PSYCH 201B

Statistical Intuitions for Social Scientists

The Mechanics of Statistical Inference

01/26/2026

Quick Announcements

- Website updates:
  - **Schedule** updated
  - **Glossary** of terms we've covered so far
- Homework 2 will be available after *tomorrow's* lab
  - **Deadline:** Midnight Monday Feb 2
  - **This time:** we'll indicate point values for each question and give you a numerical score
  - **Remember:** You can still push updates after we review next week to improve you score!
- **Next Week:**
  - Monday Feb 2: **No class**
  - **Tues Feb 3: Yes class**
  - **Web Feb 4: Yes class**

Today's Plan

1. Recap

2. We're gonna watch and discuss a video!

Last time…

The Bias-Variance Tradeoff

- **Bias**
  - Error from simplifying assumptions
  - How much a predictions are wrong *on average*

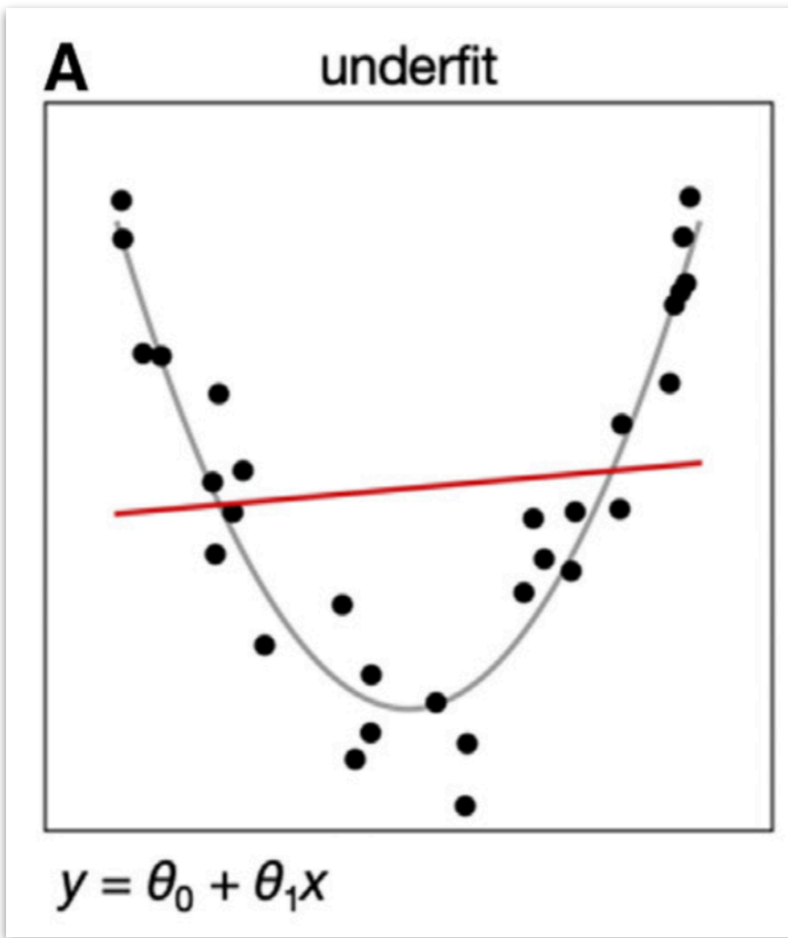
$$\text{Bias}(\hat{Y}) = E(\hat{Y}) - Y$$

- **Variance**
  - Error from data sensitivity
  - How much predictions change across samples

