Three Methods for Statistical Inference

Ben Baumer

JSM Baltimore

August 1st, 2017

(https://github.com/mine-cetinkaya-rundel/novel-first-stat-ds-jsm2017)

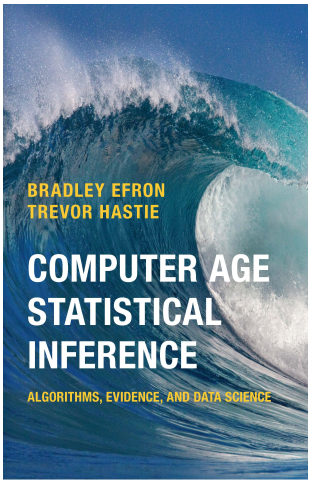
Why am I talking about inference?

Inference

- ASA Statement on P-Values and Statistical Significance
- Symposium on Statistical Inference

It's not just me...

- Brad Efron & Trevor Hastie



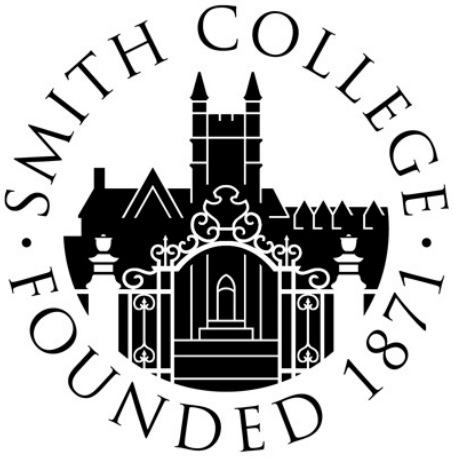
- Jon Wellner: Teaching Statistics in the Age of Data Science

The shoulders of giants

- Katherine Halvorsen
- Nick Horton
- George Cobb
- Andrew Bray



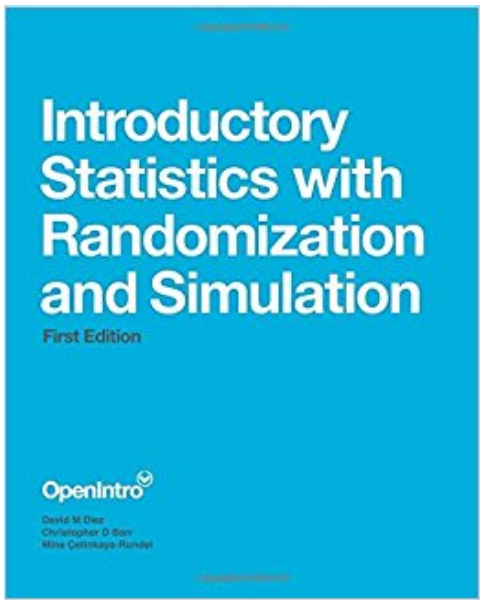

About Smith



- liberal arts college for women
- new major in Statistical & Data Sciences
  - $\geq$ *3* courses in statistics
  - $\geq$ *2* courses in programming
  - $\geq$ *1* course in data science
- no modeling or inference in data science

SDS 220

- most advanced intro stats course
- calc or discrete math pre-req
- 5 credits: 3 lectures, 1 lab
- mostly STEM majors
    - 40% EGR and BIO majors



Three Methods to construct null distribution

1. Simulation: use computer to *simulate*
2. Exact: use math to *compute*
3. Approximation: use statistical theory to *approximate*

When I took stats

1. ~~Simulation~~
2. ~~Exact~~ (in probability class)
3. Approximation

An example: one proportion

Method #1: simulation

```
library(tidyverse)

outcomes <- data_frame(candidate = c("clinton", "trump"))
p_0 <- 1/2

# http://www.cnn.com/election/results/exit-polls
n <- 246

sim <- outcomes %>%
  oilabs::rep_sample_n(size = n, replace = TRUE, reps = 10000) %>%
  group_by(replicate) %>%
  summarize(N = n(),
            clinton_votes = sum(candidate == "clinton")) %>%
  mutate(clinton_pct = clinton_votes / N)
```

