators mean we get estimates closer to the mean in each sample.

OLS: Assumptions and properties

Properties

Question: What properties might we care about for an estimator?

Answer two: Variance.



Properties

Question: What properties might we care about for an estimator?

Answer one: Bias.

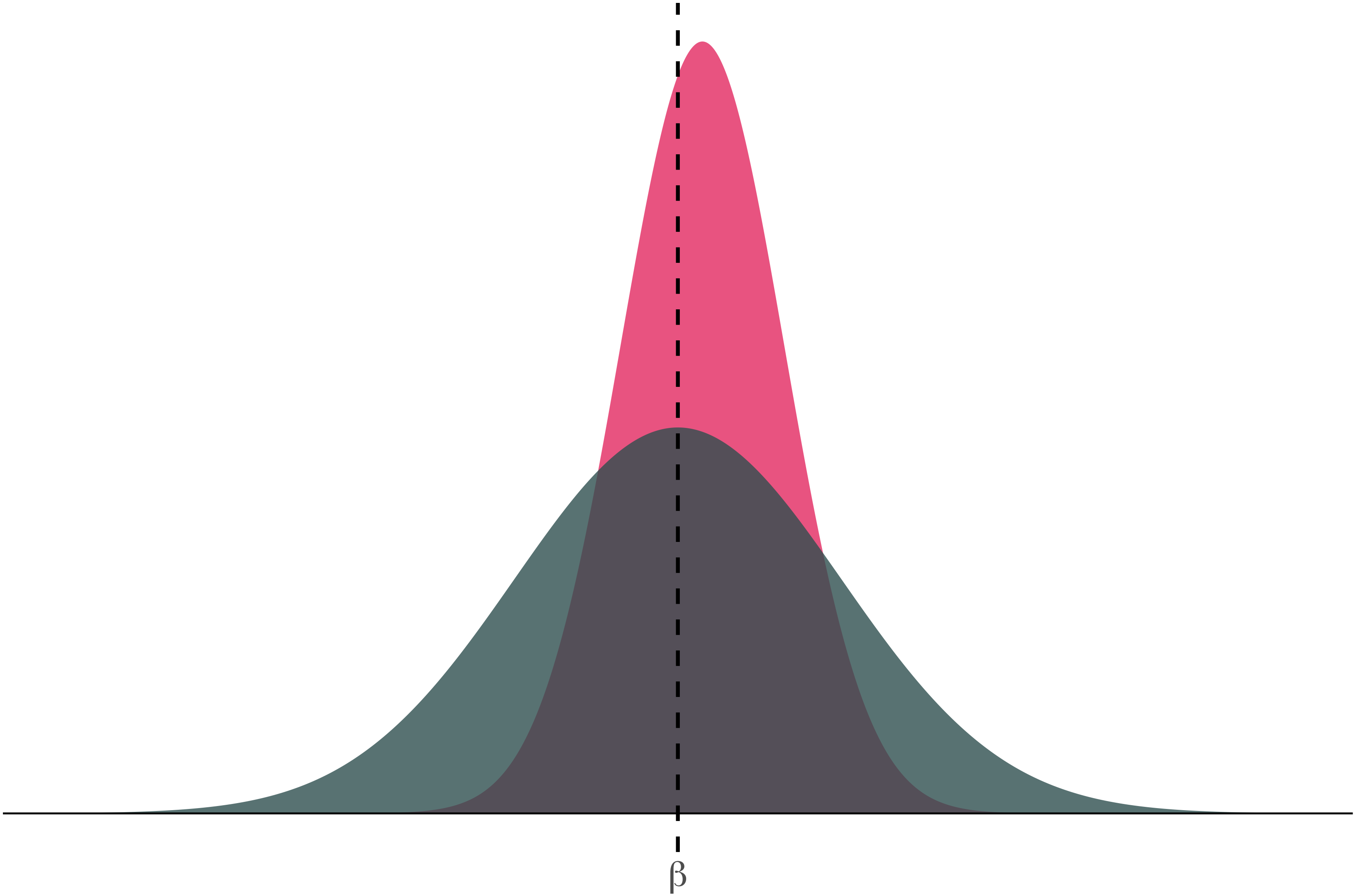
Answer two: Variance.

