Potential Outcomes

Manoel Horta Ribeiro
manoel@cs.princeton.edu

COS 598D / Spring 2026

Let’s Rewind

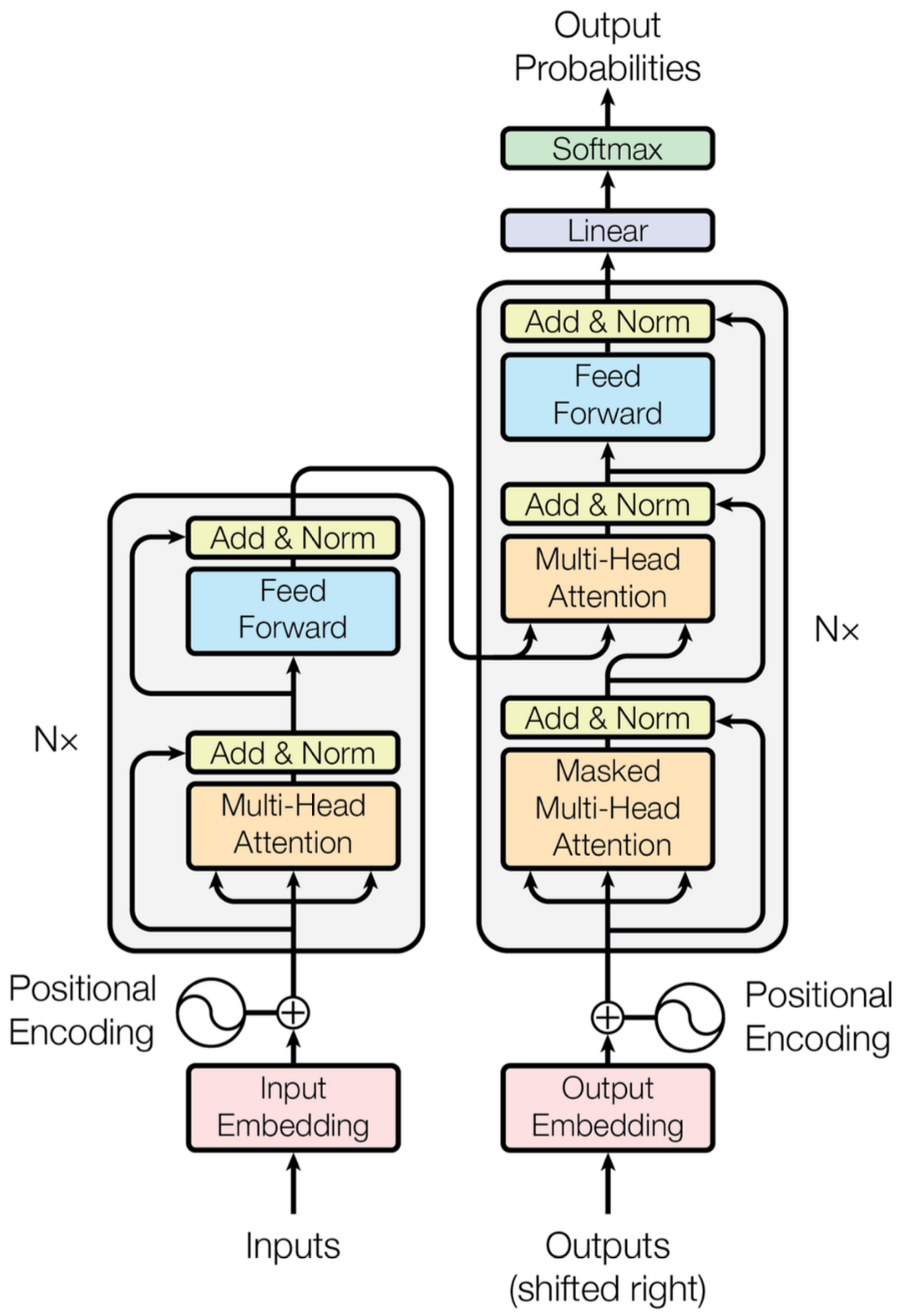


Figure 1: The Transformer - model architecture.

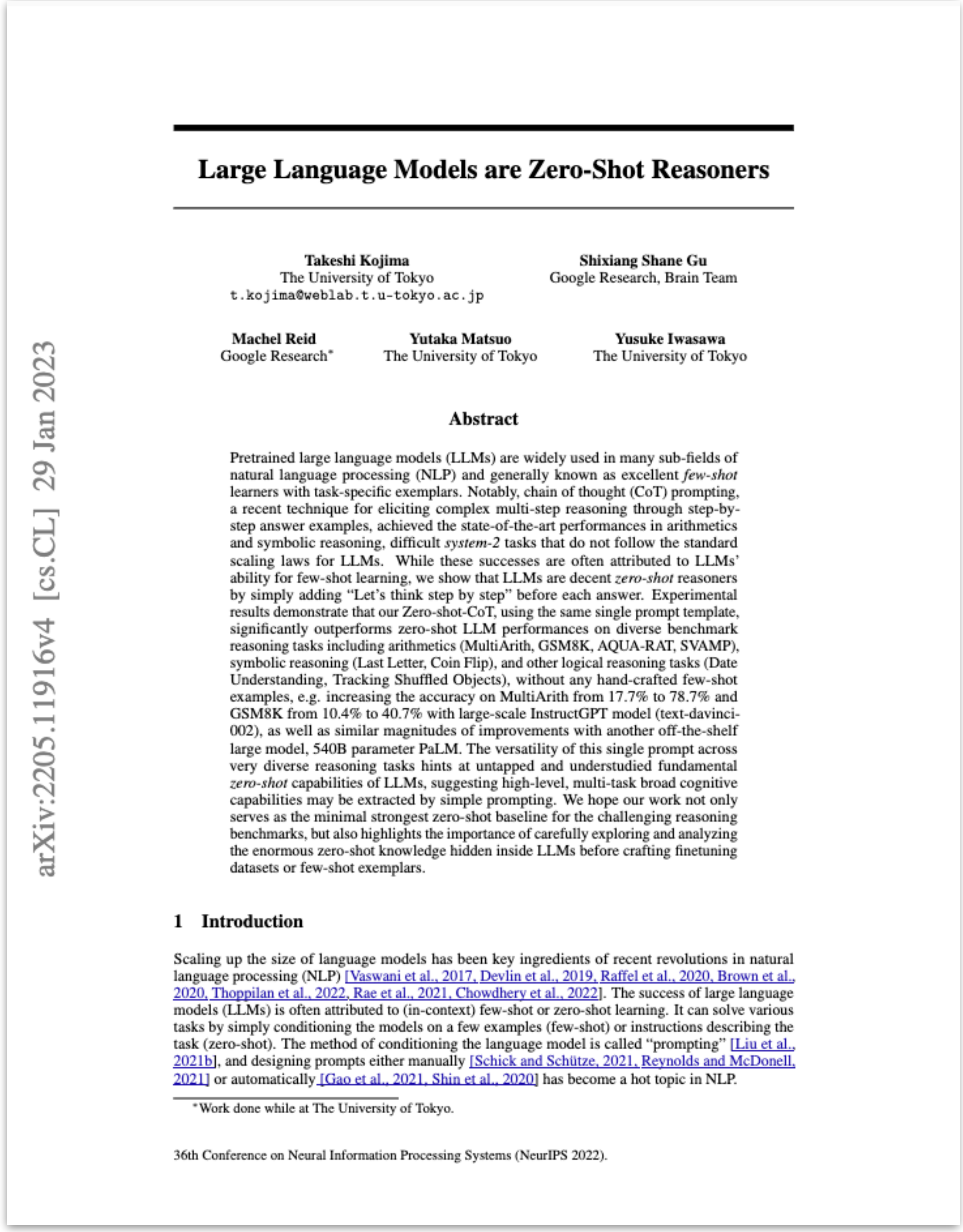
- Back in 2020, OpenAI released GPT-3, their largest model and most powerful so far.