$$\text{Variance} = E[(\hat{Y} - E[\hat{Y}]^2]$$

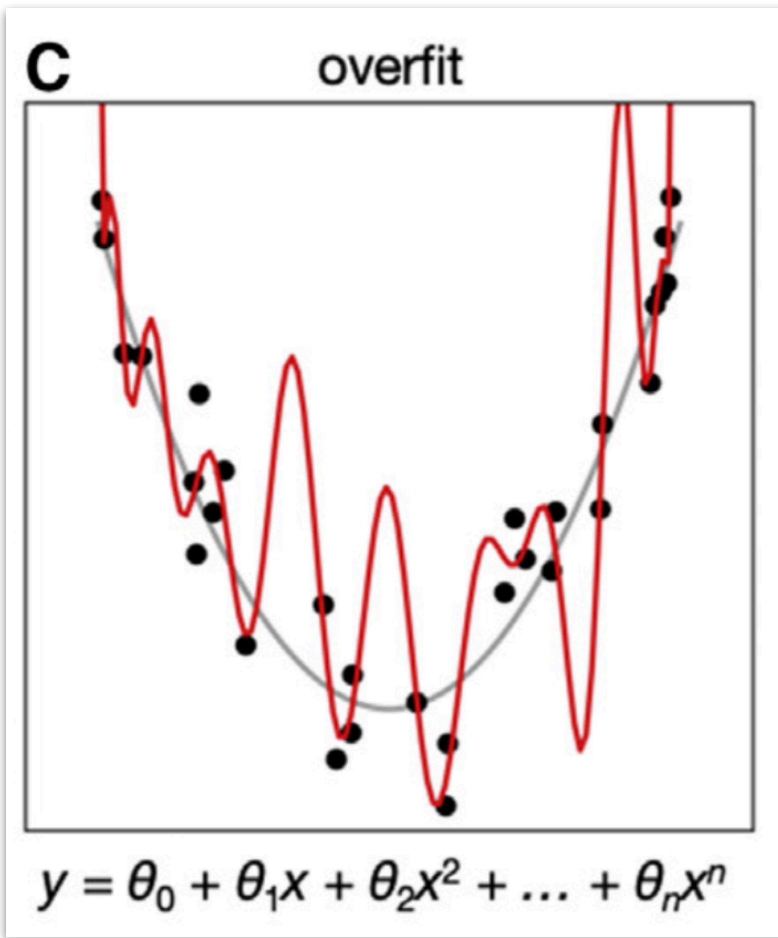
The Bias-Variance Tradeoff

- **High Bias - Low Variance**
  - Behaves very **consistently**
  - By **under-fitting** the data (generalizes poorly to new data)

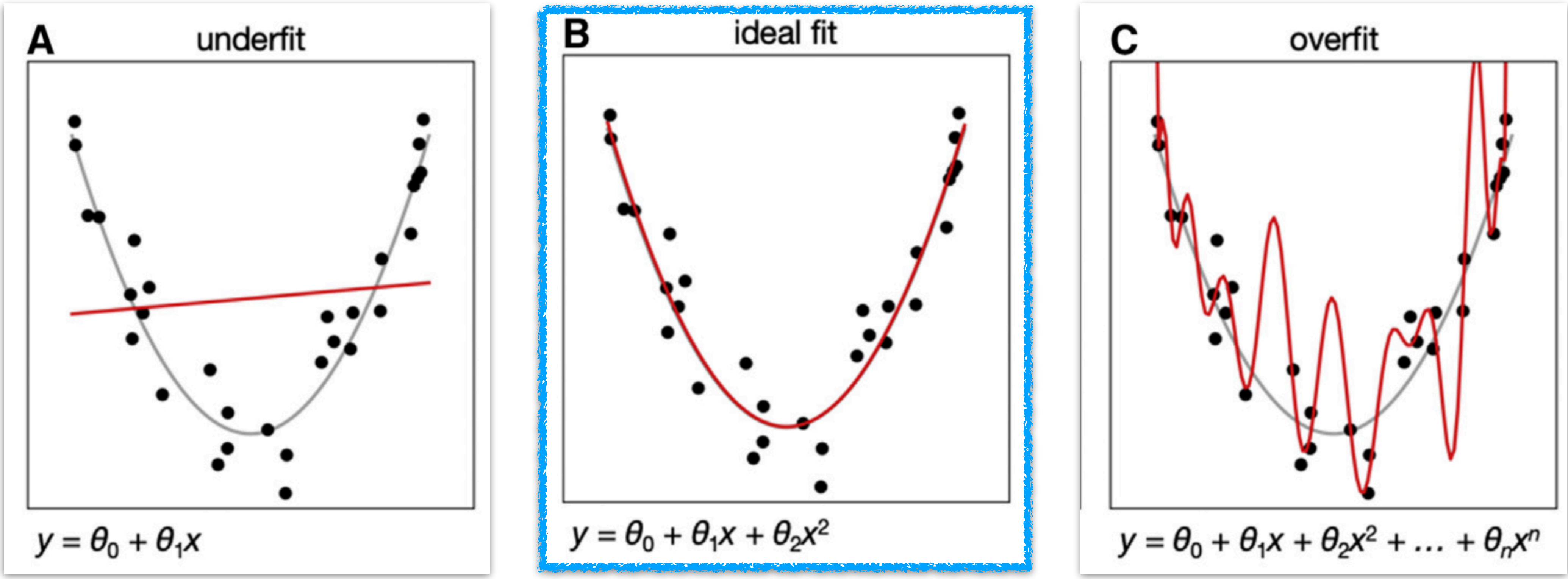


The Bias-Variance Tradeoff

- **Low Bias - High Variance**
  - Over-fitting data (memorize real patterns & noise)
  - Highly variable predictions (too sensitive to data)



