STAT 215
Using Analytic Approximations for Tests and CIs

Colin Reimer Dawson

Oberlin College

September 28, 2017

Outline

Properties of Sampling Distributions

Most (about 95%) of *simple random* samples have a sample mean ($\bar{x}$) which is within 2 Standard Errors of the population mean (μ). Therefore, about 95% of the time, the population mean will be within 2SE of the *sample* mean!

A similar statement holds for some other statistics/parameters, under a particular condition. What condition? The sampling distribution needs to be (approximately) symmetric and bell-shaped

So what's with all these bell shapes?

- Q: Why are so many distributions "bell-shaped"?
- A: The **Central Limit Theorem**
- One of the most important results in probability: for sufficiently large samples, sample means have a **Normal** (bell-shaped) **distribution**.

Sample Means Show Up A Lot

- Sample means are sample means (did you know this?)
- Sample proportions are sample means (encode binary variable as 0s and 1s)

Also Sums, Differences, Rescalings, ...

- Sum of two Normals is Normal
- Rescaling a Normal by a constant is Normal
- Difference of Normals is Normal

So...

- Differences of sample means are approximately Normal
- Differences of sample proportions are approximately Normal

Outline

P-value = Proportion of Randomized Sample Statistics

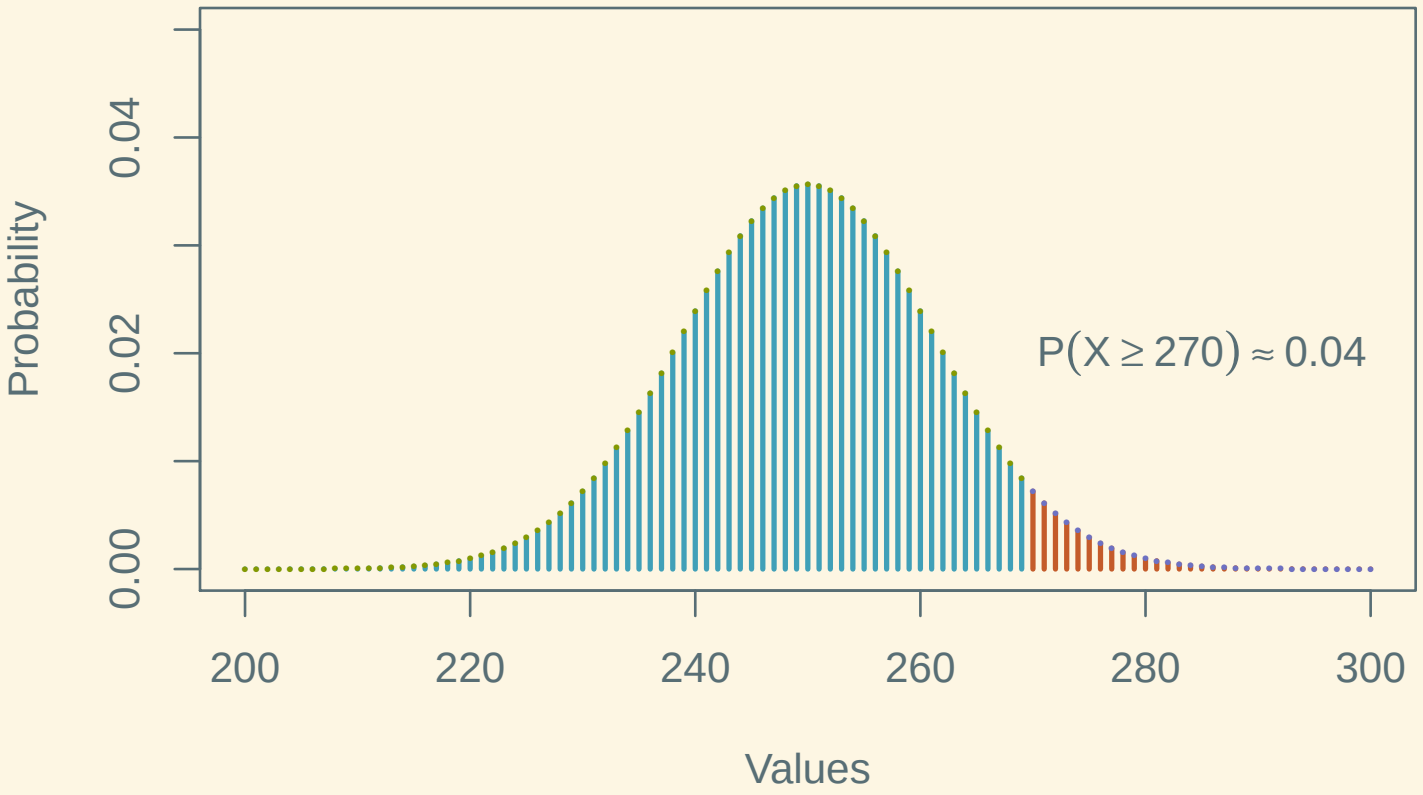


Figure: Randomization distribution for the number of heads in 500 coin flips, highlighting the one-tailed P-value testing $H_1 : p > 0.5$ for an observation of 270 heads.

Confidence Level = Proportion of Bootstrap Samples

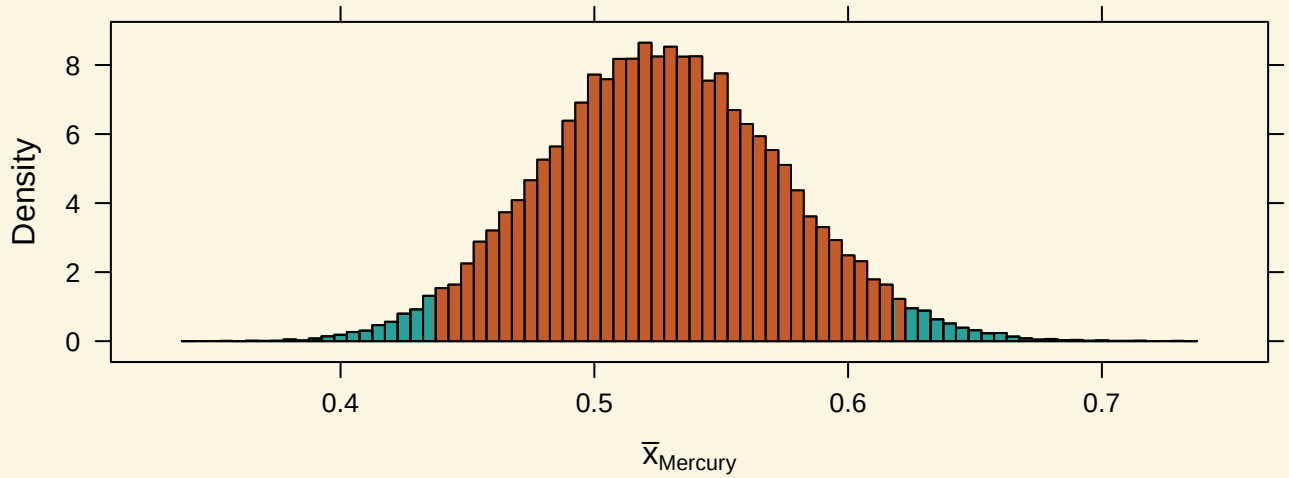


Figure: Bootstrap distribution for mean mercury level in fish in Florida Lakes (from `FloridaLakes` dataset). The middle 95% is highlighted illustrating a 95% confidence interval.