- The model was a decoder-only transformer trained *only* to predict the next token (no "post-training").

Chain of Thought (CoT)

- In an influential paper, Kojima et al. (2022) proposed changing the way we prompt models.
- **Hypothesis**: Asking the model to reason improves performance.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there? A: The answer (arabic numerals) is	Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there? **A: Let's think step by step.**



Large Language Models are Zero-Shot Reasoners

Takeshi Kojima
The University of Tokyo
t.kojima@weblab.t.u-tokyo.ac.jp

Shixiang Shane Gu
Google Research, Brain Team

Machel Reid
Google Research*

Yutaka Matsuo
The University of Tokyo

Yusuke Iwasawa
The University of Tokyo

Abstract

Pretrained large language models (LLMs) are widely used in many sub-fields of natural language processing (NLP) and generally known as excellent *few-shot* learners with task-specific exemplars. Notably, chain of thought (CoT) prompting, a recent technique for eliciting complex multi-step reasoning through step-by-step answer examples, achieved the state-of-the-art performances in arithmetics and symbolic reasoning, difficult *system-2* tasks that do not follow the standard scaling laws for LLMs. While these successes are often attributed to LLMs' ability for few-shot learning, we show that LLMs are decent *zero-shot* reasoners by simply adding "Let's think step by step" before each answer. Experimental results demonstrate that our Zero-shot-CoT, using the same single prompt template, significantly outperforms zero-shot LLM performances on diverse benchmark reasoning tasks including arithmetics (MultiArith, GSM8K, AQUA-RAT, SVAMP), symbolic reasoning (Last Letter, Coin Flip), and other logical reasoning tasks (Date Understanding, Tracking Shuffled Objects), without any hand-crafted few-shot examples, e.g. increasing the accuracy on MultiArith from 17.7% to 78.7% and GSM8K from 10.4% to 40.7% with large-scale InstructGPT model (text-davinci-002), as well as similar magnitudes of improvements with another off-the-shelf large model, 540B parameter PaLM. The versatility of this single prompt across very diverse reasoning tasks hints at untapped and understudied fundamental *zero-shot* capabilities of LLMs, suggesting high-level, multi-task broad cognitive capabilities may be extracted by simple prompting. We hope our work not only serves as the minimal strongest zero-shot baseline for the challenging reasoning benchmarks, but also highlights the importance of carefully exploring and analyzing the enormous zero-shot knowledge hidden inside LLMs before crafting finetuning datasets or few-shot exemplars.

1 Introduction

Scaling up the size of language models has been key ingredients of recent revolutions in natural language processing (NLP) [Vaswani et al., 2017, Devlin et al., 2019, Raffel et al., 2020, Brown et al., 2020, Thoppilan et al., 2022, Rae et al., 2021, Chowdhery et al., 2022]. The success of large language models (LLMs) is often attributed to (in-context) few-shot or zero-shot learning. It can solve various tasks by simply conditioning the models on a few examples (few-shot) or instructions describing the task (zero-shot). The method of conditioning the language model is called "prompting" [Liu et al., 2021b], and designing prompts either manually [Schick and Schütze, 2021, Reynolds and McDonell, 2021] or automatically [Gao et al., 2021, Shin et al., 2020] has become a hot topic in NLP.

[*] Work done while at The University of Tokyo.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

Formalizing the Setup

- Let $\mathscr{D}$ be the **full dataset** of evaluation attempts
- Let i index problems and r index runs. A single *evaluation attempt* is a pair (i, r).
- For each evaluation attempt (i, r), we choose a prompt:
    - $T_{ir} = 0$: standard (non-CoT) prompt;
    - $T_{ir} = 1$: CoT prompt;
- Let $Y_{ir} \in \{0,1\}$ indicate whether the model solves problem i correctly when run r under the chosen strategy T_{ir}.

The Fundamental Problem

- Causal inference cares about *counterfactuals:* what would have happened had something been different.
  - E.g., would we solve problem i at run r had we used CoT instead of the standard prompt (or vice versa)?

- We can formalize this with potential outcomes at the (problem, run) level:
  - Y_{ir}^{0}: correctness for (i, r) **if** we use the standard prompt
  - Y_{ir}^{1}: correctness for (i, r) **if** we use the CoT prompt

The Fundamental Problem (cont'd)

For each evaluation attempt (i, r), we cannot observe both potential outcomes.

$$Y_{ir} = \begin{cases} Y_{ir}^{0} & \text{if } T_{ir} = 0, \\ Y_{ir}^{1} & \text{if } T_{ir} = 1 \end{cases}$$

We never get to see "what would have happened" for
that same problem and same run under the alternative prompt!

The Fundamental Problem (cont'd)

- Things are typically *more dire.*
  - Imagine that units are individuals, the treatment is a vaccine, and the outcome is death.
  - Then, we can only observe each individual i under one treatment.
- In our setup, however, it is reasonable to observe the same problem across two different treatments (e.g., prompts).
- However, note that each run is stochastic.

Individual Causal Effect

- The **individual causal effect** of CoT considering problem i and run r is:

$$\tau_{ir} = Y_{ir}^{1} - Y_{ir}^{0} \in \{-1,0,1\}$$

- This is impossible to recover, per FPCI!

- Yet, we can estimate the problem-level causal effect

$$\tau_i = E_r\left[Y_{ir}^{1} \mid i\right] - E_r\left[Y_{ir}^{0} \mid i\right].$$

Average Treatment Effect

- Consider a population of evaluation attempts (i, r). The ATE of CoT prompting is

$$ATE = E\left[Y_{ir}^{1} - Y_{ir}^{0}\right]$$

- We *can* estimate the ATE under particular assumptions.

- Other quantities of potential interest:
  - $ATT = E\left[Y_{ir}^{1} - Y_{ir}^{0} \mid T_{ir} = 1\right]$
  - $ATC = E\left[Y_{ir}^{1} - Y_{ir}^{0} \mid T_{ir} = 0\right]$

Exchangeability and Consistency

- Let Y and T be binary random variables specifying treatments and outcomes.

- **Exchangeability** entails that:

$$Y^a \perp T \text{ for all } a \in \{0,1\}$$

- **Consistency** entails that:

$$\text{If } T_{ir} = a \text{ then } Y_{ir} = Y_{ir}^a$$

No interference

“The outcome of a unit is unaffected by anyone else's treatment.”

In our CoT experiment: the correctness of problem i on run r under a given prompting strategy does not depend on which prompt we used for *other* problems or runs.

Can you think of any violations?

Average Treatment Effect (cont'd)

$$
\begin{aligned}
ATE &= E\left[Y_{ir}^{1} - Y_{ir}^{0}\right] \\
&= E\left[Y_{ir}^{1}\right] - E\left[Y_{ir}^{0}\right] && (1) \text{ Linearity of Expectations} \\
&= E[Y_{ir}^{1} \mid T_{ir} = 1] - E[Y_{ir}^{0} \mid T_{ir} = 0] && (2) \text{ Exchangeability} \\
&= E[Y_{ir} \mid T_{ir} = 1] - E[Y_{ir} \mid T_{ir} = 0] && (3) \text{ Consistency}
\end{aligned}
$$

We can estimate the quantity in the bottom from the data!