The Bias-Variance Tradeoff



Hypothesis testing as model comparison

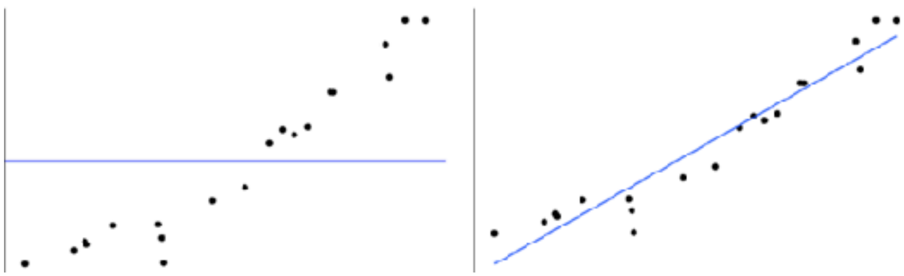
```
Error = Data - Model
```

- We build a model with parameters and estimate their values by minimizing error given data
- Adding additional parameters will **always improve model fit** (reduce error further)
- So there's a fundamental trade-off between complexity and accuracy = worth it?

We perform hypothesis tests by comparing model errors

PRE = Proportional Reduction in Error

Which model fits data best?



Compact model

model$_C$: $\hat{Balance_i} = \beta_0 + \epsilon$

Augmented model

model$_A$: $\hat{Balance_i} = \beta_0 + \beta_1 * Income_i + \epsilon$

$$\text{ERROR}(C) \geq \text{ERROR}(A)$$

Proportional reduction in error (PRE)

$$\text{PRE} = \frac{\text{ERROR}(C) - \text{ERROR}(A)}{\text{ERROR}(C)}$$

The **worth it?** question

intercept

1 parameter → compact: $\hat{Balance_i} = \beta_0 + \epsilon$

worth it?

intercept

2 parameters → augmented $\hat{Balance_i} = \beta_0 + \beta_1 * Income_i + \epsilon$

slope (income)

Decide whether it's **worth it**

- Do the additional parameter(s) reduce enough error?
- Have we used up too many degrees-of-freedom by estimating these additional parameter(s)?

We perform hypothesis tests by comparing model errors

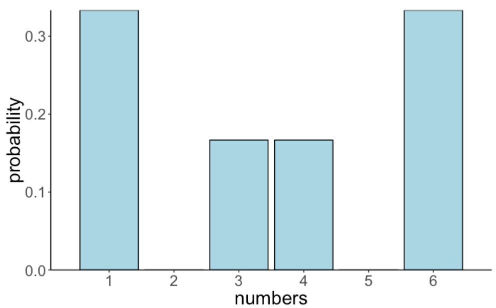
PRE = Proportional Reduction in Error

But PRE is an estimator of an **unknown** true distribution!

We need a sampling distribution of PRE:

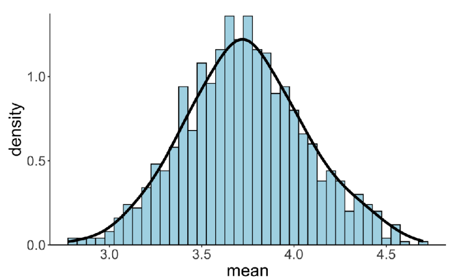
aka *how much PRE would change* if we collected a new dataset (sample)