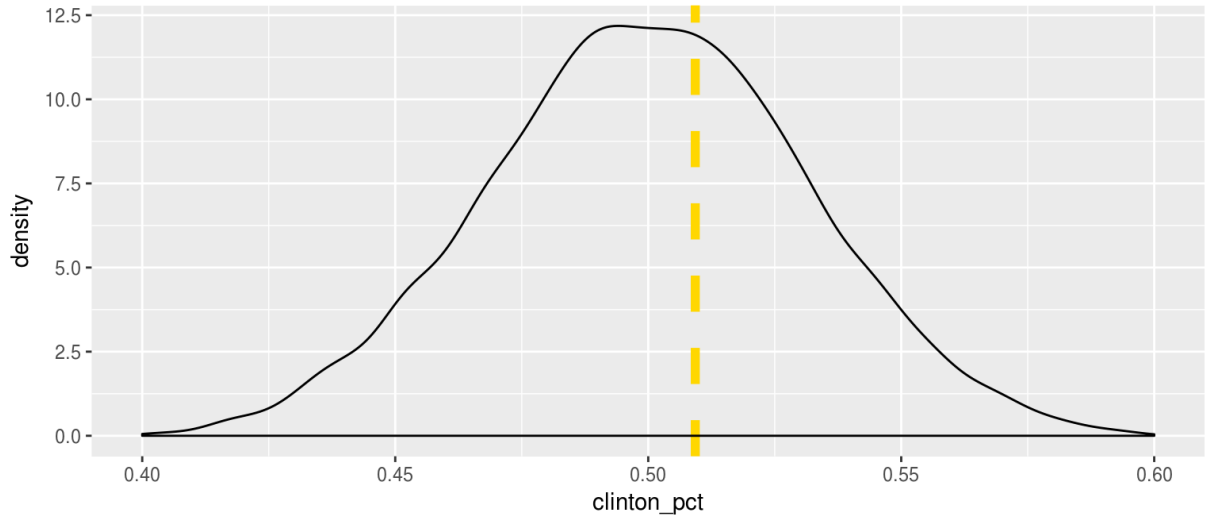

Method #1: simulation, observed

```
# observed proportion
p_hat <- data_frame(clinton_pct = 0.5092953)

# what we know from data
observed <- ggplot(data = p_hat) +
  geom_vline(aes(xintercept = clinton_pct),
             color = "gold", size = 2, linetype = 2) +
  scale_x_continuous(limits = c(0.4, 0.6))
```

Method #1: simulation, plot

```
observed +
  geom_density(data = sim, aes(x = clinton_pct))
```

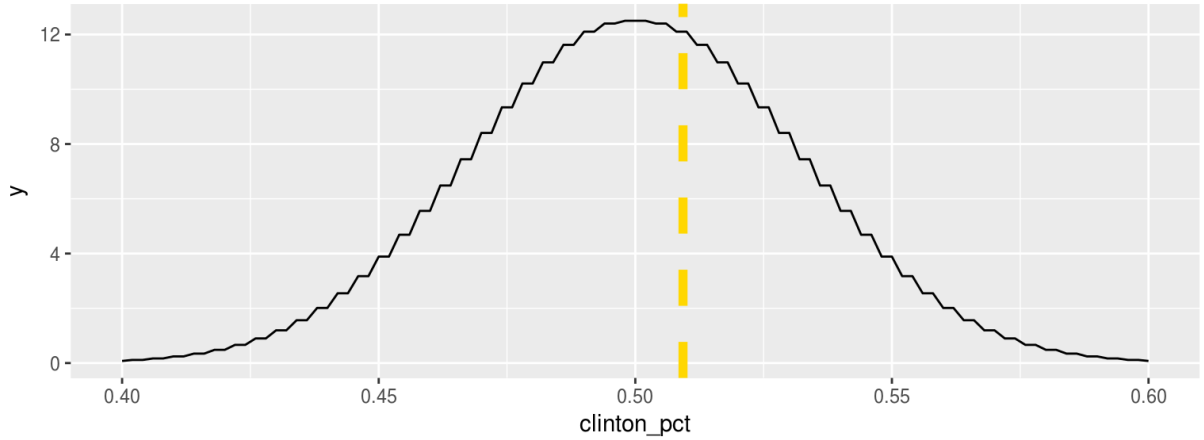


Method #2: exact

- Let $X \sim Bernoulli(p_0)$, then
  - $\mathbb{E}[X] = p_0, \qquad Var[X] = p_0(1 - p_0)$
- Let $Z = \frac{X_1 + \cdots + X_n}{n}$, then
  - $\mathbb{E}[Z] = p_0, \qquad Var[Z] = \frac{p_0(1-p_0)}{n}$
  - for later, $sd(Z) = \sqrt{\frac{p_0(1-p_0)}{n}}$