Narrowing Bins: Frequency

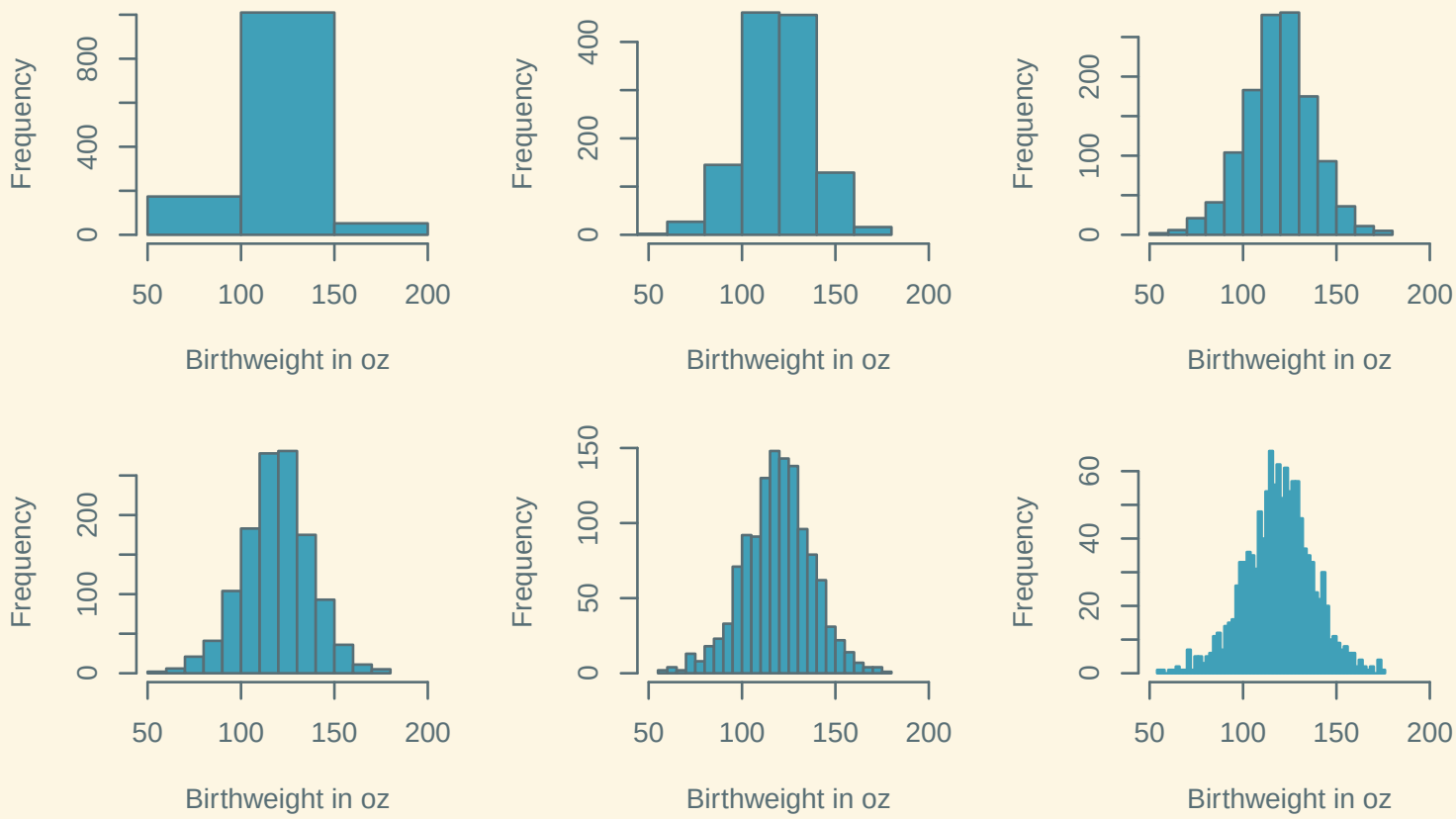


Figure: Histograms of Babies' Birth Weights (Nolan and Speed, 2000)

Density

$$\text{Proportion} = \text{Area} = \text{Height} \times \text{Width}$$

$$\text{Density} = \text{Height} = \frac{\text{Proportion}}{\text{Width}}$$

This quantity (proportion divided by width) is called "density" by analogy to physics: "amount of stuff" divided by "amount of space".

Narrowing Bins: Density

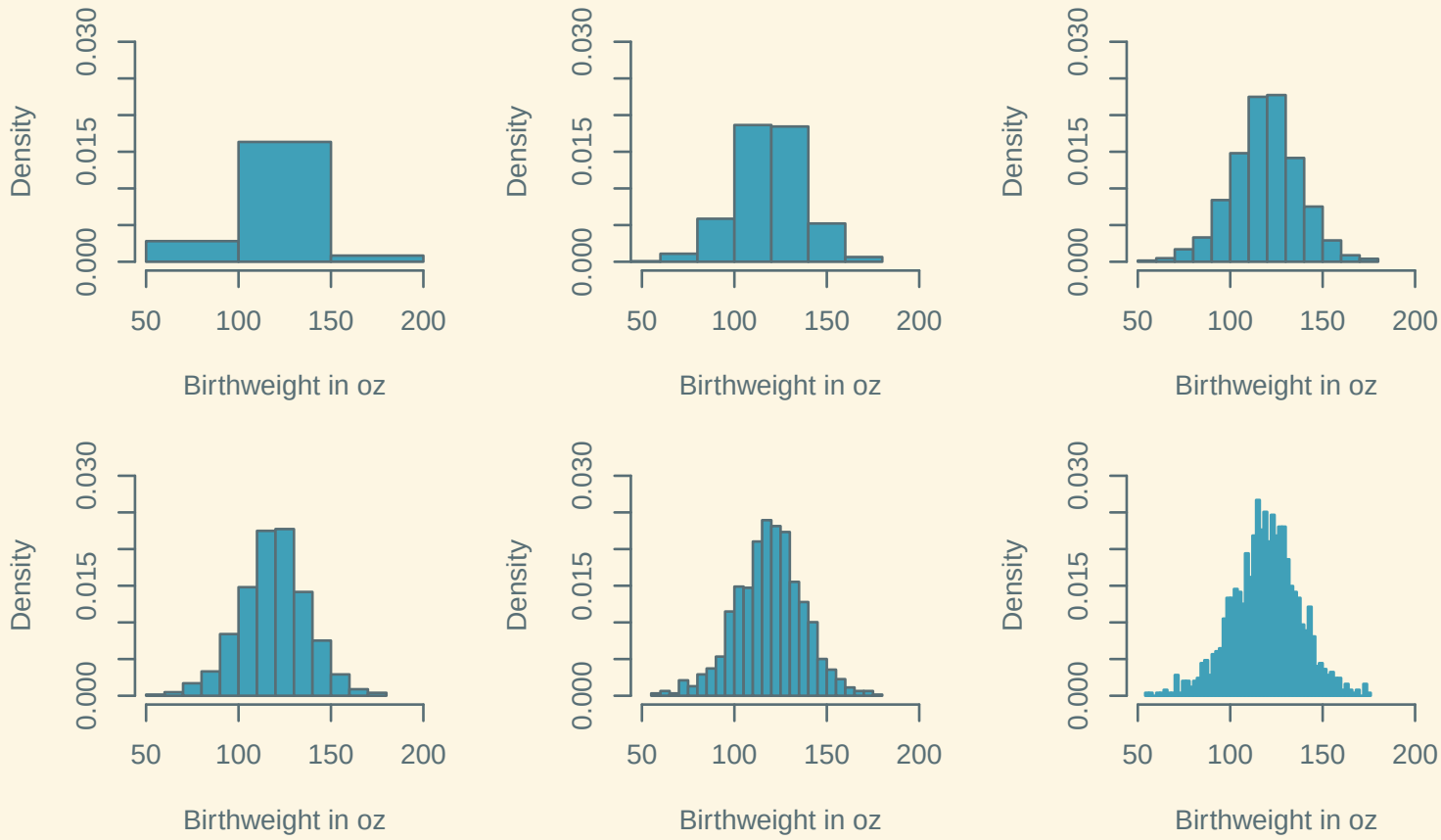


Figure: Histograms of Babies' Birth Weights (Nolan and Speed, 2000)