Reminder: 3 main distributions we work with



Population distribution

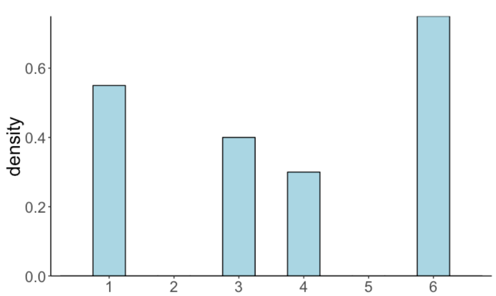
So we can make a claim about this



Sampling distribution

But we need to know this…

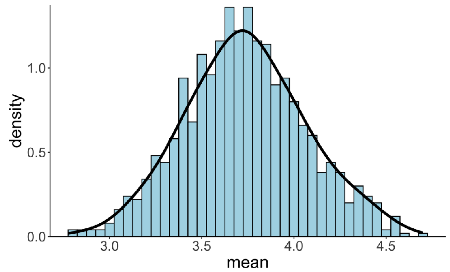
if only there was a distribution that already does this…



Sample distribution

We calculated PRE using this

Reminder: 3 main distributions we work with



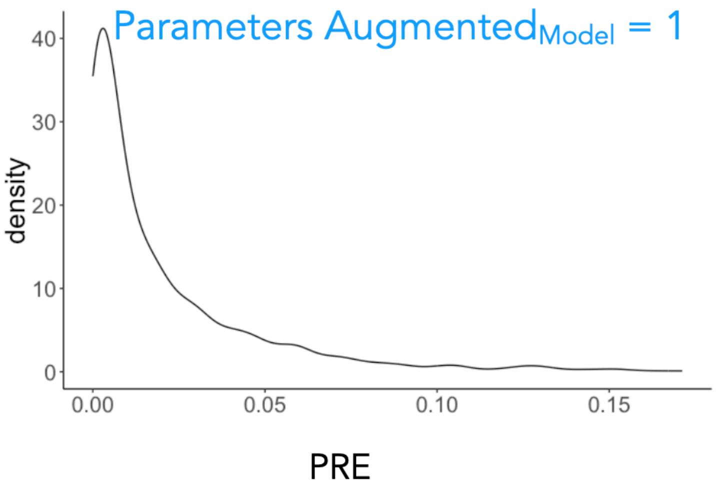
Sampling distribution

F-distribution = sampling distribution of PRE

sampling distribution of PRE

Parameters $\text{Compact}_{\text{Model}} = 0$

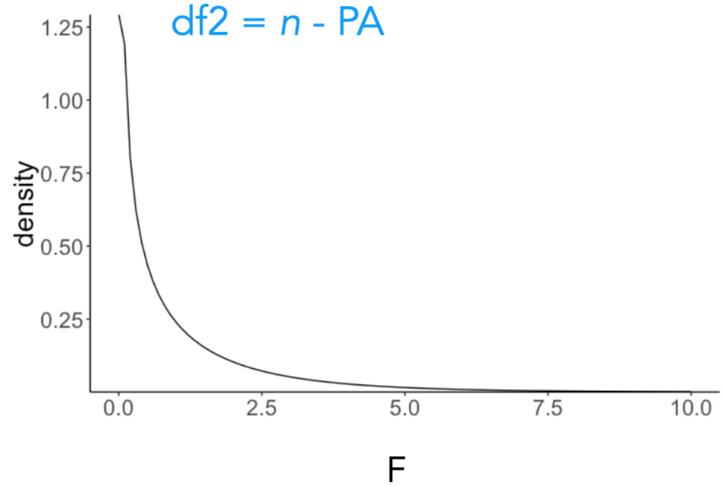
Parameters $\text{Augmented}_{\text{Model}} = 1$