Subtlety: The bias-variance tradeoff.

Should we be willing to take a bit of bias to reduce the variance?

In econometrics, we generally stick with unbiased (or consistent) estimators. But other disciplines (especially computer science) think a bit more about this tradeoff.

The bias-variance tradeoff.



OLS: Assumptions and properties

Properties

As you might have guessed by now,

- OLS is **unbiased**.
- OLS has the **minimum variance** of all unbiased linear estimators.

OLS: Assumptions and properties

Properties

As you might have guessed by now,

- OLS is **unbiased**.
- OLS has the **minimum variance** of all unbiased linear estimators.

These (very nice) properties depend upon a set of assumptions:

1. The population relationship is linear in parameters with an additive disturbance.

2. Our $\boldsymbol{X}$ variable is **exogenous**, *i.e.*, $\boldsymbol{E}[u|X] = 0$.

3. The $\boldsymbol{X}$ variable has variation. And if there are multiple explanatory variables, they are not perfectly collinear.

4. The population disturbances u_i are independently and identically distributed as normal random variables with mean zero ($\boldsymbol{E}[u] = 0$) and variance σ^2 (*i.e.*, $\boldsymbol{E}\left[u^2\right] = \sigma^2$). Independently distributed and mean zero jointly imply $\boldsymbol{E}\left[u_i u_j\right] = 0$ for any $i \neq j$.

OLS: Assumptions and properties

Assumptions

- Assumptions (1), (2), and (3) make OLS unbiased.
- Assumption (4) gives us an unbiased estimator for the variance of our OLS estimator.

During our course, we will discuss the many ways real life may **violate these assumptions**. For instance:

- Non-linear relationships in our parameters/disturbances (or misspecification).
- Disturbances that are not identically distributed and/or not independent.
- Violations of exogeneity (especially omitted-variable bias).

Uncertainty and inference

Is there more?

Up to this point, we know OLS has some nice properties, and we know how to estimate an intercept and slope coefficient via OLS.

Our current workflow:

- Get data (points with x and y values)
- Regress y on x
- Plot the OLS line (*i.e.*, $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1$)
- Done?

But how do we actually **learn** something from this exercise?

- Based upon our value of $\hat{\beta}_1$, can we rule out previously hypothesized values?
- How confident should we be in the precision of our estimates?
- How well does our model explain the variation we observe in y?

We need to be able to deal with uncertainty. Enter: **Inference.**

Uncertainty and inference

Learning from our errors

As our previous simulation pointed out, our problem with **uncertainty** is that we don't know whether our sample estimate is *close* or *far* from the unknown population parameter.[†]

However, all is not lost. We can use the errors ($e_i = y_i - \hat{y}_i$) to get a sense of how well our model explains the observed variation in y.

When our model appears to be doing a "nice" job, we might be a little more confident in using it to learn about the relationship between y and x.

Now we just need to formalize what a "nice job" actually means.

[†]: Except when we run the simulation ourselves—which is why we like simulations.

Learning from our errors

First off, we will estimate the variance of u_i (recall: $\mathop{\text{Var}}(u_i) = \sigma^2$) using our squared errors, *i.e.*,

$$s^2 = \dfrac{\sum_i e_i^2}{n - k}$$

where k gives the number of slope terms and intercepts that we estimate (*e.g.*, β_0 and β_1 would give $k=2$).

s^2 is an unbiased estimator of σ^2.

Learning from our errors

You then showed that the variance of $\hat{\beta}_1$ (for simple linear regression) is

$$\mathbf{Var}\left(\hat{\beta}_1\right) = \frac{s^2}{\sum_i \left(x_i - \overline{x}\right)^2}$$

which shows that the variance of our slope estimator

1. increases as our disturbances become noisier
2. decreases as the variance of x increases

Learning from our errors

More common: The **standard error** of $\hat{\beta}_1$

$$\hat{\text{SE}}\left(\hat{\beta}_1\right) = \sqrt{\frac{s^2}{\sum_i \left(x_i - \overline{x}\right)^2}}$$

Recall: The standard error of an estimator is the standard deviation of the estimator's distribution.