Density Functions

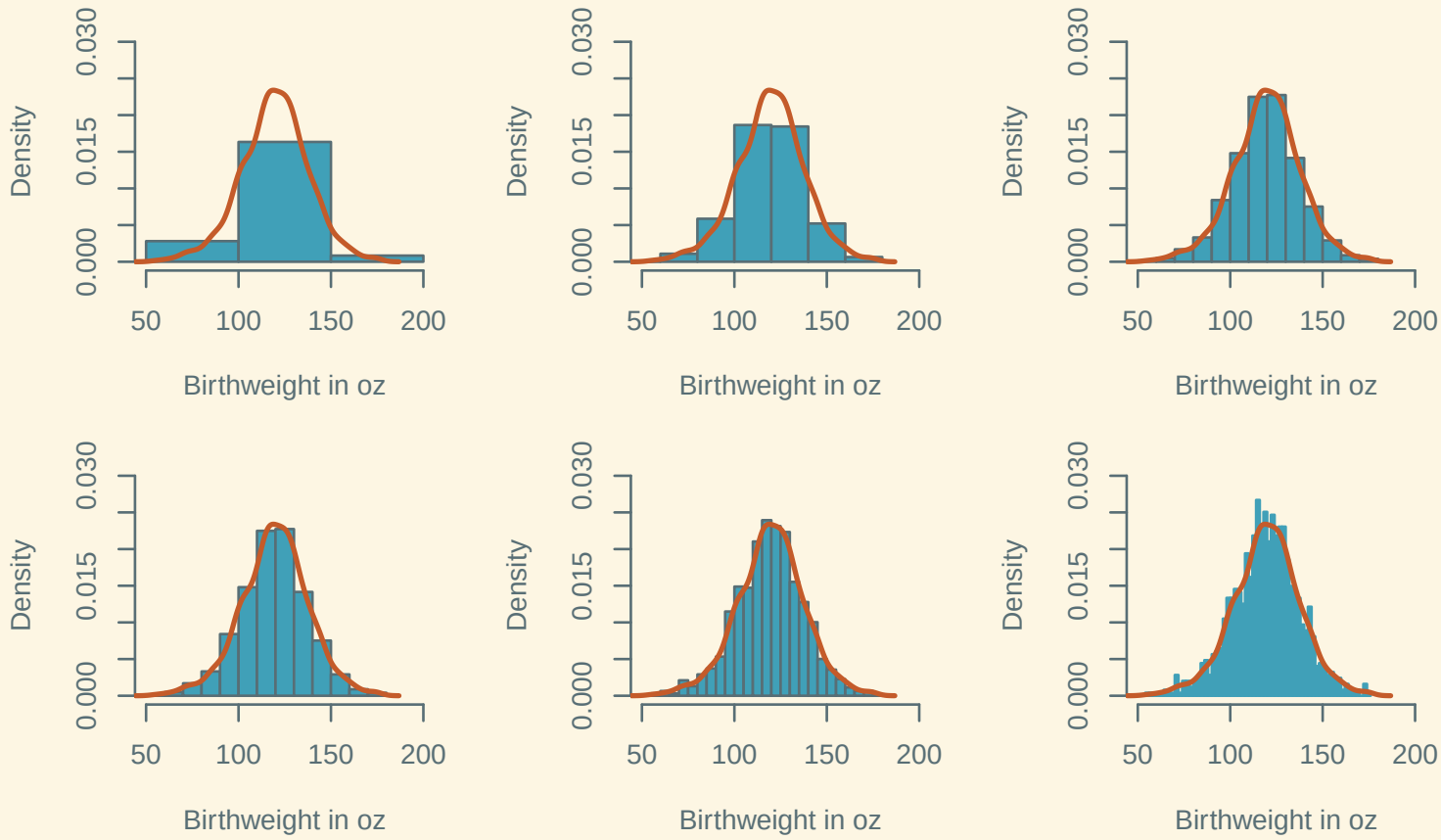


Figure: Densities of Babies' Birth Weights (Nolan and Speed, 2000)

Proportion = Area Under the Density Curve

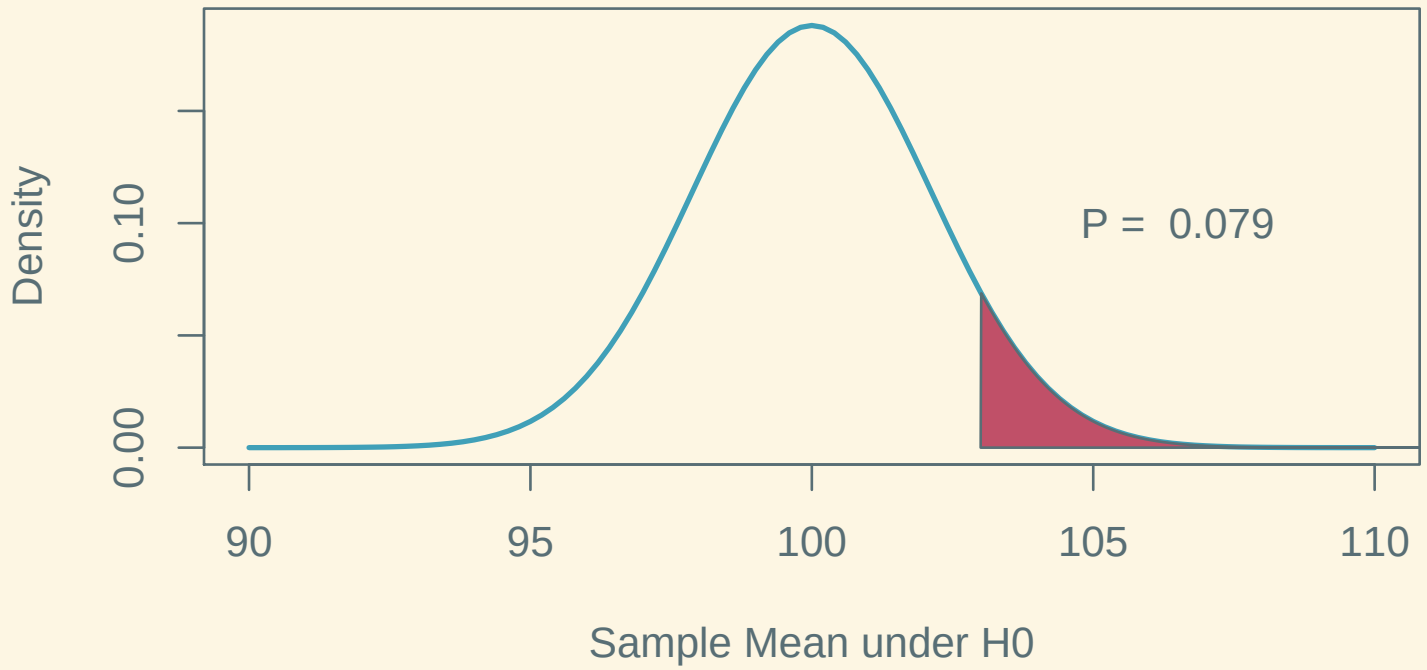
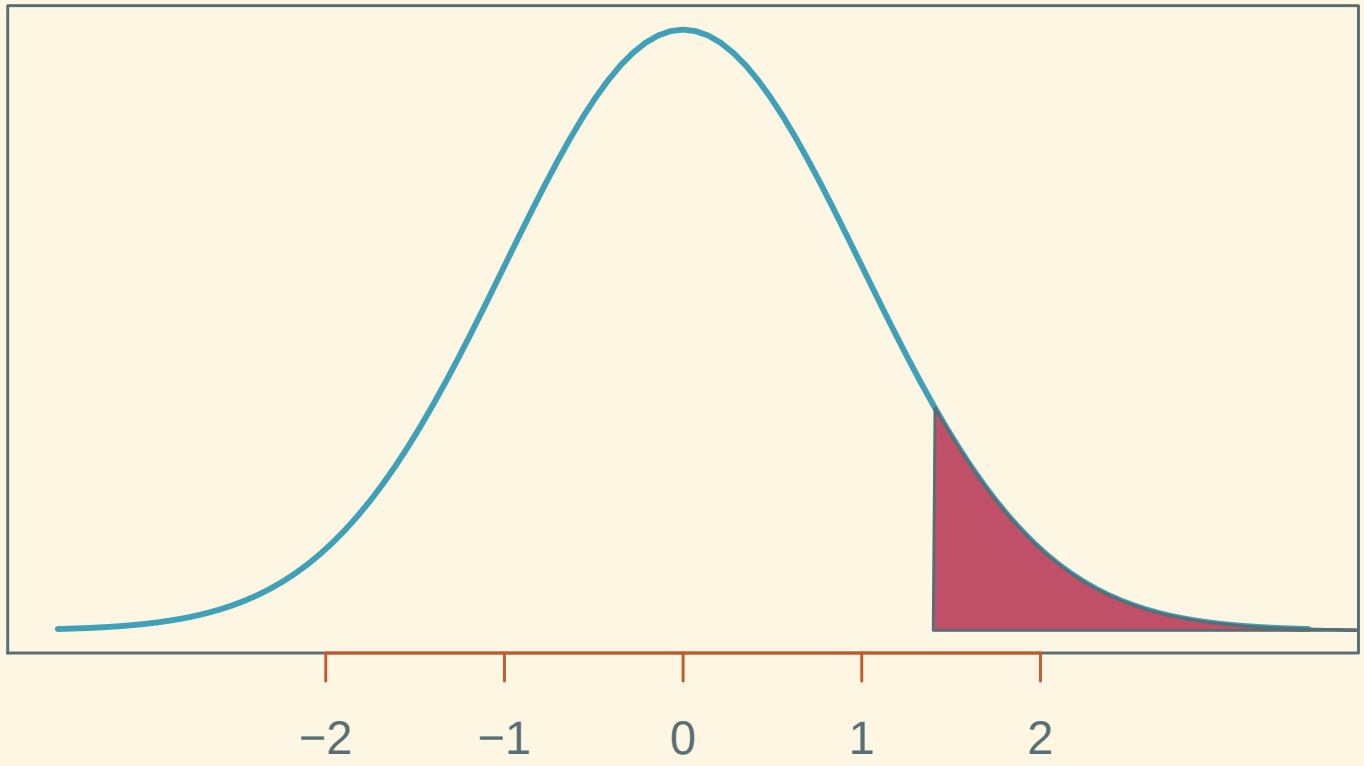


Figure: Approximating Sampling Distribution of $\bar{X}$ using a Normal. Shaded area is $P(\bar{X} > 103)$

Area Under Normal Curve



Area under a curve using calculus:

$$\int_{1.4}^{\infty} \frac{1}{\sigma\sqrt{2\pi}} e^{-\left(\frac{x-0}{1}\right)^2} \, dx$$