The Beauty of Experiments

- Suppose we wanted to assess the impact of adding “let’s think step by step” from *observational data*!
- People may be more likely to use CoT for harder problems.
- This means no exchangeability:

$$(Y_{ir}^0, Y_{ir}^1) \not\perp T_{ir} \,.$$

The Beauty of Experiments (Cont'd)

- Experiments guarantee exchangeability *by design.*
  - *E.g., assign units to treatment/control with a coin flip.*

- In our benchmark example, we assign each problem to treatment and control for the same number of runs. This also guarantees exchangeability.

- **Why?** Because the index r is just an arbitrary label that has no relationship to the potential outcomes.

Benchmarks and Experiments

- ML papers evaluate results using benchmarks.
- MATH-500 is a subset of the MATH dataset [Lightman et al. (2023)] commonly used to measure the reasoning capabilities of LMs.

Convert the point $(0,3)$ in rectangular coordinates to polar coordinates.
Enter your answer in the form (r,θ), where $r > 0$ and $0 \le \theta < 2\pi$

- Kojima et al. (2022) are doing an experiment!

Envisioning the Experiment

- Each *evaluation attempt* is a pair (i, r) comprising a problem index i and a run index r.

- For each (i, r) we record:
  - $T_{ir} \in \{0,1\}$
  - $Y_{ir} \in \{0,1\}$

- Let us assume we do 30 runs per problem per condition.

Data Example

Problem	Run	Prompt	Correct?
1	1	0 (Standard)	0
1	2	1 (CoT)	1
…	…	…	…
1	30	1 (CoT)	1
…	…	…	…
2	1	0 (Standard)	1
…	…	…	…

Estimating the ATE

- Recall that under our various assumptions, we have that:

$$ATE = E[Y_{ir} \mid T_{ir} = 1] - E[Y_{ir} \mid T_{ir} = 0]$$

- The sample estimator of the ATE is:

$$\hat{\tau} = \frac{1}{|\mathcal{D}_1|} \sum_{(i,r):T_{ir}=1} Y_{ir} - \frac{1}{|\mathcal{D}_0|} \sum_{(i,r):T_{ir}=0} Y_{ir}$$

- $\hat{\tau}$ converges to the true ATE as the sample size grows.

Brief Recap: Estim*and*|*ator*|*ate*

- **Estimand**: what you wish you could know!

$$ATE = E[Y_{ir} \mid T_{ir} = 1] - E[Y_{ir} \mid T_{ir} = 0]$$

- **Estimator**: a function of the data that targets the estimand

$$\hat{\tau} = \frac{1}{|\mathscr{D}_1|} \sum_{(i,r):T_{ir}=1} Y_{ir} - \frac{1}{|\mathscr{D}_0|} \sum_{(i,r):T_{ir}=0} Y_{ir}$$

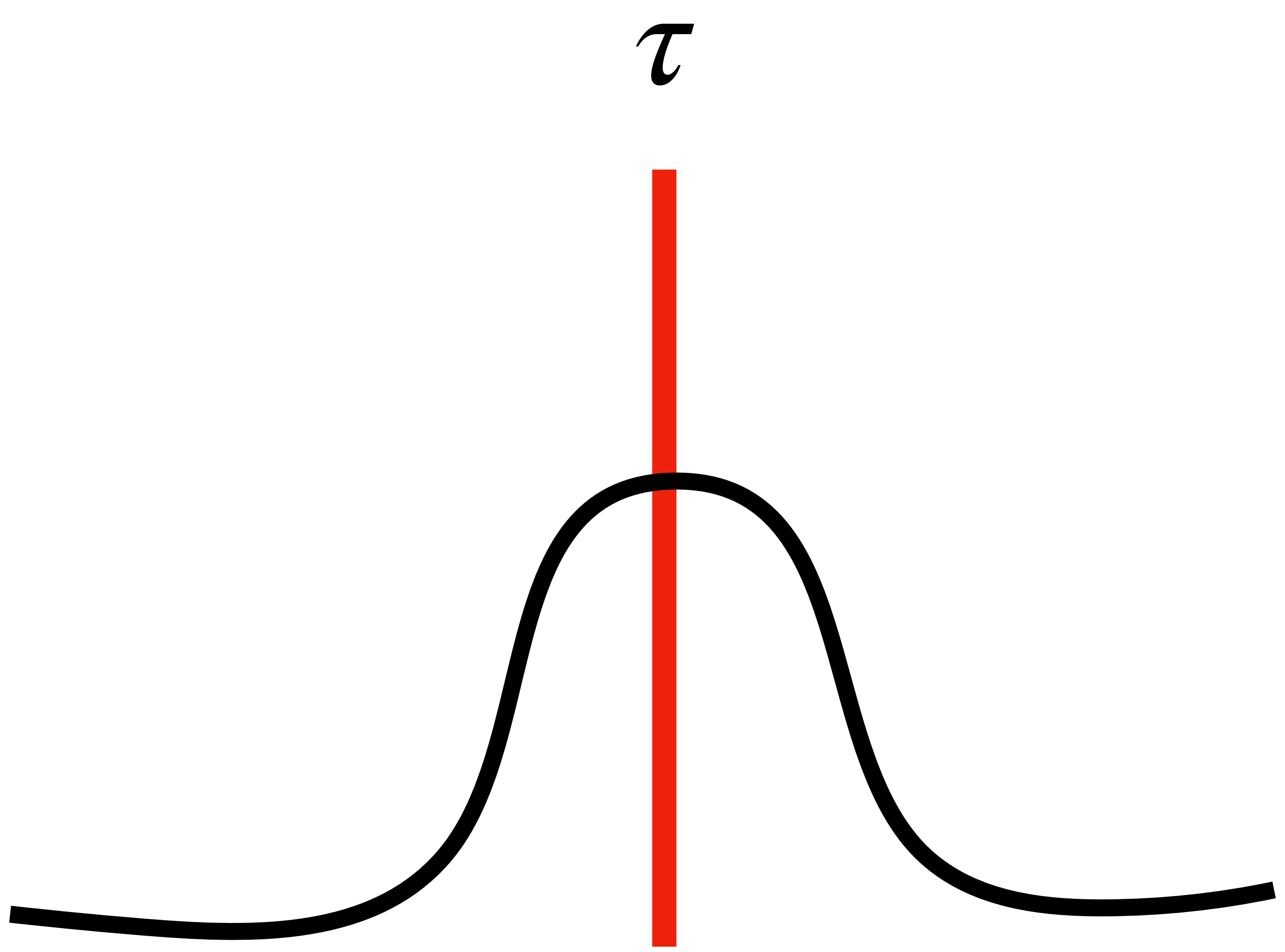
- **Estimate**: the value you computed from the data.

$$0.3$$

Estimator properties

- Because the estimator is random, you can think about it as a distribution.
- We favor estimators whose distributions have some specific properties.