Standard error output is standard in `R`'s `lm`:

```
tidy(lm(y ~ x, pop_df))
```

```
#> # A tibble: 2 x 5
#>   term        estimate std.error statistic  p.value
#>   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
#> 1 (Intercept)    2.53     0.422       6.00 3.38e- 8
#> 2 x              0.567    0.0793      7.15 1.59e-10
```

Learning from our errors

We use the standard error of $\hat{\beta}_1$, along with $\hat{\beta}_1$ itself, to learn about the parameter β_1.

After deriving the distribution of $\hat{\beta}_1$,[†] we have two (related) options for formal statistical inference (learning) about our unknown parameter β_1:

- **Confidence intervals:** Use the estimate and its standard error to create an interval that, when repeated, will generally[††] contain the true parameter.
- **Hypothesis tests:** Determine whether there is statistically significant evidence to reject a hypothesized value or range of values.

[†]: *Hint:* it's normal with the mean and variance we've derived/discussed above)

[††]: *E.g.*, Similarly constructed 95% confidence intervals will contain the true parameter 95% of the time.

Confidence intervals

Under our assumptions, we can construct $(1 - \alpha)$-level confidence intervals for β_1 as

$$\hat{\beta}_1 \pm t_{\alpha/2,\text{df}}\ \hat{\text{SE}}\left(\hat{\beta}_1\right)$$

$t_{\alpha/2,\text{df}}$ denotes the $\alpha/2$ quantile of a t distribution with $n - k$ degrees of freedom.

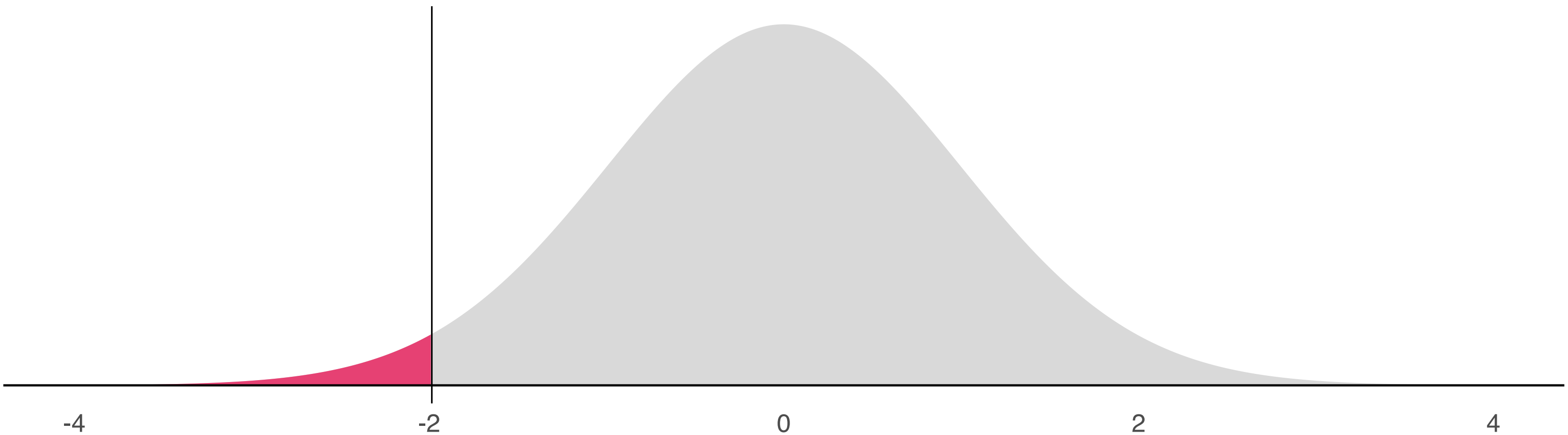
Confidence intervals

Under our assumptions, we can construct $(1-\alpha)$-level confidence intervals for β_1 as

$$\hat{\beta}_1 \pm t_{\alpha/2,\text{df}}\ \hat{\text{SE}}\left(\hat{\beta}_1\right)$$

$t_{\alpha/2,\text{df}}$ denotes the $\alpha/2$ quantile of a t distribution with $n-k$ degrees of freedom.