Method #2: exact, plot

```
dbinom_p <- function (x, size, prob, log = FALSE) {
  n * dbinom(round(x * size), size, prob, log)
}
observed +
  stat_function(fun = dbinom_p, args = c(size = n, prob = p_0))
```



Method #3: approximation

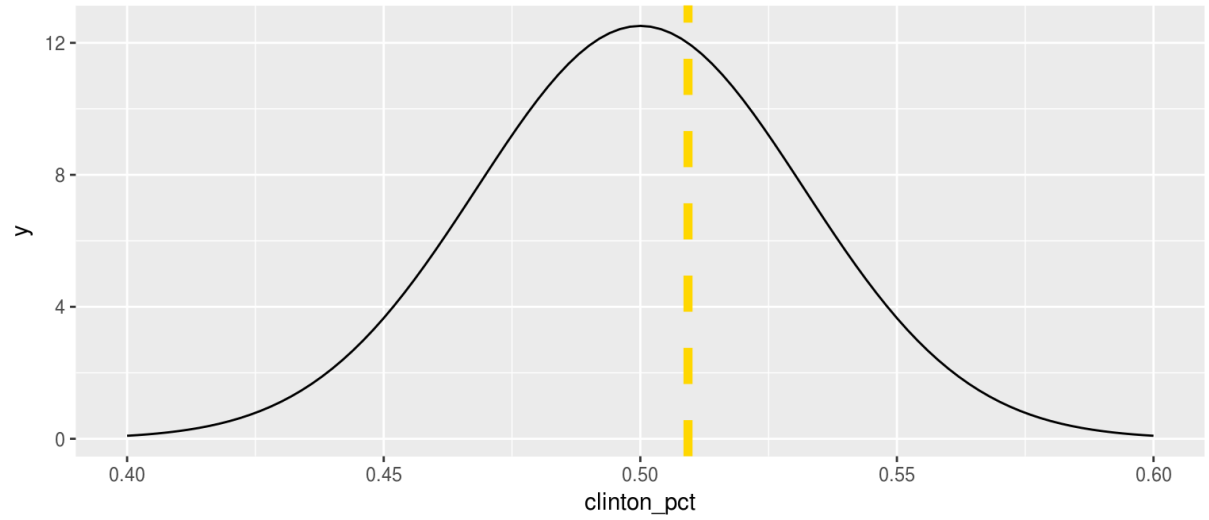
- For $np > 10$ and $n(1-p) > 10$,

$$\frac{1}{n} \cdot Binomial(n, p_0) \approx Normal\left(p_0, \sqrt{\frac{p_0(1-p_0)}{n}}\right)$$

```
se_p0 <- sqrt(p_0 * (1-p_0) / n)
```

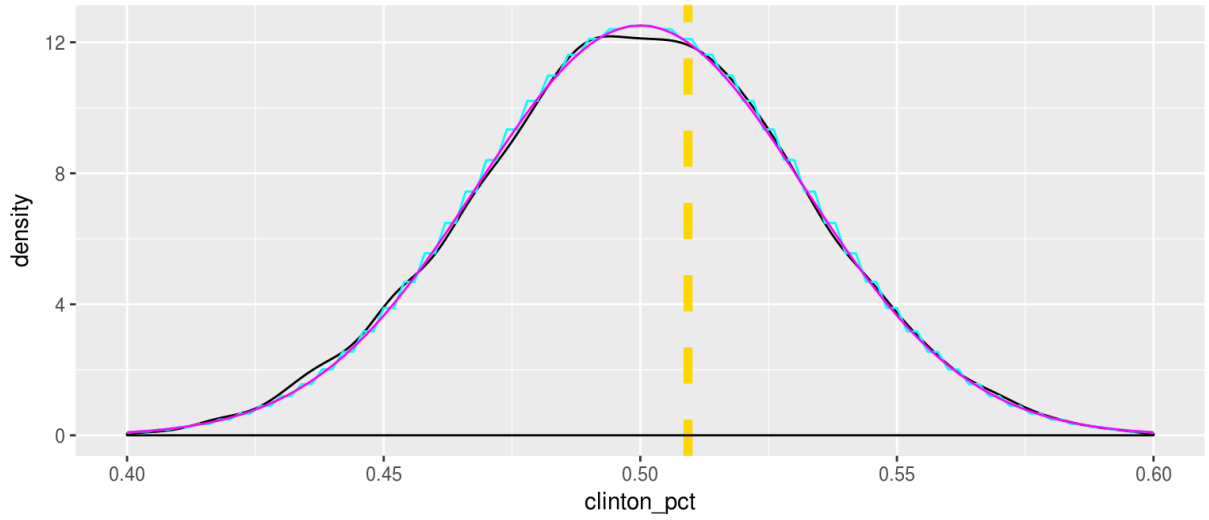
Method #3: approximation, plot

```
observed +
  stat_function(fun = dnorm, args = c(mean = p_0, sd = se_p0))
```