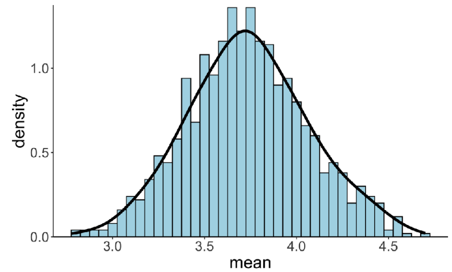
***F*(df1, df2) distribution**

df1 = PA - PC

df2 = *n* - PA

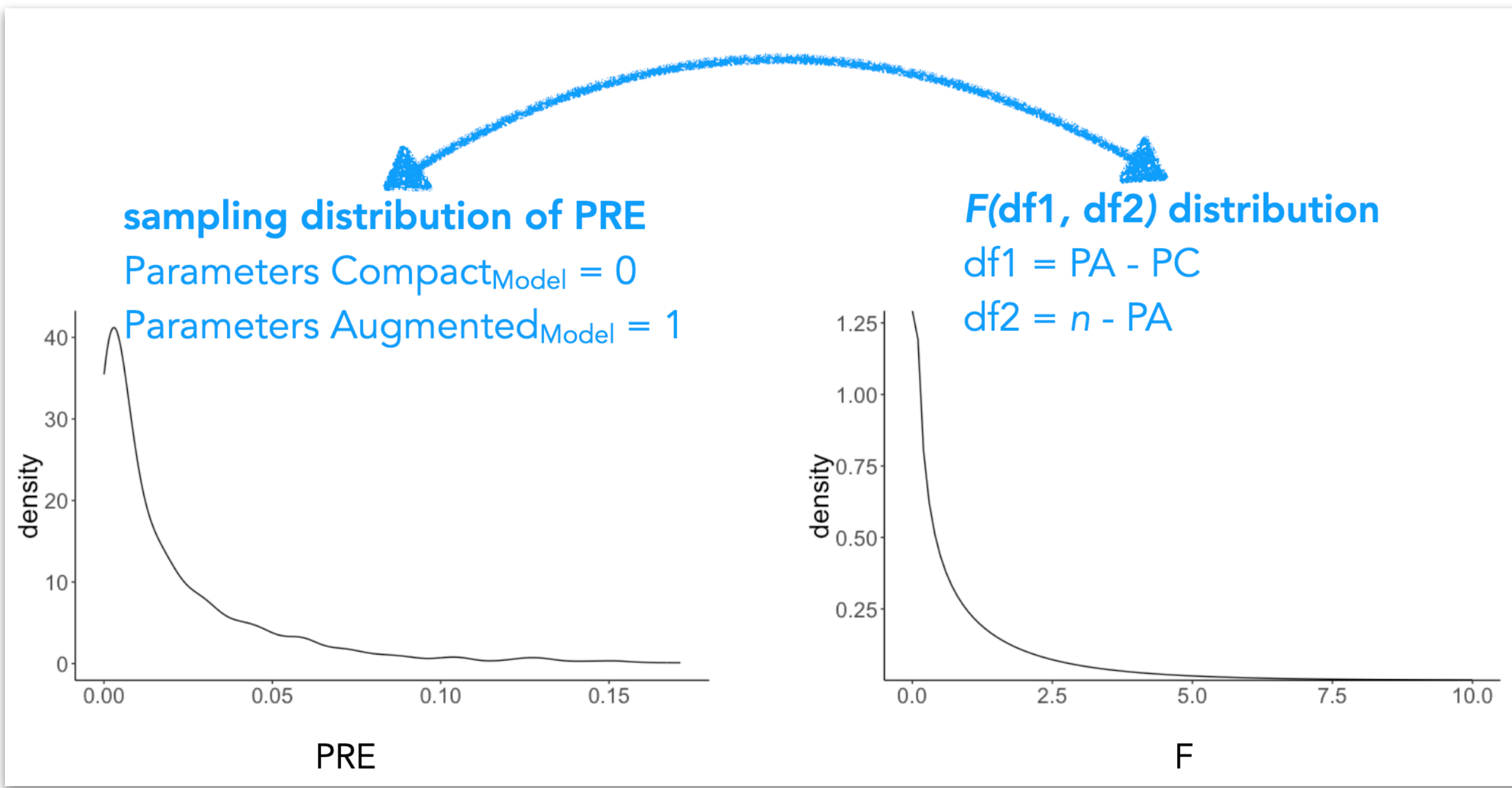


We quantify uncertainty with sampling distributions



Sampling distribution

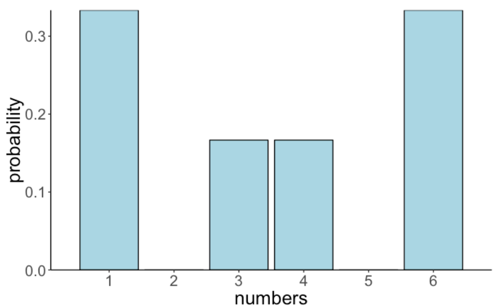
F-distribution = sampling distribution of PRE



The four fundamental intuitions

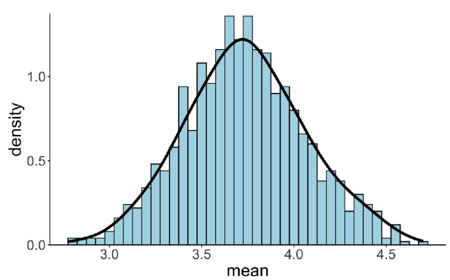
- **Aggregation**
  - Gives us a model
- **Sampling**
  - Tells us where it applies
- **Uncertainty**
  - Keeps us honest
- **Learning**
  - Forces iteration