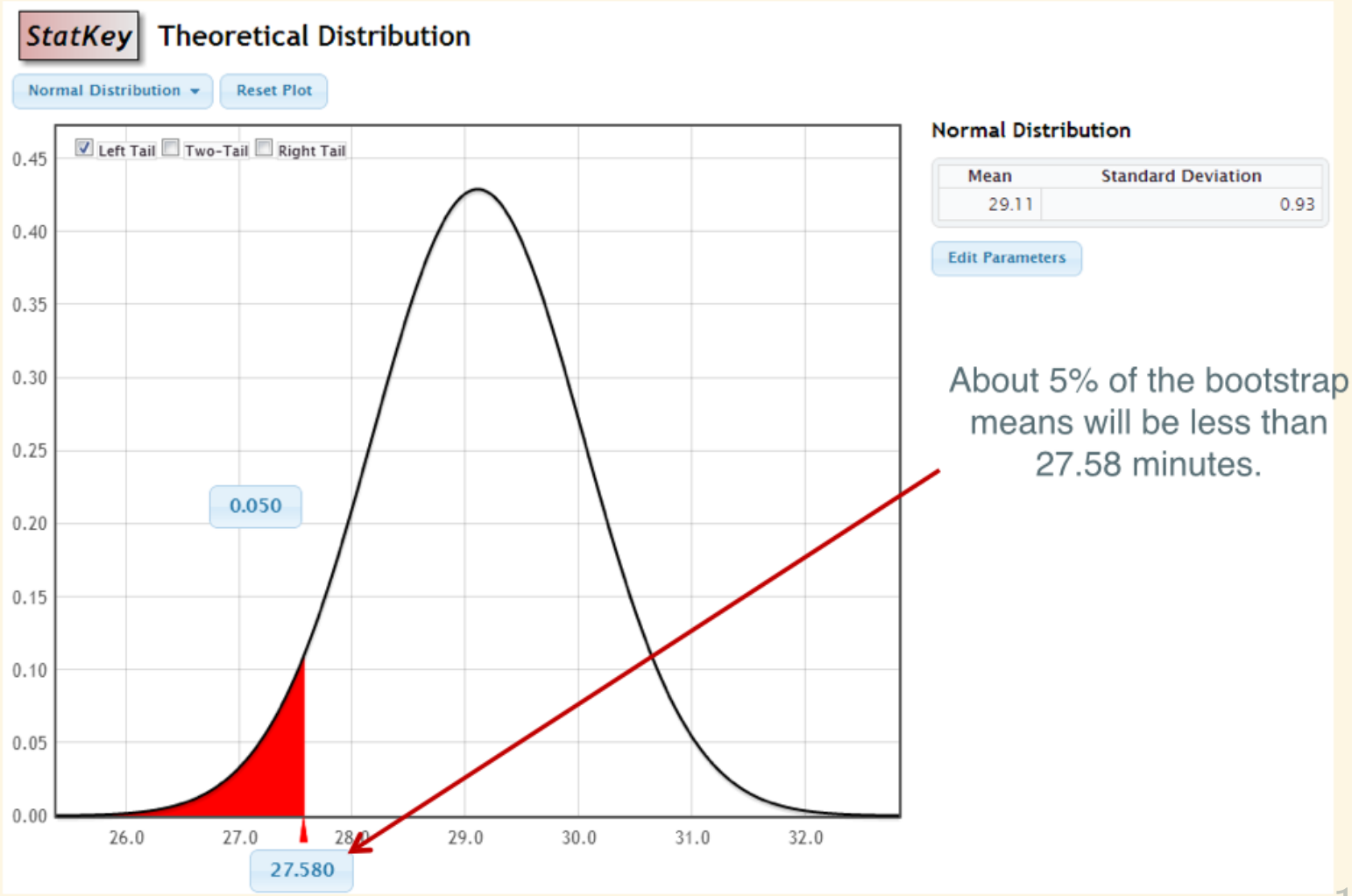
but this integrand doesn't have a closed-form antiderivative, so we

Quantiles of a Normal Curve

Example: Atlanta Commutes

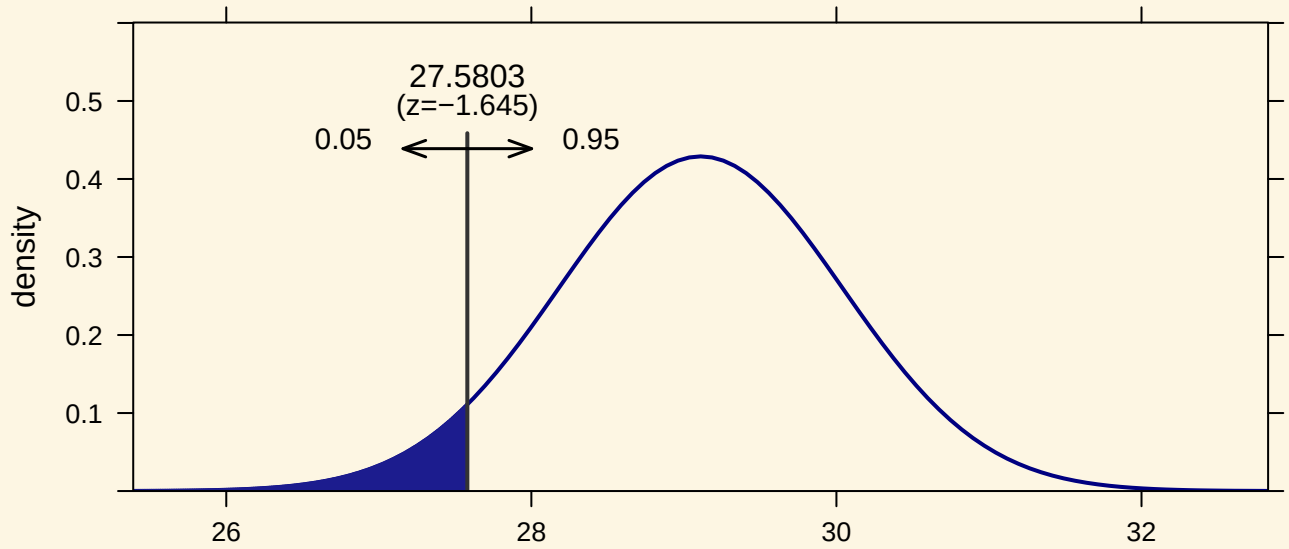
Suppose that the bootstrap distribution of means for samples of size 500 Atlanta commute times is $\mathcal{N}(29.11, 0.93)$. Find an endpoint (percentile) so that just 5% of the bootstrap means are smaller.

StatKey...


StatKey
Theoretical Distribution
Normal Distribution
Reset Plot
Left Tail
Two-Tail
Right Tail
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0.050
26.0
27.0
28.0
29.0
30.0
31.0
32.0
27.580
Normal Distribution
Mean
Standard Deviation
29.11
0.93
Edit Parameters
About 5% of the bootstrap means will be less than 27.58 minutes.

And in R ...

```
xqnorm(0.05, mean = 29.11, sd = 0.93)

##  P(X <= 27.5802861269351) = 0.05
##  P(X >  27.5802861269351) = 0.95
```




```
## [1] 27.58029
```

Outline

Goals

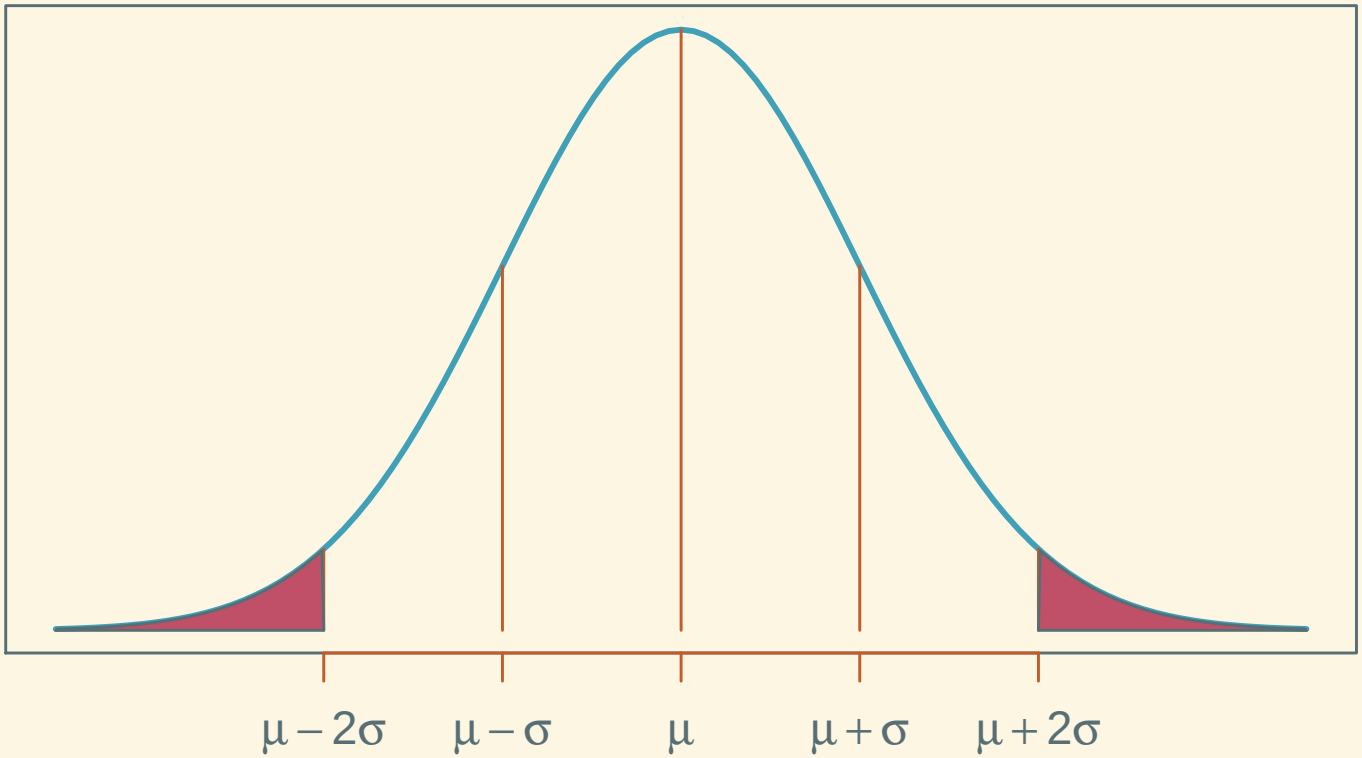
Confidence Intervals

If we can approximate a bootstrap distribution with a Normal, we can construct a confidence interval.