For example, 100 obs., two coefficients (*i.e.*, $\hat{\beta}_0$ and $\hat{\beta}_1 \implies k=2$), and $\alpha=0.05$ (for a 95% confidence interval) gives us $t_{0.025,\,98}=-1.98$



Confidence intervals

Under our assumptions, we can construct $(1-\alpha)$-level confidence intervals for β_1 as

$$\hat{\beta}_1 \pm t_{\alpha/2,\text{df}}\ \hat{\text{SE}}\left(\hat{\beta}_1\right)$$

$t_{\alpha/2,\text{df}}$ denotes the $\alpha/2$ quantile of a t distribution with $n-k$ degrees of freedom.

Example:

```
lm(y ~ x, data = pop_df) %>% tidy()
```

```
#> # A tibble: 2 x 5
#>   term        estimate std.error statistic  p.value
#>   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
#> 1 (Intercept)    2.53     0.422       6.00 3.38e- 8
#> 2 x              0.567    0.0793      7.15 1.59e-10
```

Confidence intervals

Under our assumptions, we can construct $(1-\alpha)$-level confidence intervals for β_1 as

$$\hat{\beta}_1 \pm t_{\alpha/2,\text{df}}\ \hat{\text{SE}}\left(\hat{\beta}_1\right)$$

$t_{\alpha/2,\text{df}}$ denotes the $\alpha/2$ quantile of a t distribution with $n-k$ degrees of freedom.

Example:

```
lm(y ~ x, data = pop_df) %>% tidy()
```

```
#> # A tibble: 2 x 5
#>   term        estimate std.error statistic  p.value
#>   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
#> 1 (Intercept)    2.53     0.422       6.00 3.38e- 8
#> 2 x              0.567    0.0793      7.15 1.59e-10
```

Our 95% confidence interval is thus

$$0.567 \pm 1.98 \times 0.0793 = [0.410,\ 0.724]$$

Confidence intervals

So we have a confidence interval for β_1, *i.e.*, $[\mathbf{0.410},\ \mathbf{0.724}]$.

What does it mean?

Confidence intervals

So we have a confidence interval for β_1, *i.e.*, $[\mathbf{0.410}, \mathbf{0.724}]$.

What does it mean?

Informally: The confidence interval gives us a region (interval) in which we can place some trust (confidence) for containing the parameter.

Confidence intervals

So we have a confidence interval for β_1, *i.e.*, $[0.410,\ 0.724]$.

What does it mean?

Informally: The confidence interval gives us a region (interval) in which we can place some trust (confidence) for containing the parameter.

More formally: If repeatedly sample from our population and construct confidence intervals for each of these samples, $(1 - \alpha)$ percent of our intervals (*e.g.*, 95%) will contain the population parameter *somewhere in the interval*.

Confidence intervals

So we have a confidence interval for β_1, *i.e.*, $[\mathbf{0.410}, \mathbf{0.724}]$.

What does it mean?

Informally: The confidence interval gives us a region (interval) in which we can place some trust (confidence) for containing the parameter.

More formally: If repeatedly sample from our population and construct confidence intervals for each of these samples, $(1-\alpha)$ percent of our intervals (*e.g.*, 95%) will contain the population parameter *somewhere in the interval*.

Now back to our simulation...

Confidence intervals

We drew 10,000 samples (each of size $n = 30$) from our population and estimated our regression model for each of these simulations:

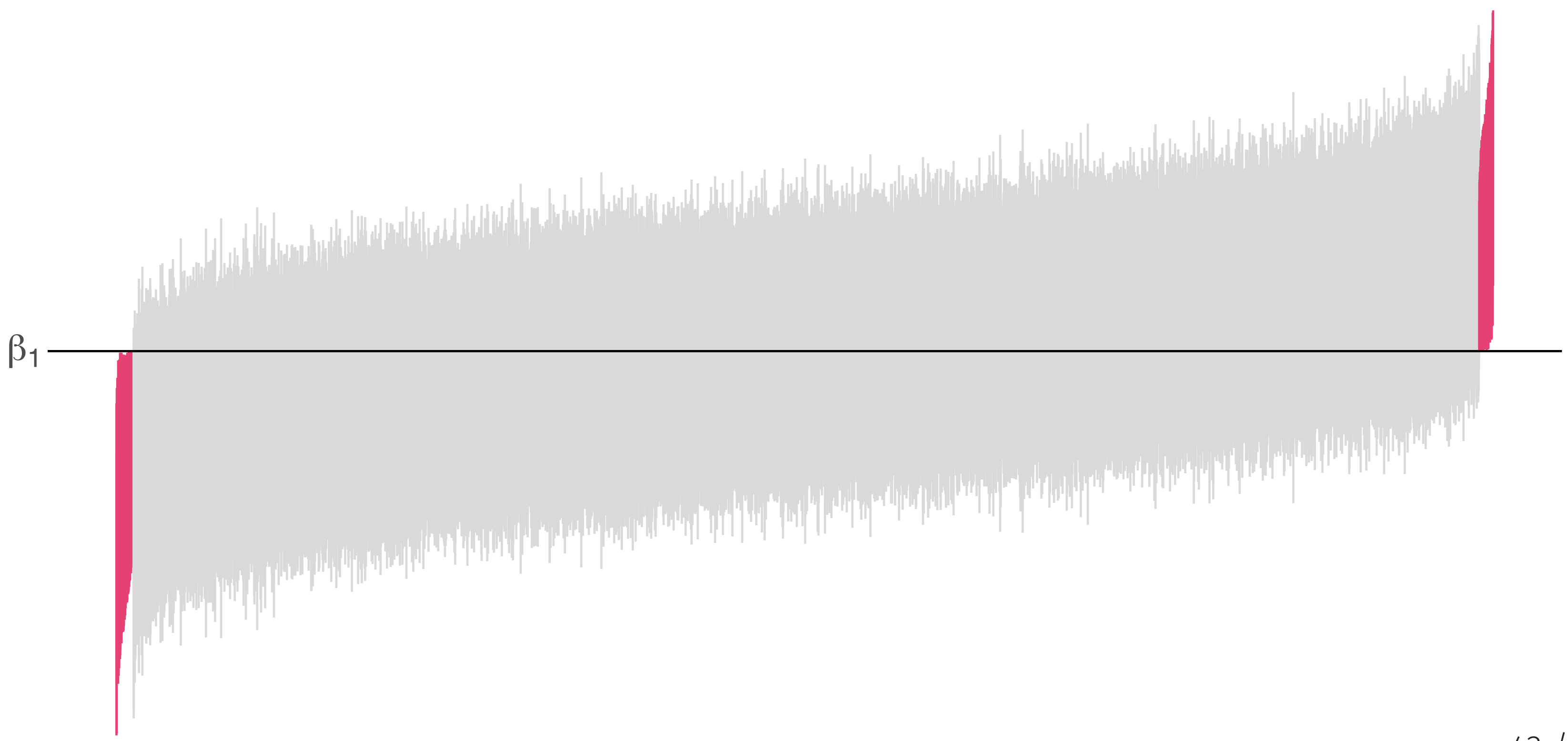
$$y_i = \hat{\beta}_0 + \hat{\beta}_1 x_i + e_i$$

(repeated 10,000 times)

Now, let's estimate 95% confidence intervals for each of these intervals...

Confidence intervals

From our previous simulation: 97.7% of 95% confidences intervals contain the true parameter value of β_1.



Hypothesis testing

In many applications, we want to know more than a point estimate or a range of values. We want to know what our statistical evidence says about existing theories.

We want to test hypotheses posed by officials, politicians, economists, scientists, friends, weird neighbors, *etc.*

Examples

- Does increasing police presence **reduce crime**?
- Does building a giant wall **reduce crime**?
- Does shutting down a government **adversely affect the economy**?
- Does legalizing cannabis **reduce drunk driving** and/or **reduce opiod use**?
- Do more stringent air quality standards **increase birthweights** and/or **reduce jobs**?

Hypothesis testing

Hypothesis testing relies upon very similar results and intuition as confidence intervals.

While uncertainty certainly exists, we can build statistical tests that generally get the answers right (rejecting or failing to reject a posited hypothesis).

Hypothesis testing

Hypothesis testing relies upon very similar results and intuition as confidence intervals.

While uncertainty certainly exists, we can build statistical tests that generally get the answers right (rejecting or failing to reject a posited hypothesis).