We quantify uncertainty with sampling distributions



The value of the estimator we *hope* we calculated but can never know directly

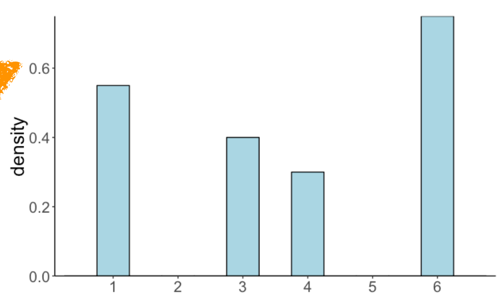
Population distribution



The uncertainty bridge (standard-error, confidence intervals)

how much our estimator varies across different data

Sampling distribution



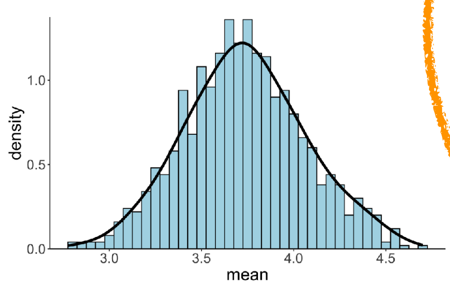
The data we used to fit our estimator

Sample distribution

We quantify uncertainty with sampling distributions

Population distribution

What R or Python calculate for you when you use: `t.test()`, `lm()` ...



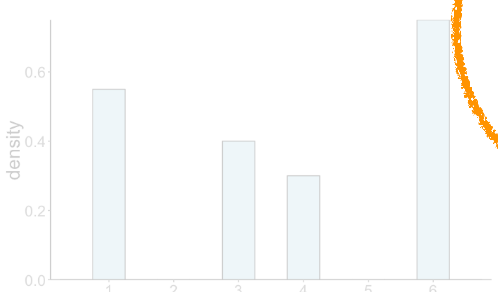
Sampling distribution

The uncertainty bridge (standard-error, confidence intervals)

how much our estimator varies across different data

Approach (1) Asymptotic (we look it up from a formula)

Approach (2): We build it ourselves via resampling



Sample distribution

But math is hard and formulas don't build intuitions, so instead....

Programming as theory-building

PETER NAUR, PROGRAMMING AS THEORY BUILDING

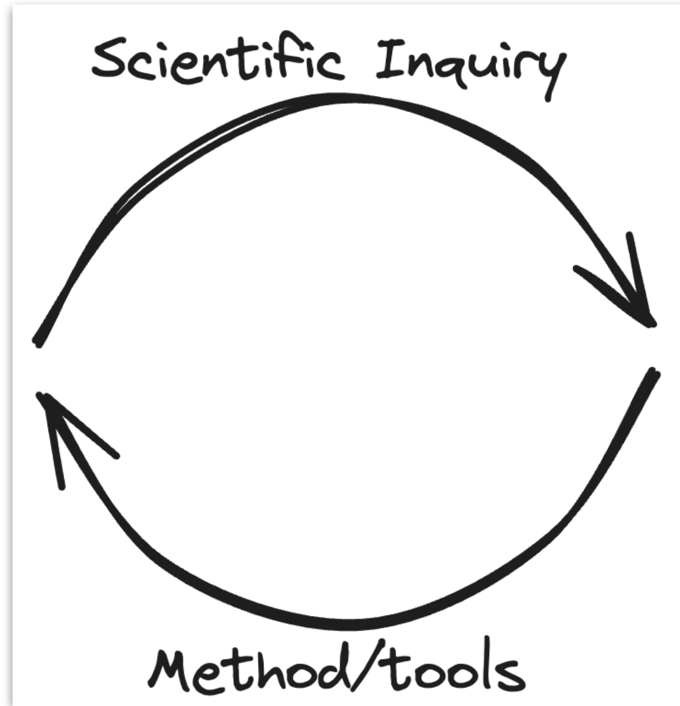
Peter Naur, widely known as one of the authors of the programming language syntax notation "Backus-Naur Form" (BNF), wrote "Programming as Theory Building" in 1985. It was reprinted in his collection of works, *Computing: A Human Activity* (Naur 1992).