Consistency: $E[\hat{\tau}] = \tau$

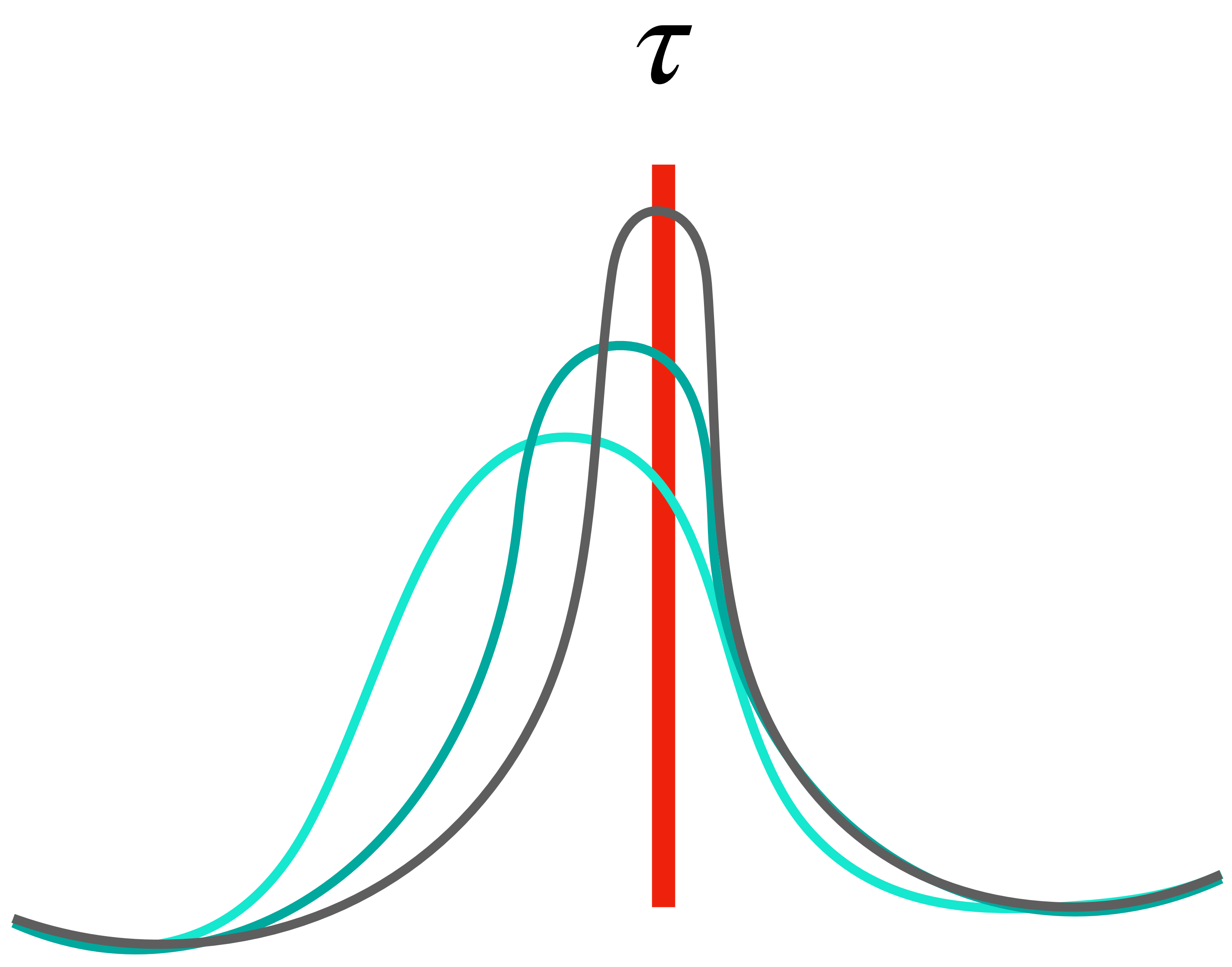


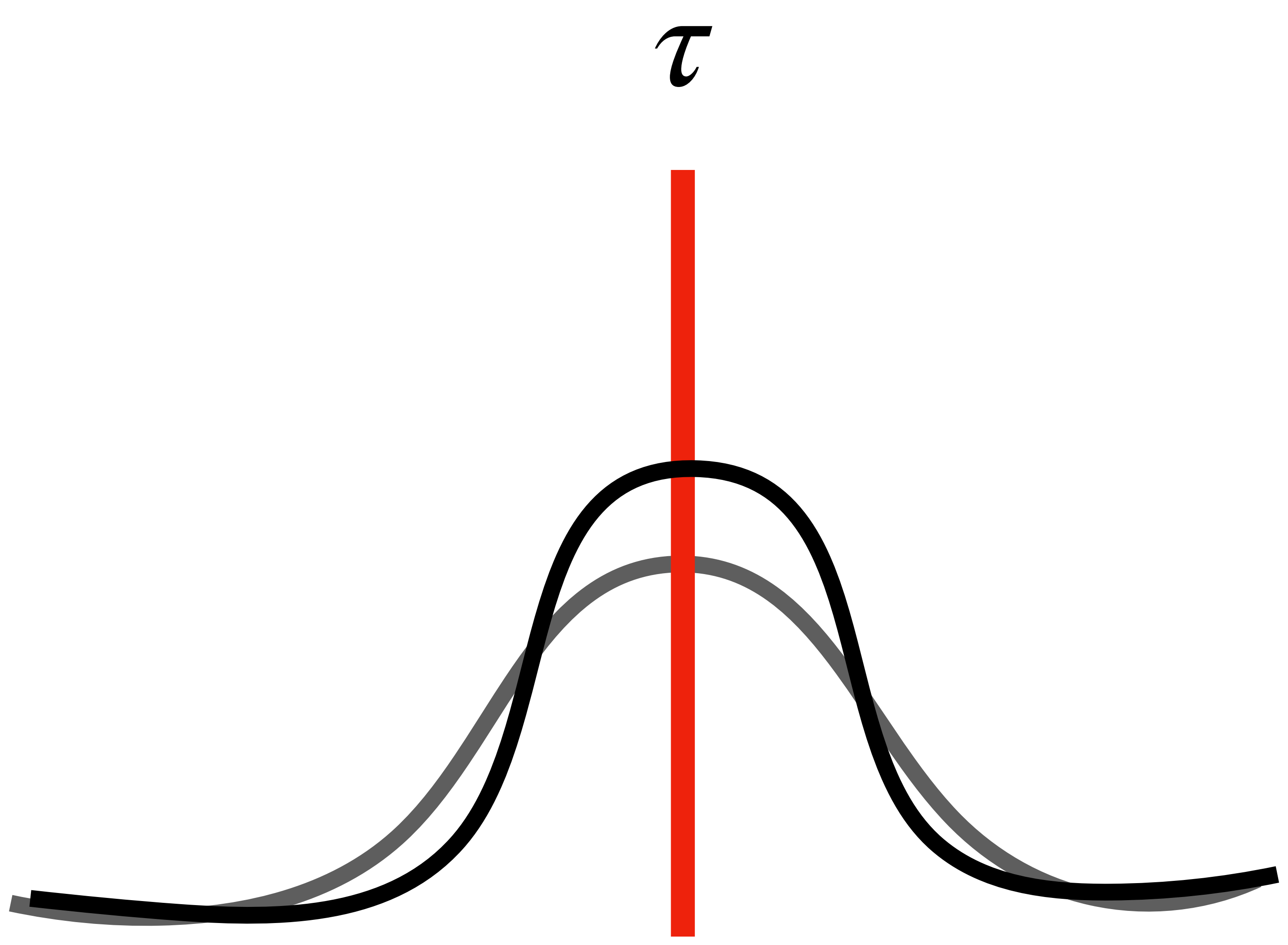
Estimator properties (cont'd)