OLS t test

Imagine a (null) hypothesis which states that β_1 is equal to some value c, *i.e.*, $H_0 :\ \beta_1 = c$

From OLS's properties, we can show that the test statistic

$$t_{\text{stat}} = \frac{\hat{\beta}_1 - c}{\hat{\text{SE}}\left(\hat{\beta}_1\right)}$$

follows the t distribution with $n - k$ degrees of freedom.

Hypothesis testing

For an α-level, **two-sided** test, we reject the null hypothesis (and conclude with the alternative hypothesis) when

$$\left|t_{\text{stat}}\right| > \left|t_{1-\alpha/2,\,df}\right|$$

meaning that our **test statistic is more extreme than the critical value**.

Alternatively, we can calculate the **p-value** that accompanies our test statistic, which effectively gives us the probability of seeing our test statistic *or a more extreme test statistic* if the null hypothesis were true.

Very small p-values (generally < 0.05) mean that it would be unlikely to see our results if the null hyopthesis were really true—we tend to reject the null for p-values below 0.05.

Hypothesis testing

Example

Standard `R` and `Stata` output tends to test hypotheses against the value zero.

```
lm(y ~ x, data = pop_df) %>% tidy()
```

```
#> # A tibble: 2 x 5
#>   term        estimate std.error statistic  p.value
#>   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
#> 1 (Intercept)    2.53     0.422       6.00 3.38e- 8
#> 2 x              0.567    0.0793      7.15 1.59e-10
```

Hypothesis testing

Example

Standard `R` and `Stata` output tends to test hypotheses against the value zero.

```
lm(y ~ x, data = pop_df) %>% tidy()
```

```
#> # A tibble: 2 x 5
#>   term        estimate std.error statistic  p.value
#>   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
#> 1 (Intercept)    2.53     0.422       6.00 3.38e- 8
#> 2 x              0.567    0.0793      7.15 1.59e-10
```

$H_0: \beta_1 = 0$ and $H_1: \beta_1 \neq 0$

Hypothesis testing

Example

Standard `R` and `Stata` output tends to test hypotheses against the value zero.

```
lm(y ~ x, data = pop_df) %>% tidy()
```

```
#> # A tibble: 2 x 5
#>   term        estimate std.error statistic  p.value
#>   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
#> 1 (Intercept)    2.53     0.422       6.00 3.38e- 8
#> 2 x              0.567    0.0793      7.15 1.59e-10
```

$H_0 : \beta_1 = 0$ and $H_1 : \beta_1 \neq 0$

$t_{\text{stat}} = 7.15$ and $t_{0.975,\, 28} = 2.05$

Hypothesis testing

Example

Standard `R` and `Stata` output tends to test hypotheses against the value zero.

```
lm(y ~ x, data = pop_df) %>% tidy()
```

```
#> # A tibble: 2 x 5
#>   term        estimate std.error statistic  p.value
#>   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
#> 1 (Intercept)    2.53     0.422       6.00 3.38e- 8
#> 2 x              0.567    0.0793      7.15 1.59e-10
```

$H_0:\ \beta_1 = 0$ and $H_1:\ \beta_1 \neq 0$

$t_\text{stat} = 7.15$ and $t_\text{0.975, 28} = 2.05$

$\text{p-value} < 0.05$

Uncertainty and inference

Hypothesis testing

Example

Standard `R` and `Stata` output tends to test hypotheses against the value zero.

```
lm(y ~ x, data = pop_df) %>% tidy()
```

```
#> # A tibble: 2 x 5
#>   term        estimate std.error statistic  p.value
#>   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
#> 1 (Intercept)    2.53     0.422       6.00 3.38e- 8
#> 2 x              0.567    0.0793      7.15 1.59e-10
```

$H_0 :\ \beta_1 = 0$ and $H_1 :\ \beta_1 \neq 0$

$t_\text{stat} = 7.15$ and $t_\text{0.975, 28} = 2.05$

$\text{p-value} < 0.05$

Reject H_0

Hypothesis testing

Back to our simulation! Let's see what our t statistic is actually doing.

In this situation, we can actually know (and enforce) the null hypothesis, since we generated the data.

For each of the 10,000 samples, we will calculate the t statistic, and then we can see how many t statistics exceed our critical value (2.05, as above).

The answer should be approximately 5 percent—our α level.

Uncertainty and inference