This article is, to my mind, the most accurate account of what goes on in designing and coding a program. I refer to it regularly when discussing how much documentation to create, how to pass along tacit knowledge, and the value of the XP's metaphor-setting exercise. It also provides a way to examine a methodol-

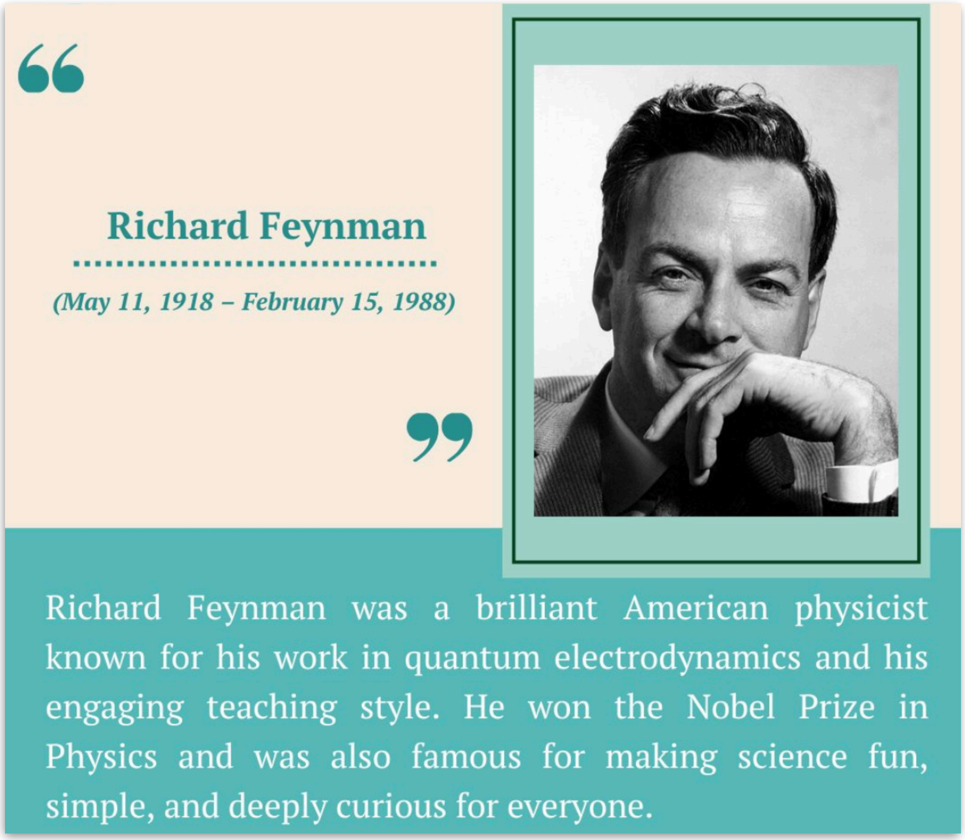
"PROGRAMMING AS THEORY BUILDING"

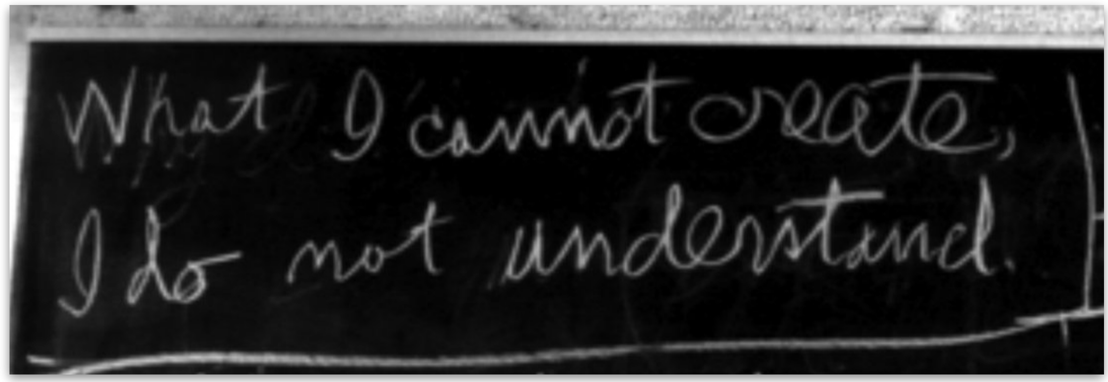
Introduction

The present discussion is a contribution to the understanding of what programming is. It suggests that programming properly should be regarded as an activity by which the programmers form or achieve a certain kind of insight, a theory, of the matters at hand. This suggestion is in contrast to what appears to be a more common notion, that programming should be regarded as a production of a program and certain other texts.