Usually under a bunch of assumptions

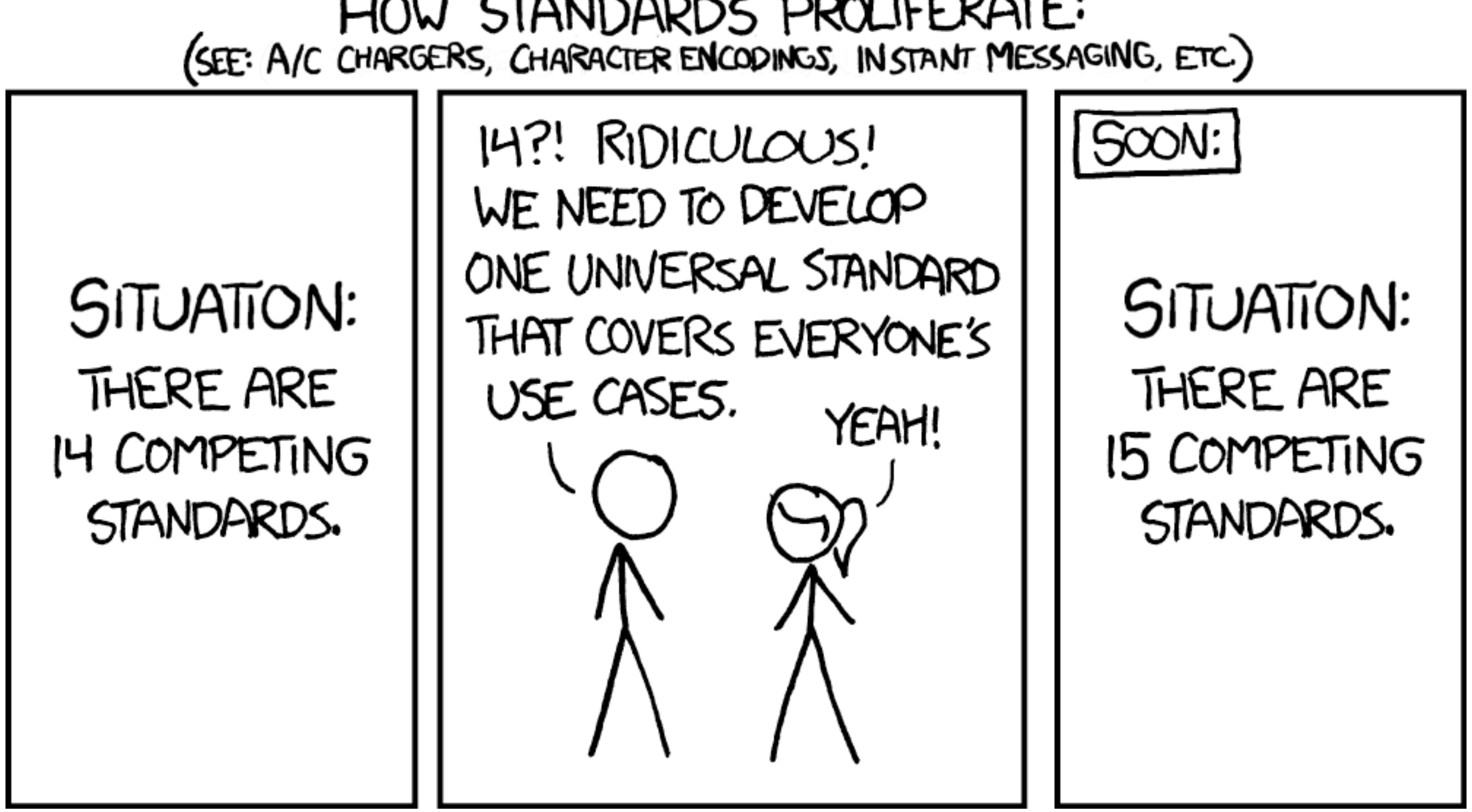
Consistency: converges to the estimand as $n \to \infty$!

Efficiency: The estimator has the smallest possible variance.





Commercial Break


HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC.)
SITUATION: THERE ARE 14 COMPETING STANDARDS.
14?! RIDICULOUS! WE NEED TO DEVELOP ONE UNIVERSAL STANDARD THAT COVERS EVERYONE'S USE CASES.
YEAH!
SOON:
SITUATION: THERE ARE 15 COMPETING STANDARDS.

Quantifying Uncertainty

- In finite samples, $\hat{\tau}$ will fluctuate by chance.
    - Suppose no difference between CoT/normal prompting.
    - We evaluate 5 problems, 1 run per condition.
    - Equivalent to: tossing two fair points 5x, comparing # of heads.

- So how can we tell whether we are not observing a positive or a negative $\hat{\tau}$ just by chance?

- We need ways to **quantify uncertainty**!

Standard Error

- Let's assume that each evaluation attempt is *iid*.
- Considering $a \in \{0,1\}$, we can define:

$$s_a^2 = \frac{1}{n_1 - 1} \sum_{(i,r):T_{ir}=a} \left(Y_{ir} - \bar{Y}_a\right)^2$$

$$SE(\hat{\tau}) = \sqrt{\frac{s_1^2}{|\mathscr{D}_1|} + \frac{s_0^2}{|\mathscr{D}_0|}}\,.$$

Hypothesis Testing

- Let’s formalize what we mean by “just by chance.”
- Null hypothesis (no effect of CoT):

$$H_0 : ATE = 0$$

- Alternative hypothesis (some effect):

$$H_1 : ATE \neq 0$$

- **Idea**: Compute a statistic. Ask how surprising given H_0?

Hypothesis Testing (cont'd)

- Under certain conditions, and under H_0, i.e., $\tau = 0$.

$$Z = \frac{\hat{\tau} - \tau}{SE(\hat{\tau})} = \frac{\hat{\tau} - 0}{SE(\hat{\tau})} \sim \mathcal{N}(0,1)$$

- p-value: the probability, under the null, of observing a test statistic as extreme or more extreme than observed:

$$p = P(|Z| \geq |z_{obs}| \mid H_0)\,.$$

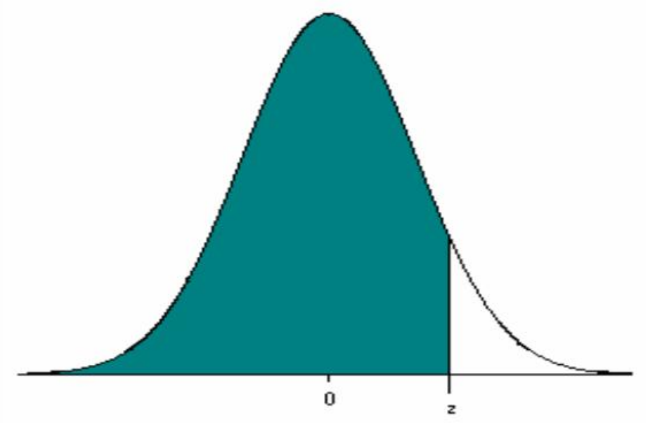
$\Phi(\,\cdot\,)$

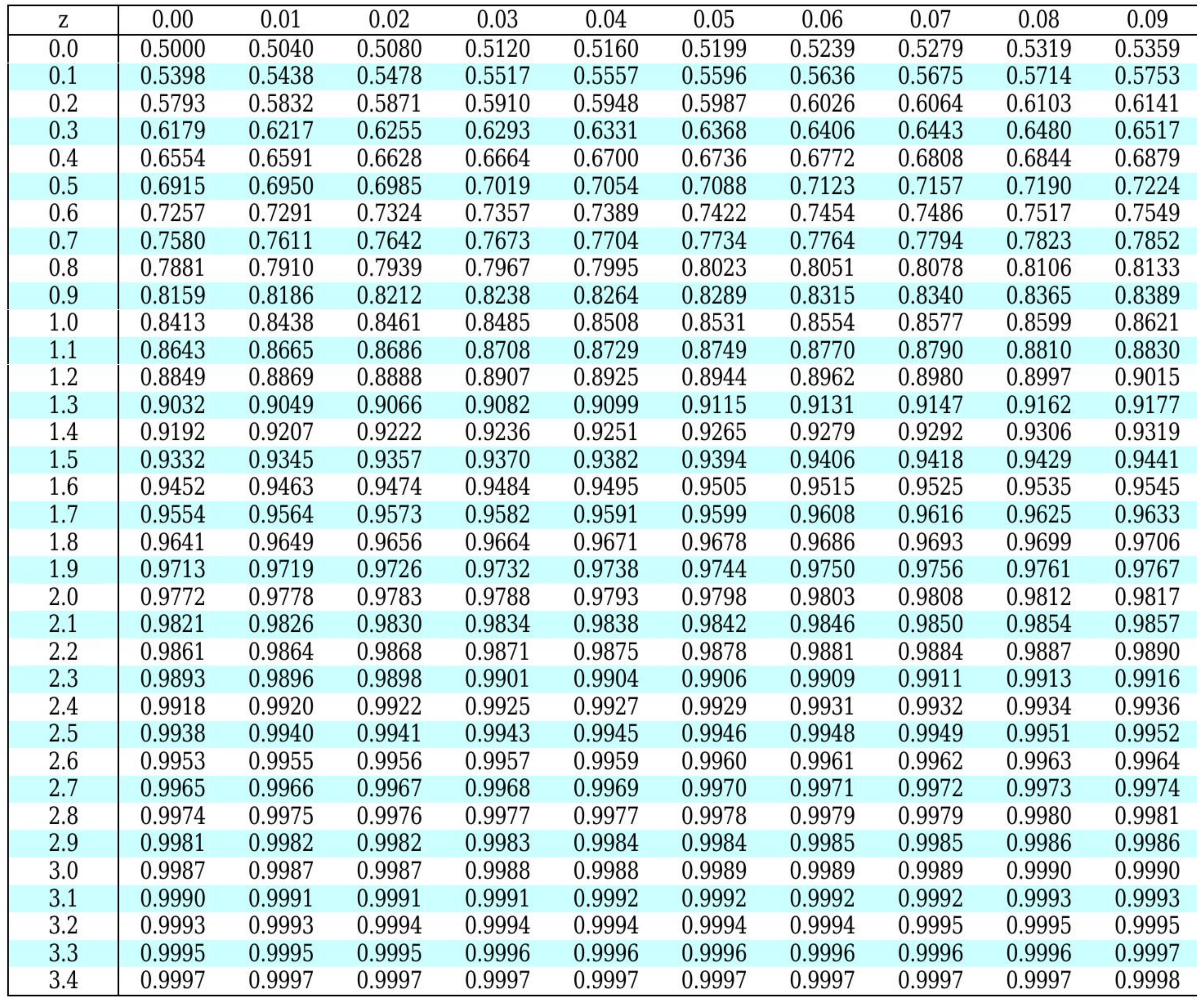
- We have a formula for that!

$$\Phi(x) = \Pr(Z \leq x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-t^2/2}\, dt$$

- You can use a table…

- Or a few lines of code…





z	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0.0	0.5000	0.5040	0.5080	0.5120	0.5160	0.5199	0.5239	0.5279	0.5319	0.5359
0.1	0.5398	0.5438	0.5478	0.5517	0.5557	0.5596	0.5636	0.5675	0.5714	0.5753
0.2	0.5793	0.5832	0.5871	0.5910	0.5948	0.5987	0.6026	0.6064	0.6103	0.6141
0.3	0.6179	0.6217	0.6255	0.6293	0.6331	0.6368	0.6406	0.6443	0.6480	0.6517
0.4	0.6554	0.6591	0.6628	0.6664	0.6700	0.6736	0.6772	0.6808	0.6844	0.6879
0.5	0.6915	0.6950	0.6985	0.7019	0.7054	0.7088	0.7123	0.7157	0.7190	0.7224
0.6	0.7257	0.7291	0.7324	0.7357	0.7389	0.7422	0.7454	0.7486	0.7517	0.7549
0.7	0.7580	0.7611	0.7642	0.7673	0.7704	0.7734	0.7764	0.7794	0.7823	0.7852
0.8	0.7881	0.7910	0.7939	0.7967	0.7995	0.8023	0.8051	0.8078	0.8106	0.8133
0.9	0.8159	0.8186	0.8212	0.8238	0.8264	0.8289	0.8315	0.8340	0.8365	0.8389
1.0	0.8413	0.8438	0.8461	0.8485	0.8508	0.8531	0.8554	0.8577	0.8599	0.8621
1.1	0.8643	0.8665	0.8686	0.8708	0.8729	0.8749	0.8770	0.8790	0.8810	0.8830
1.2	0.8849	0.8869	0.8888	0.8907	0.8925	0.8944	0.8962	0.8980	0.8997	0.9015
1.3	0.9032	0.9049	0.9066	0.9082	0.9099	0.9115	0.9131	0.9147	0.9162	0.9177
1.4	0.9192	0.9207	0.9222	0.9236	0.9251	0.9265	0.9279	0.9292	0.9306	0.9319
1.5	0.9332	0.9345	0.9357	0.9370	0.9382	0.9394	0.9406	0.9418	0.9429	0.9441
1.6	0.9452	0.9463	0.9474	0.9484	0.9495	0.9505	0.9515	0.9525	0.9535	0.9545
1.7	0.9554	0.9564	0.9573	0.9582	0.9591	0.9599	0.9608	0.9616	0.9625	0.9633
1.8	0.9641	0.9649	0.9656	0.9664	0.9671	0.9678	0.9686	0.9693	0.9699	0.9706
1.9	0.9713	0.9719	0.9726	0.9732	0.9738	0.9744	0.9750	0.9756	0.9761	0.9767
2.0	0.9772	0.9778	0.9783	0.9788	0.9793	0.9798	0.9803	0.9808	0.9812	0.9817
2.1	0.9821	0.9826	0.9830	0.9834	0.9838	0.9842	0.9846	0.9850	0.9854	0.9857
2.2	0.9861	0.9864	0.9868	0.9871	0.9875	0.9878	0.9881	0.9884	0.9887	0.9890
2.3	0.9893	0.9896	0.9898	0.9901	0.9904	0.9906	0.9909	0.9911	0.9913	0.9916
2.4	0.9918	0.9920	0.9922	0.9925	0.9927	0.9929	0.9931	0.9932	0.9934	0.9936
2.5	0.9938	0.9940	0.9941	0.9943	0.9945	0.9946	0.9948	0.9949	0.9951	0.9952
2.6	0.9953	0.9955	0.9956	0.9957	0.9959	0.9960	0.9961	0.9962	0.9963	0.9964
2.7	0.9965	0.9966	0.9967	0.9968	0.9969	0.9970	0.9971	0.9972	0.9973	0.9974
2.8	0.9974	0.9975	0.9976	0.9977	0.9977	0.9978	0.9979	0.9979	0.9980	0.9981
2.9	0.9981	0.9982	0.9982	0.9983	0.9984	0.9984	0.9985	0.9985	0.9986	0.9986
3.0	0.9987	0.9987	0.9987	0.9988	0.9988	0.9989	0.9989	0.9989	0.9990	0.9990
3.1	0.9990	0.9991	0.9991	0.9991	0.9992	0.9992	0.9992	0.9992	0.9993	0.9993
3.2	0.9993	0.9993	0.9994	0.9994	0.9994	0.9994	0.9994	0.9995	0.9995	0.9995
3.3	0.9995	0.9995	0.9995	0.9996	0.9996	0.9996	0.9996	0.9996	0.9996	0.9997
3.4	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9998