P-values

If we can approximate a randomization distribution with a Normal, we can compute P-values.

Quantiles of Normal Curves



For proportions and quantiles, only z-scores matter!

Converting to and from z-scores

Recall

$$Z = \frac{X - \mu}{\sigma} \qquad X = \sigma Z + \mu$$

Shifting observations left or right shifts the mean without changing σ. Scaling by a constant scales the mean and standard deviation by that constant.

$$X \sim \mathcal{N}(\mu, \sigma) \rightarrow Z = \frac{X - \mu}{\sigma} \rightarrow Z \sim \mathcal{N}(0, 1)$$
$$Z \sim \mathcal{N}(0, 1) \rightarrow X = Z\sigma + \mu \rightarrow X \sim \mathcal{N}(\mu, \sigma)$$

Outline

CI Summary

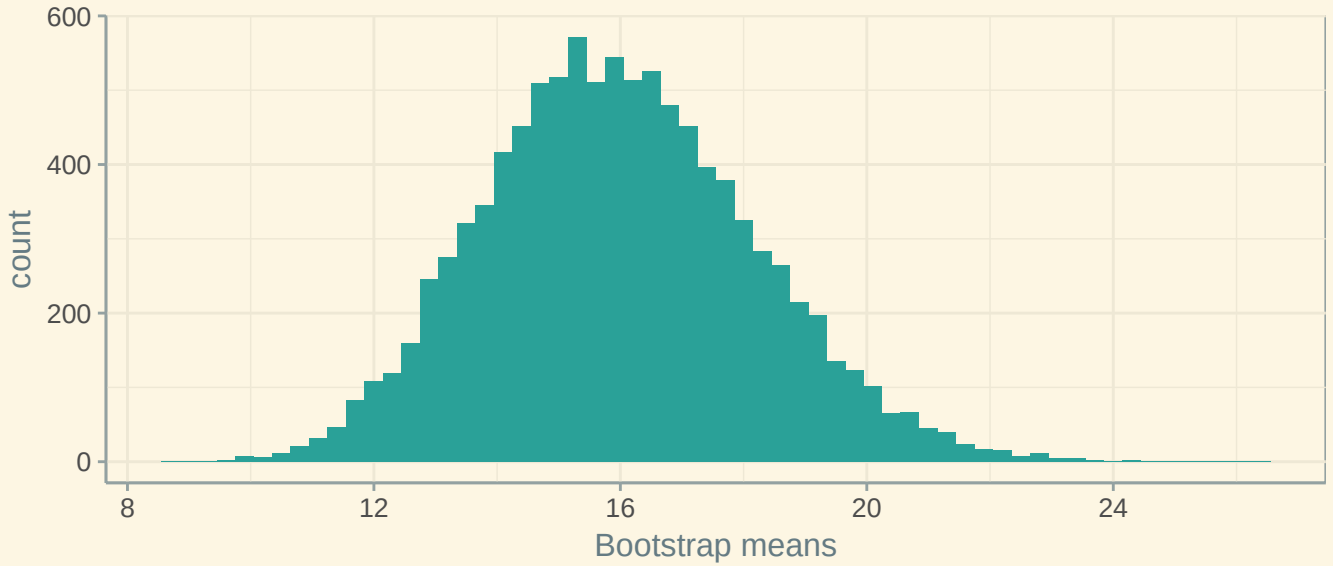
To compute a confidence interval when the bootstrap distribution is Normal, use

$$\text{Endpoint} = \text{Observed Statistic} + Z^* \cdot \text{Bootstrap SE}$$

where Z^* is the Z-score of the endpoint appropriate for the confidence level, computed from a standard normal ($\mathcal{N}(0, 1)$).

A "Pure" Bootstrap CI

```
library("Lock5Data")
data("MustangPrice")
xbar <- mean(~Price, data = MustangPrice)
boot.means <- do(10000) * mean(~Price, data = resample(MustangPrice))
```

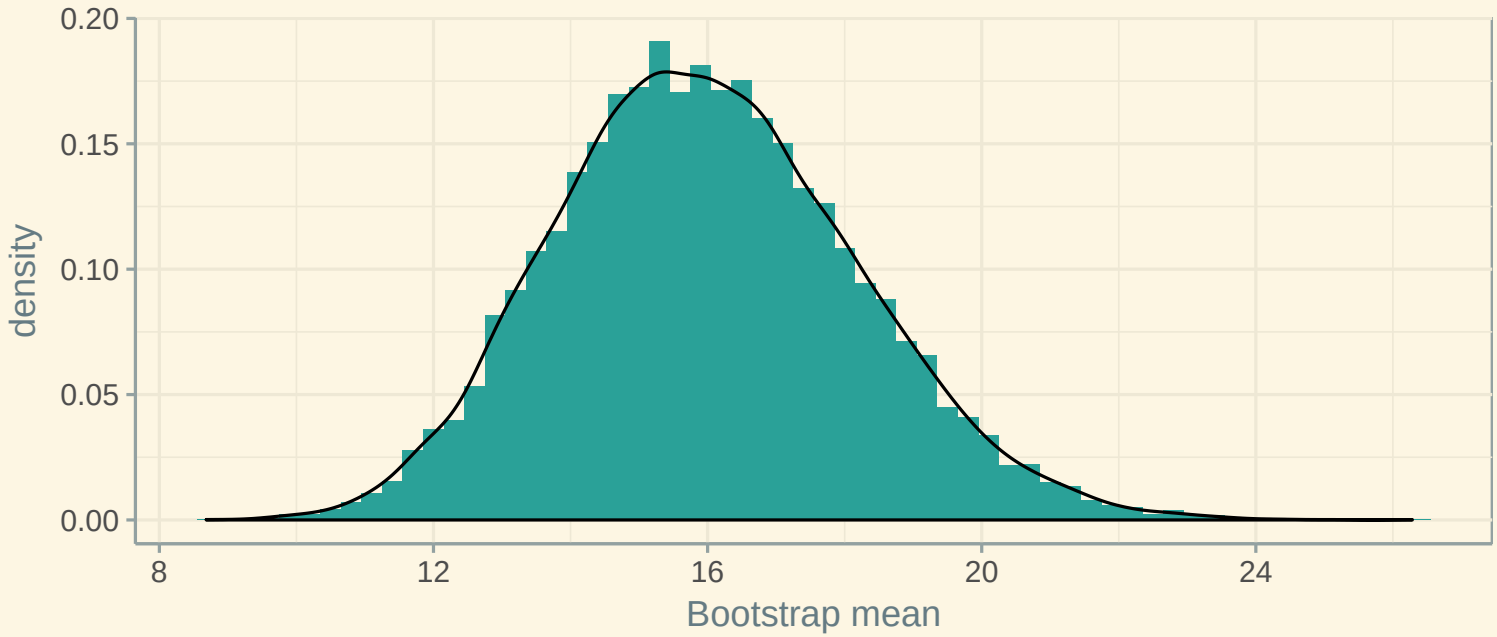



```
CI.99.boot <- quantile(~mean, data = boot.means, prob = c(0.005, 0.995))
CI.99.boot
```

```
      0.5%    99.5%
10.95594 22.02018
```

Normal CI Using z-scores and Bootstrap SE

```
zstar.99.lower <- qnorm(0.005) # without the 'x', no extra output
zstar.99.upper <- qnorm(0.995)
se <- sd(~mean, data = boot.means)
## compute the endpoints and concatenate them
CI.99.from.z <- c(xbar + zstar.99.lower * se, xbar + zstar.99.upper * se)
CI.99.from.z # show the results

## [1] 10.35545 21.60455
```



Outline

P-values Using a Standard Normal

P-values from a Standard Normal

Computing P-values when the randomization distribution is Normal is the reverse process:

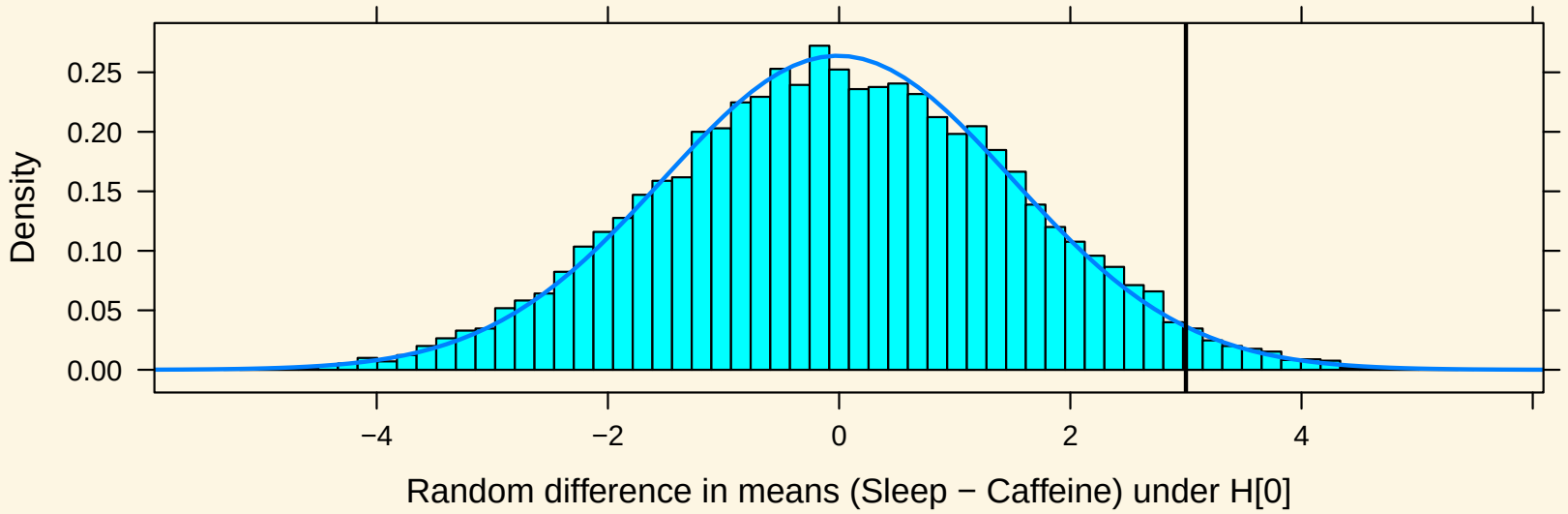
1. Convert the observed statistic to a z-score within the randomization distribution (i.e., using its mean and standard deviation).

$$Z_{observed} = \frac{\text{observed statistic} - \text{null parameter}}{\text{randomization SD}}$$

2. Find the relevant area beyond $Z_{observed}$ using a Standard Normal

Example: Sleep and Caffeine

```
library("Lock5Data"); data("SleepCaffeine"); set.seed(00029747)
obs.diff <- diffmean(Words ~ Group, data = SleepCaffeine)
random.diffs <- do(10000) *
    diffmean(Words ~ shuffle(Group), data = SleepCaffeine)
```

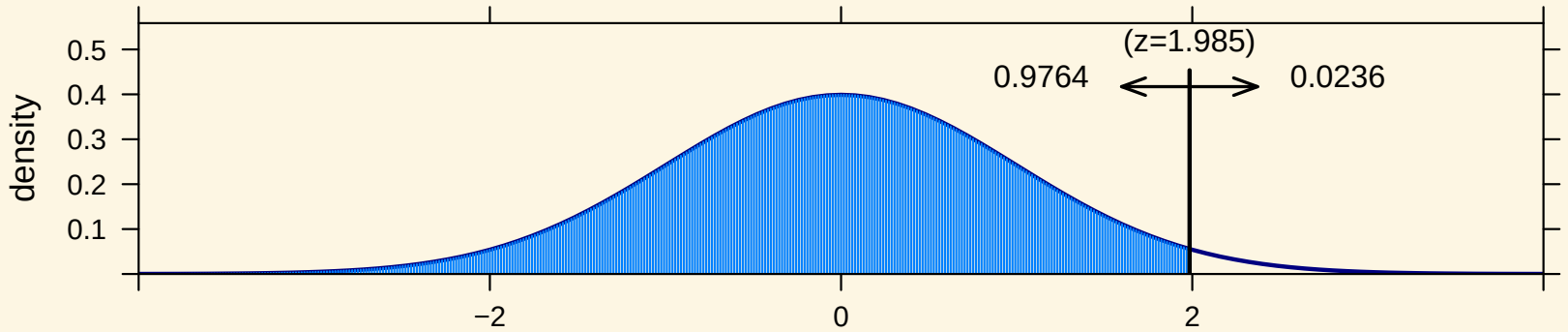


Example: Sleep and Caffeine

```
## Let's calculate the P-value from the randomization distribution first
P.right <- prop(~(diffmean >= obs.diff), data = random.diffs)
P.right

##    TRUE
## 0.0243
```

Example: Sleep and Caffeine

```
## Now from the standard Normal:
null.diff <- 0
se <- sd(~diffmean, data = random.diffs)
z.obs <- (obs.diff - null.diff) / se
P.right <- xpnorm(z.obs, lower.tail = FALSE, verbose = FALSE); P.right
```




```
##   diffmean
## 0.02359091
```

Outline

Limits of Normal Approximation So Far

- We have still needed to do all that randomization / resampling to calculate the standard error.
- We can avoid that with some more theory.

Cases to Address

We might be interested in CIs and tests for the following parameters:

1. Single Proportion
2. Single Mean
3. Difference of Proportions
4. Difference of Means
5. Mean of Differences
6. Correlation

Analytic Approximations of Sampling Distributions

Param.	Stat.	Randomization	Theory SE	Test Dist.
p	$\hat{p}$	Simulate from p_0	$\sqrt{\frac{p_0(1-p_0)}{n}}$	Normal
μ	$\bar{x}$	Bootstrap + shift	$\frac{s}{\sqrt{n}}$	t_{n-1}
$p_A - p_B$	$\hat{p}_A - \hat{p}_B$	Scramble groups	$\sqrt{\frac{\hat{p}(1-\hat{p})}{n_A} + \frac{\hat{p}(1-\hat{p})}{n_B}}$	Normal
$\mu_A - \mu_B$	$\bar{x}_A - \bar{x}_B$	Scramble groups	$\sqrt{\frac{s_A^2}{n_A} + \frac{s_B^2}{n_B}}$	$t_{\min(n_A-1, n_B-1)}$
μ_D	$\bar{x}_D$	Flip pairs*	$\frac{s_D}{\sqrt{n_D}}$	t_{n_D-1}
ρ	r	Scramble pairings	$\sqrt{\frac{1-r^2}{n-2}}$	t_{n-2}

$$CI: \quad \text{Statistic} \pm \text{Critical Value} \times \widehat{SE}$$

$$\text{Sandardized Test Statistic}: \quad \frac{\text{Statistic} - \text{Null Param.}}{\widehat{SE}}$$

Distribution of $\hat{p}$

For pop. proportion p and the samples size n, the sampling distribution of $\hat{p}$ has mean p and standard error

$$SE_{\hat{p}} = \sqrt{\frac{p(1-p)}{n}}$$

It is approximately normal, for large n and "medium" p. Roughly:

$$np \geq 10 \quad \text{AND} \quad n(1-p) \geq 10$$

CI Summary: Single Proportion

To compute a confidence interval for a proportion when the bootstrap distribution for $\hat{p}$ is approximately Normal (i.e., $n\hat{p}$ and $n(1-\hat{p}) \geq 10$), use

$$\hat{p} \pm Z^* \cdot \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$$

where Z^* is the Z-score of the endpoint appropriate for the confidence level, computed from a standard normal ($\mathcal{N}(0,1)$).