Programming as theory-building

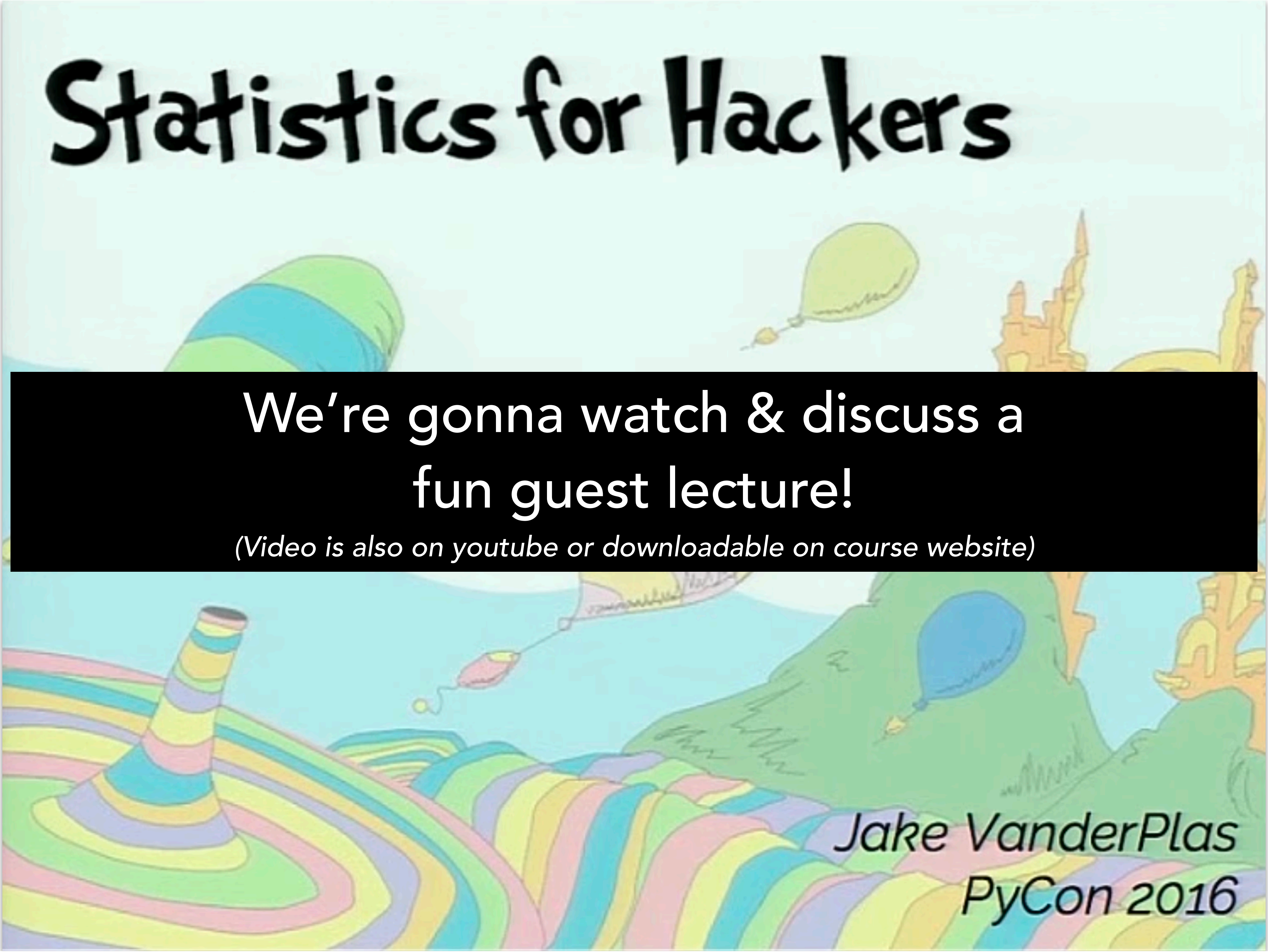




"What I cannot create, I do not understand."

~ Richard Feynman

Nobel in Physics


Statistics for Hackers
We’re gonna watch & discuss a
fun guest lecture!
(Video is also on youtube or downloadable on course website)
Jake VanderPlas
PyCon 2016