Confidence Intervals

- Under certain conditions, our estimator $\hat{\tau}$ is approx normal:

$$\hat{\tau} \approx \mathcal{N}\left(\tau,\ \mathrm{Var}(\hat{\tau})\right)$$

- Thus, it follows that:

$$Z\ =\ \frac{\hat{\tau} - \tau}{SE(\hat{\tau})} \sim \mathcal{N}(0,1)$$

Confidence Intervals (cont'd)

- We can define a value $z_{1-\alpha/2}$ such that:

$$P\left(\left| \frac{\hat{\tau} - \tau}{SE(\hat{\tau})} \right| \leq z_{1-\alpha/2} \right) \approx 1 - \alpha$$

- Now we isolate τ:

$$P\Big(\hat{\tau} - z_{1-\alpha/2} \times SE(\hat{\tau}) \leq \tau \leq \hat{\tau} + z_{1-\alpha/2} \times SE(\hat{\tau})\Big) = 1 - \alpha\,.$$

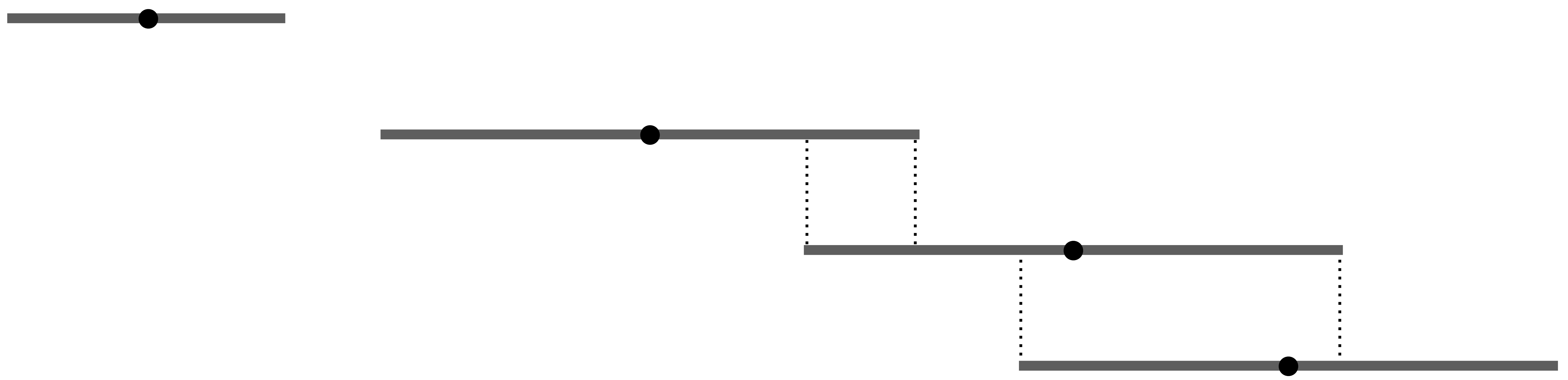
- For the sig. level $\alpha = 0.05$, we have $z_{0.975} = 1.96$.

How to Look Smart with CIs

- To quickly approximate a 95% CI for an estimate $\hat{\tau}$
    1. Compute the estimate $\hat{\tau}$ (your "center").
    2. Compute the standard error $SE(\hat{\tau})$ (your "uncertainty").
    3. Report: $\hat{\tau} \pm 1.96\, SE(\hat{\tau})$.

- If 95% CI *includes 0* $\rightarrow H_0 : \tau = 0$ would have $p \geq 0.05$.

How to Look Smart with CIs (cont'd)



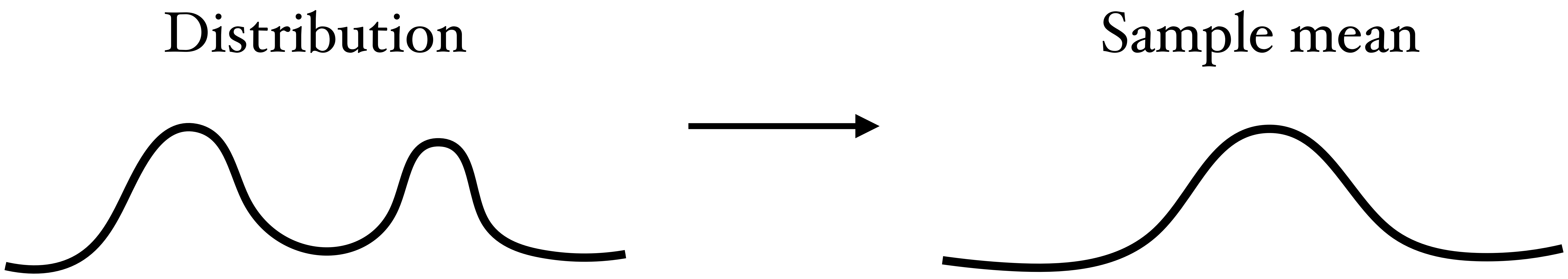
- If two CIs don't overlap, the difference between the two estimates is statistically significant.

- If two CIs overlap, the diff. between the two estimates is statistically significant if the overlap is $< 50\%$.

You have been lied to!

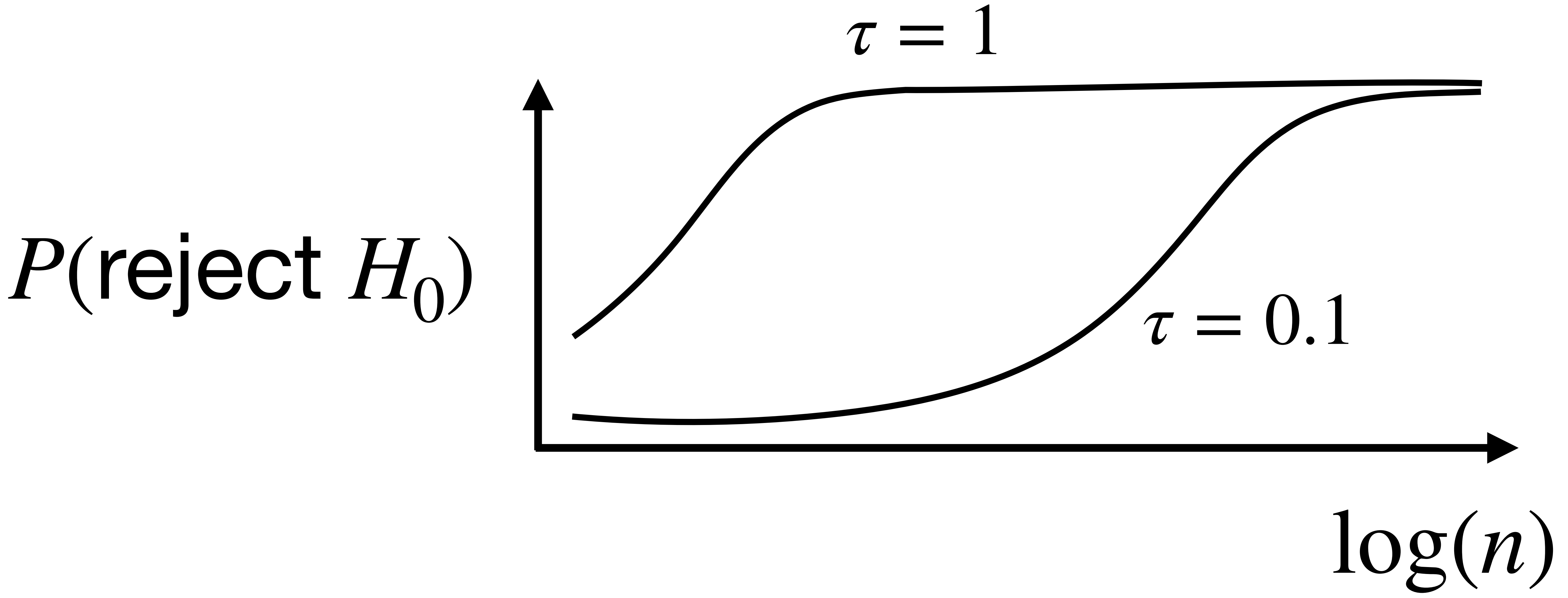
Common Misunderstandings

- I can't use Z tests if the data is not normal.
- For large n, the sample mean is $\sim$ normal per CLT.