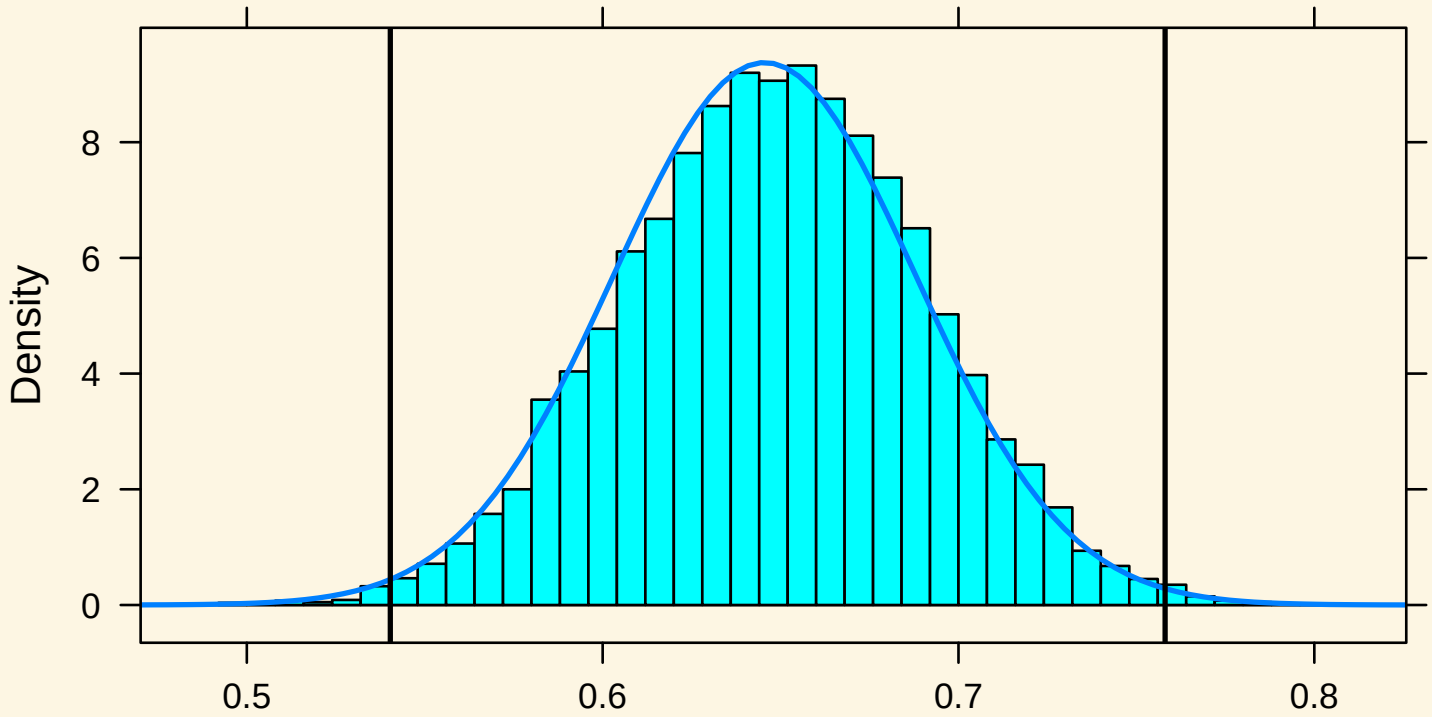
Example: Kissing Right

```
KissingRight <-
    data.frame(direction = rep(c("right","left"), times = c(80, 44)))
p.hat <- prop(~(direction == "right"), data = KissingRight)
p.hat

##      TRUE
## 0.6451613
```

```
boot.phats <- do(10000) *
    resample(KissingRight) %>% prop(~(direction == "right"), data = .)
### Changing the variable name (the auto-provided one is not useful)
names(boot.phats) <- "sim.p.hat"
```

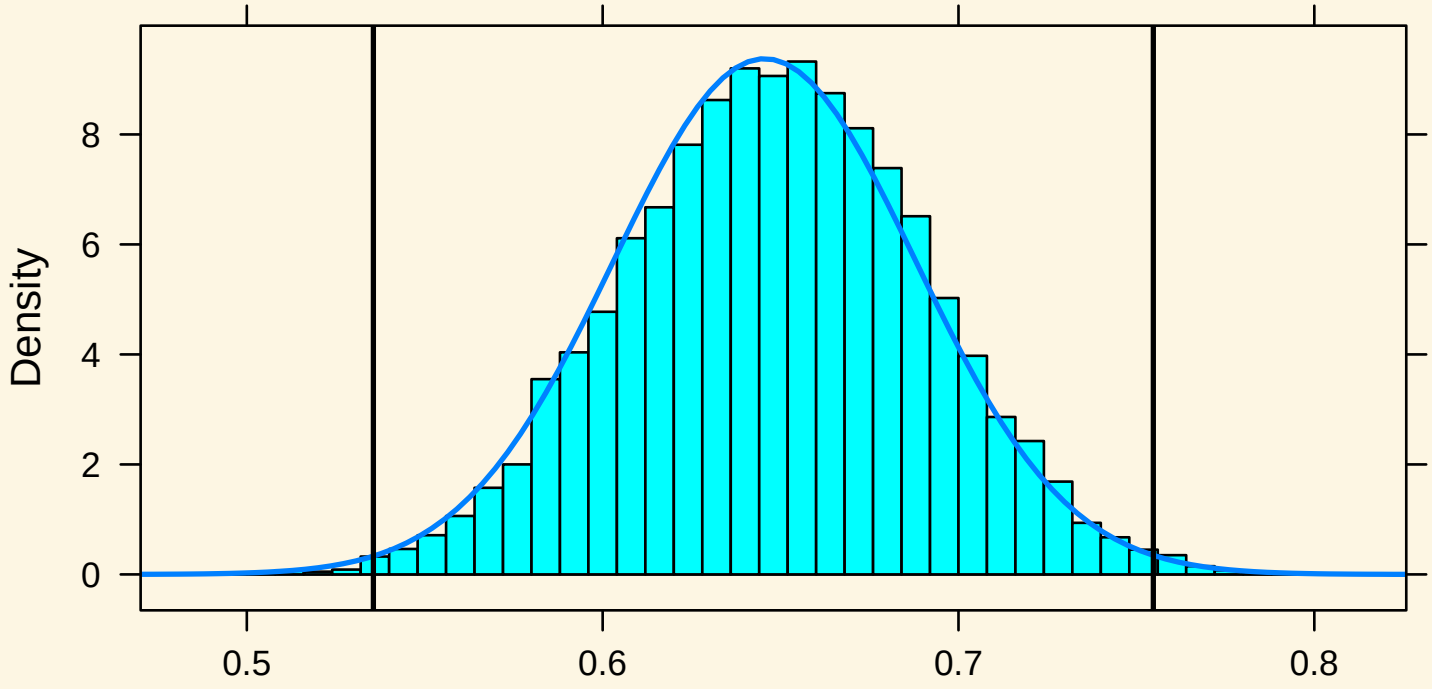
Kissing Right: "Pure" Bootstrap CI

```
CI.99.boot <- quantile(~sim.p.hat, data = boot.phats, prob = c(0.005, 0.995))
CI.99.boot

##       0.5%     99.5%
## 0.5403226 0.7580645
```



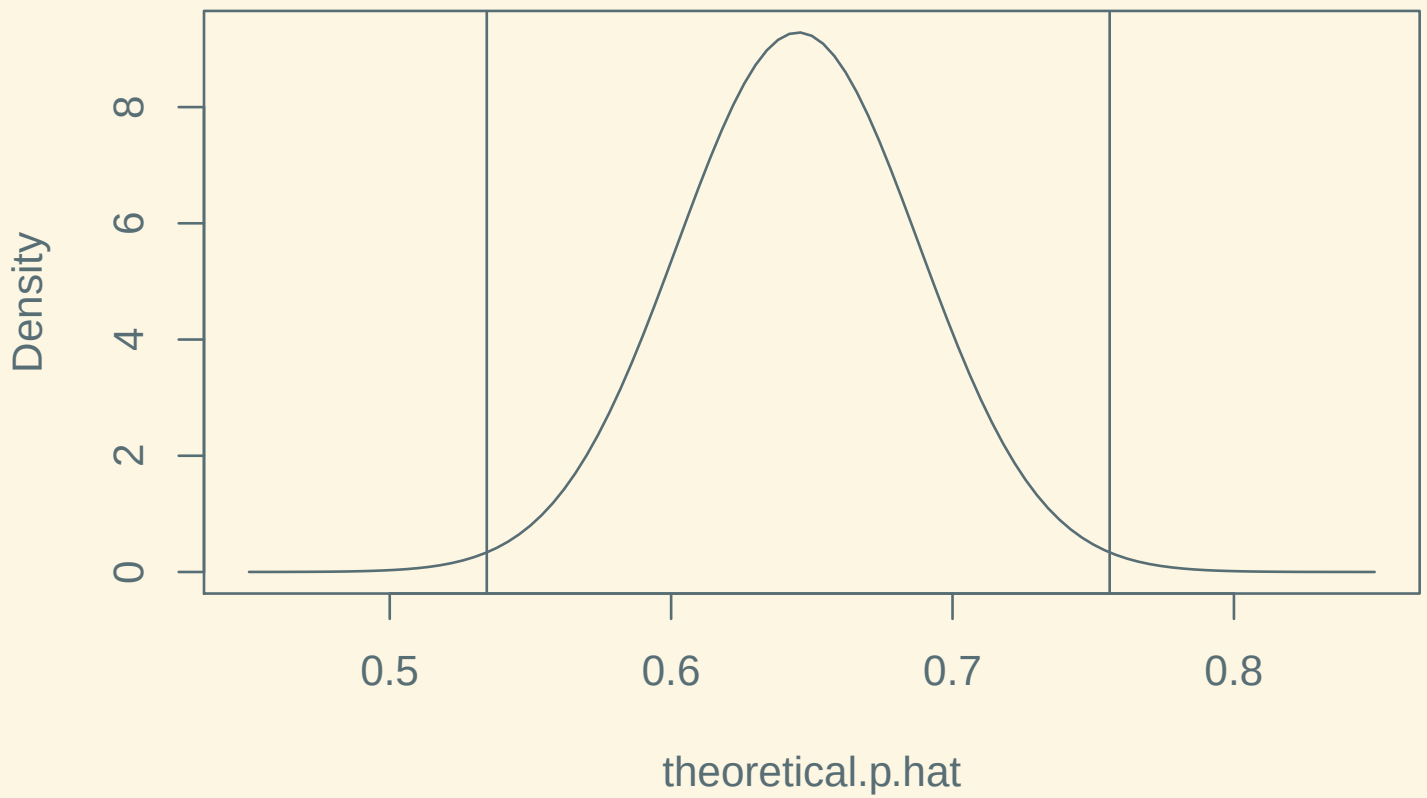
Kissing Right: Normal CI Using Bootstrap SE

```
zstar.99 <- qnorm(c(0.005, 0.995)) # computing both ends at once
boot.se <- sd(~sim.p.hat, data = boot.phats)
CI.99.normal.boot.se <- p.hat + zstar.99 * boot.se
CI.99.normal.boot.se

## [1] 0.5355704 0.7547522
```