Three methods comparison, plot

```
observed +
  geom_density(data = sim, aes(x = clinton_pct)) +
  stat_function(fun = dbinom_p, args = c(size = n, prob = p_0), color
  stat_function(fun = dnorm, args = c(mean = p_0, sd = se_p0), color
```



Three methods comparison, table

	Simulation	**Exact**	**Approximation**
Assumptions	independence	independence probability model	independence $np > 10$, etc.
Pros	no math required flexible	exact solution	uses normal approx. usu. good no CPU required
Cons	requires computer non-deterministic	usu. HARD to derive! not always known	more assumptions not exact

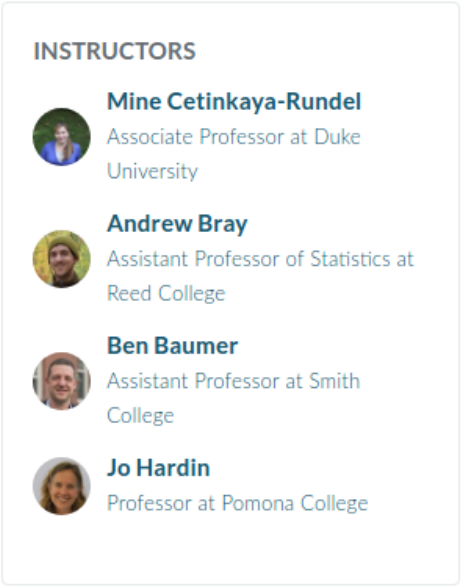
Example #2: two categorical variables

- Simulation
  - randomization test
- Exact
  - Fisher's exact test, hypergeometric
- Approximation
  - χ^2-test of independence

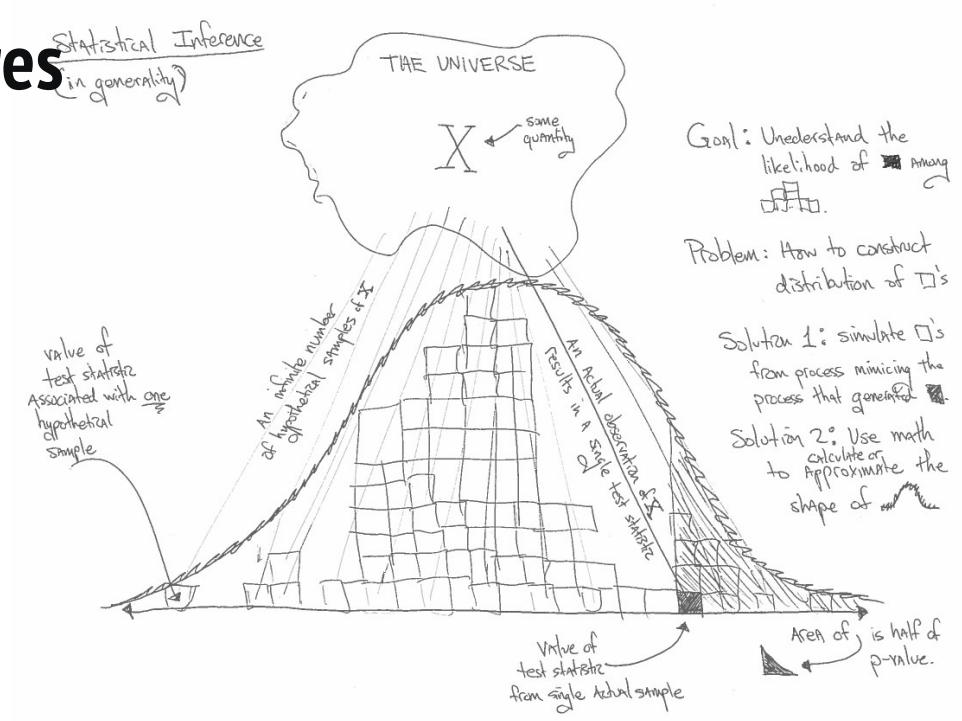
Unification?