Common Misunderstandings (cont'd)

- If $p = 0.03$ the effect is stronger than if $p = 0.02$
- Standard errors shrink like $1/\sqrt{n}$, with large enough n you can make very tiny non-zero effects statistically significant.



Common Misunderstandings (cont'd)

- A p-value is the probability that the null is true.
- If $p > 0.05$ there's no effect.
- If $p \leq 0.05$ the effect is "important."
- A 95% CI = 95% chance the true parameter is inside.
- Standard error *vs. s*tandard deviation

A problem with our setup

- **Let's assume that each evaluation attempt is *iid*.**
- Considering $a \in \{0,1\}$, we can define:

$$s_a^2 = \frac{1}{n_1 - 1} \sum_{(i,r):T_{ir}=a} \left(Y_{ir} - \bar{Y}_a\right)^2$$

$$SE(\hat{\tau}) = \sqrt{\frac{s_1^2}{|\mathscr{D}_1|} + \frac{s_0^2}{|\mathscr{D}_0|}}\,.$$

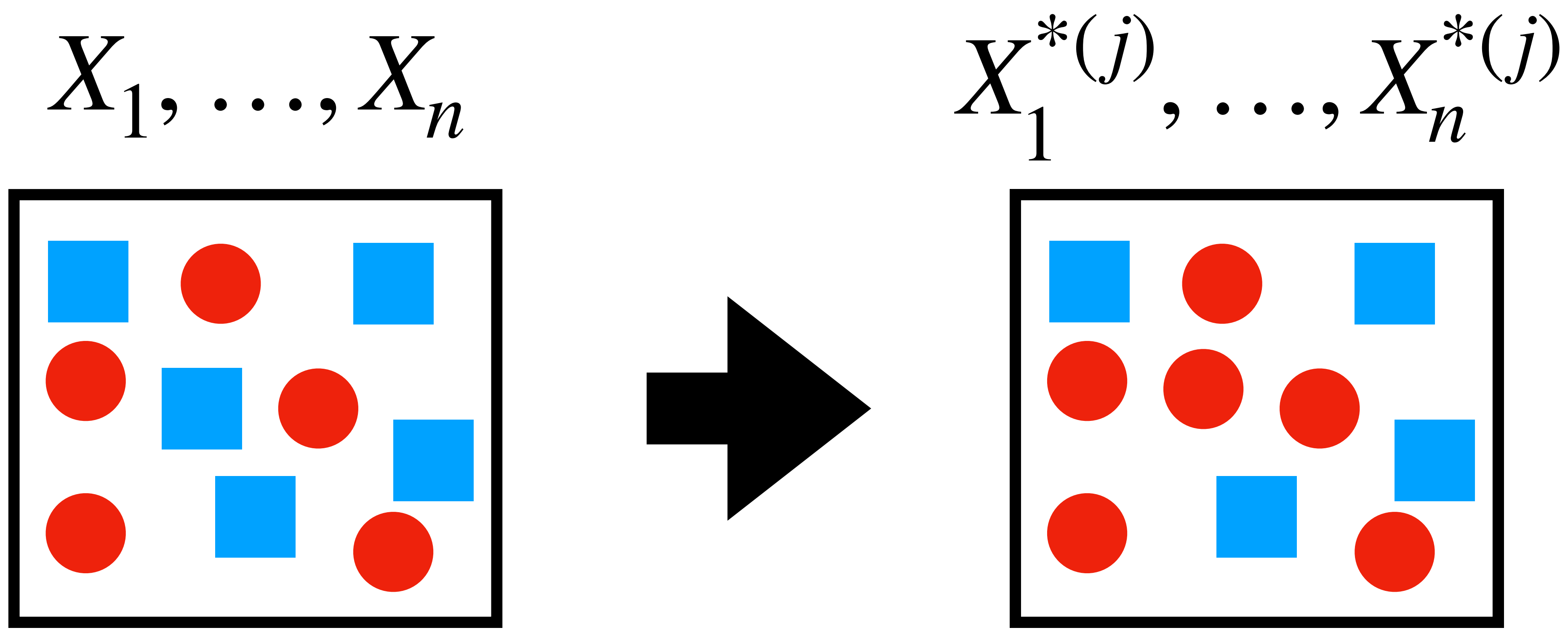
A problem with our setup (Cont’d)

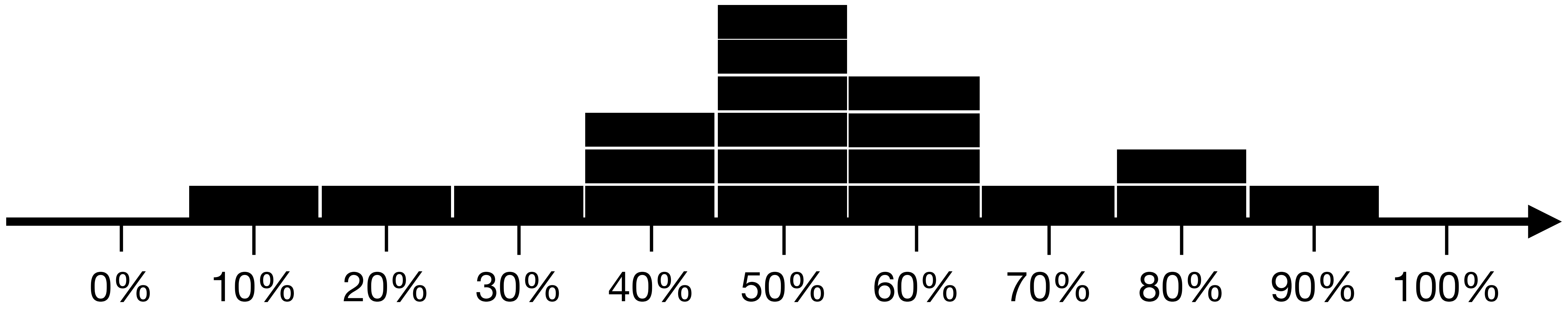
- In our running example:
  - Multiple runs of the same problem are not *iid.*
- **One easy solution**: take the average per problem before doing the confidence estimate.
- **Problem**: this does not account for variance within each problem (“cluster”).
- We will see how to handle this in regression later.

Honorable Mention

Bootstrapping Confidence Intervals

- Let:
  - $X_1, \ldots, X_n$ be some data.
  - $\hat{\theta}_n = t(X_1, \ldots, X_n)$ some statistic.
- For $j = 1, \ldots, B$
  - Sample n units with replacement.
  - Estimate $\hat{\theta}_n^{*(j)}$.
  - "Store" $\hat{\theta}_n^{*(j)}$ into a sorted array.
- Get the quantiles of the obtained distribution





Bootstrap is magic

- Allows you to get confidence intervals for any statistic.
    - Medians, quantiles, ratios, weird nonlinear stats.
- Captures skew and asymmetry.
- Minimal assumptions.
- You can resample the right unit (*e.g.*, clusters, subjects, blocks), so dependence is handled naturally.