Kissing Right: Normal CI Using Theoretical SE

```
zstar.99 <- qnorm(c(0.005, 0.995))
theory.se <- sqrt(p.hat * (1 - p.hat) / 124) #Careful with parentheses!
(CI.99.normal.theoretical.se <- p.hat + zstar.99 * theory.se)

## [1] 0.5344847 0.7558379
```



P-values for a sample proportion from a Standard Normal

Computing P-values when the null sampling distribution is approximately Normal (i.e., np_0 and $np_0(1-p_0) \geq 10$) is the reverse process:

1. Convert $\hat{p}$ to a z-score within the theoretical distribution .

$$Z_{observed} = \frac{\hat{p} - p_0}{\sqrt{\frac{p_0(1-p_0)}{n}}}$$

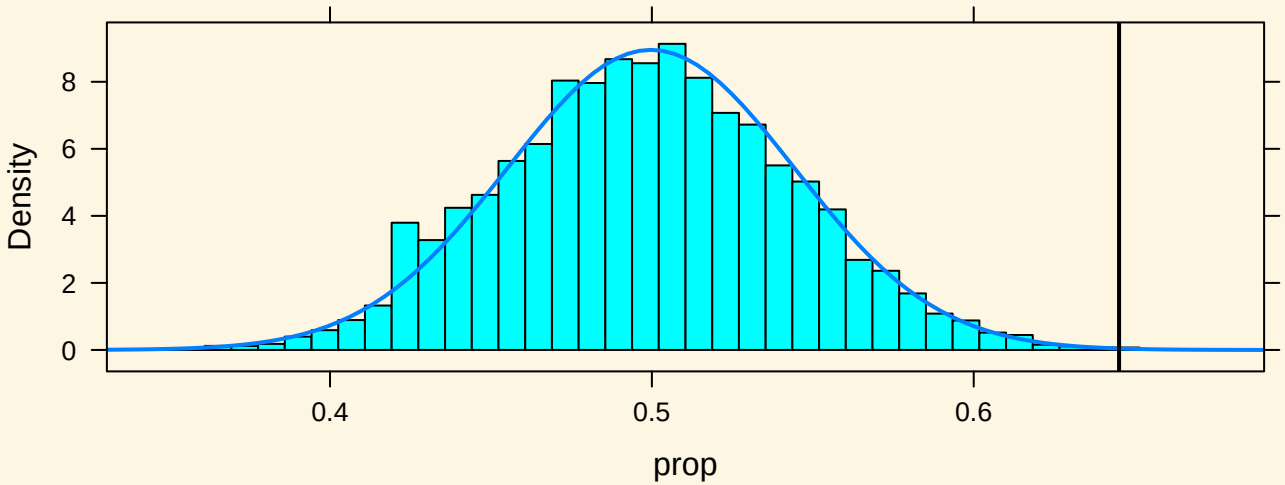
2. Find the relevant area beyond $Z_{observed}$ using a Standard Normal

Example: Kissing Right Hypothesis Test

```
library("Lock5Data")
KissingRight <-
    data.frame(direction = rep(c("right","left"), times = c(80, 44)))
p.hat <- prop(~(direction == "right"), data = KissingRight)
random.phats <- do(10000) * rflip(124, 0.5)
```

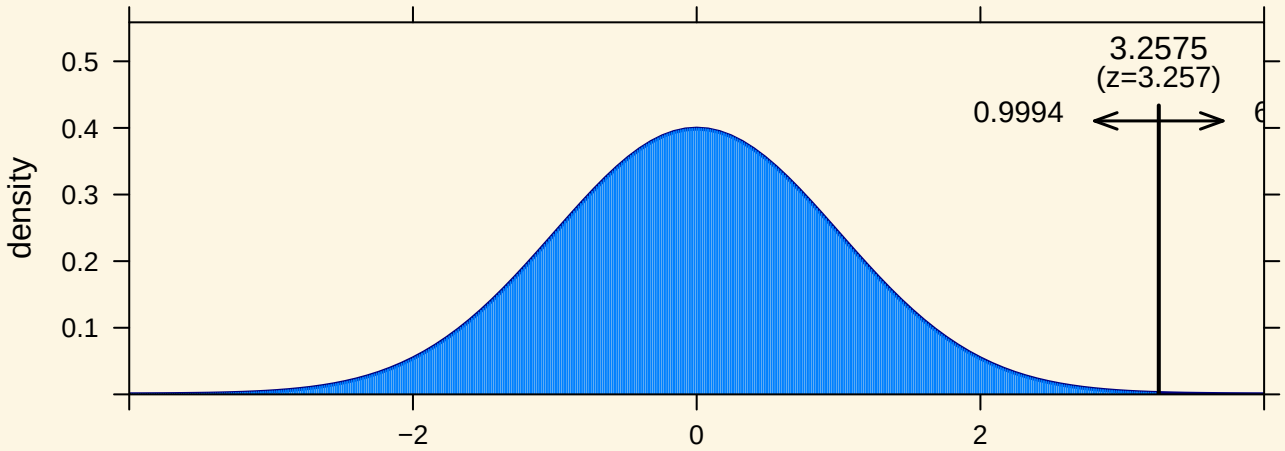
Kissing Right: "Pure" Randomization P-value

```
P.value.randomization <- prop(~(prop >= p.hat), data = random.phats)
P.value.randomization

##  TRUE
## 8e-04
```



Kissing Right: Normal P-value Using Randomization SE

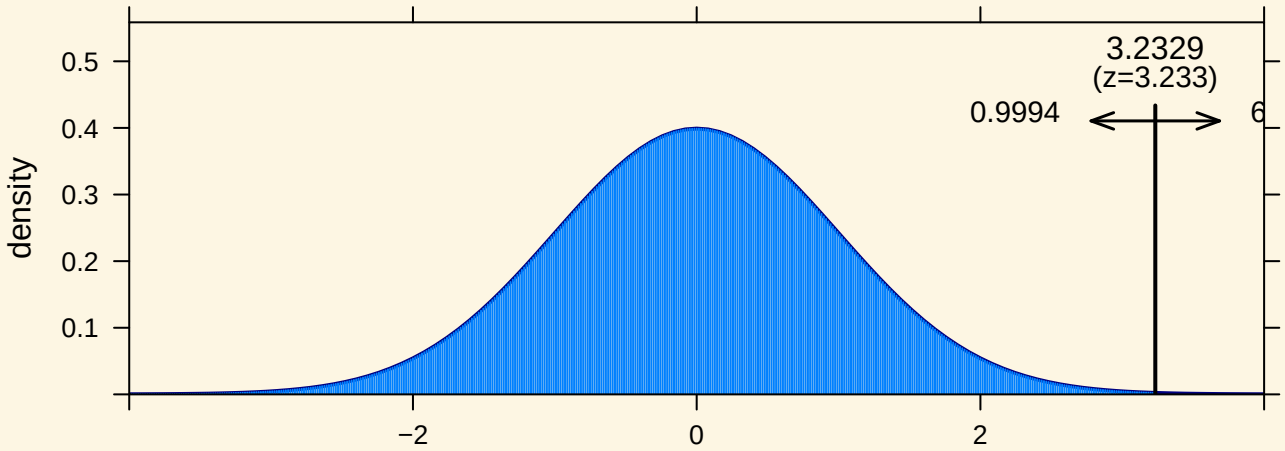
```
se.randomization <- sd(~prop, data = random.phats)
p0 <- 0.5; z.obs <- (p.hat - p0) / se.randomization
P.value.randomization.se <- xpnorm(z.obs, lower.tail = FALSE, verbose = FALSE)
```




```
P.value.randomization.se

##         TRUE
## 0.0005620616
```

Kissing Right: Normal P-value Using Theoretical SE

```
p0 <- 0.5; n <- 124; se.theory <- sqrt(p0 * (1 - p0) / n)
z.observed <- (p.hat - p0) / se.theory
P.value.theory.se <- xpnorm(z.observed, lower.tail = FALSE, verbose = FALSE)
```




```
P.value.theory.se

##         TRUE
## 0.000612712
```