DataCamp

- Statistics with R skills track
- 4 courses available now
- 4 more courses available soon



Pictures



Common setup with `infer`

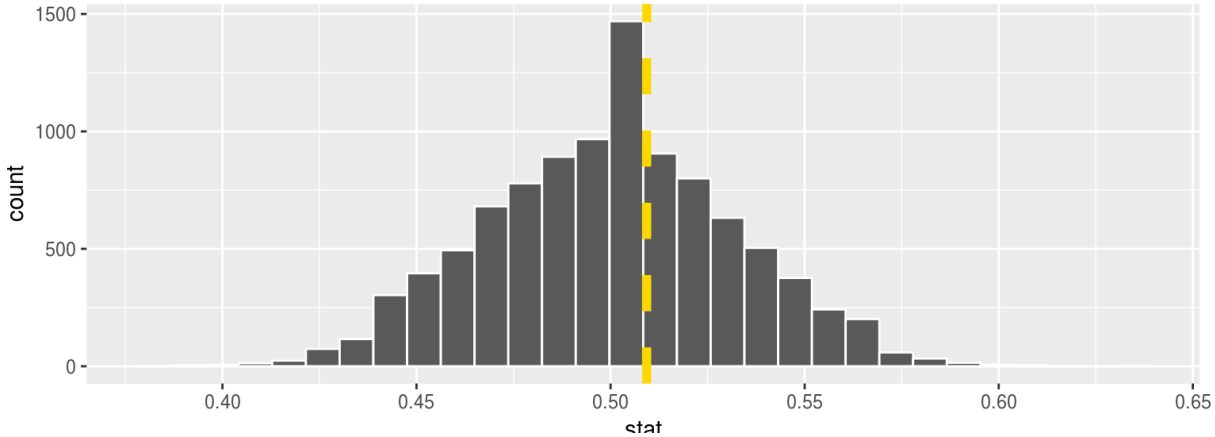
```
# devtools::install_github("andrewpbray/infer")
library(infer)

fake_null <- data_frame(
  candidate = rep(c("clinton", "trump"), each = n/2)
)

# independent of three methods
setup <- fake_null %>%
  specify(response = candidate) %>%
  hypothesize(null = "point", p = c("clinton" = 0.5, "trump" = 0.5))
```

Method #1 with `infer`

```
setup %>%
  generate(reps = 10000, type = "simulate") %>%
  calculate(stat = "prop") %>%
  visualize() +
  geom_vline(data = p_hat, aes(xintercept = clinton_pct),
             color = "gold", size = 2, linetype = 2)
```



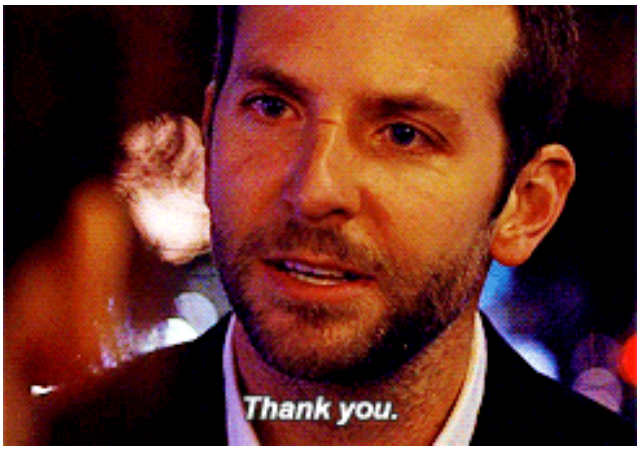
Method #2 with `infer`

```
# not implemented yet; compare to binom.test()
setup %>%
  calculate(stat = "prop") %>%
  visualize() +
  geom_vline(data = p_hat, aes(xintercept = clinton_pct),
             color = "gold", size = 2, linetype = 2)
```

Method #3 with `infer`

```
# not implemented yet; compare to prop.test()
setup %>%
  calculate(stat = "prop") %>%
  visualize() +
  geom_vline(data = p_hat, aes(xintercept = clinton_pct),
             color = "gold", size = 2, linetype = 2)
```


Thank you!



bbaumer@smith.edu

GitHub beanumber

twitter @BaumerBen