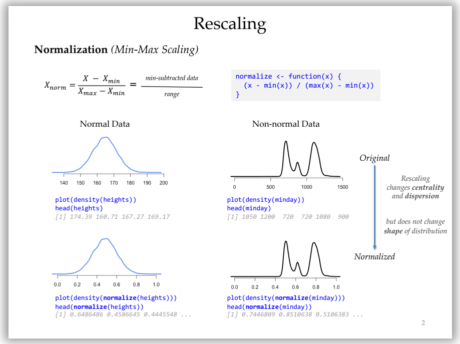
Business Analytics Using
Computational Statistics

Week 4
Computational Intervals

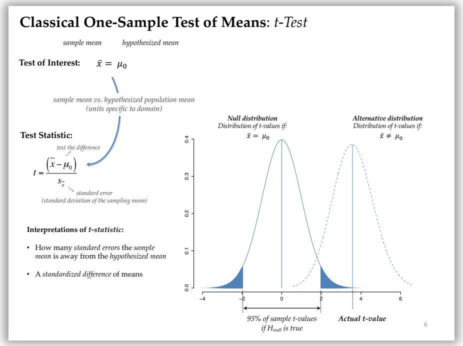
Week 5
Bootstrapping for Inference

Week 6
Permutation Tests

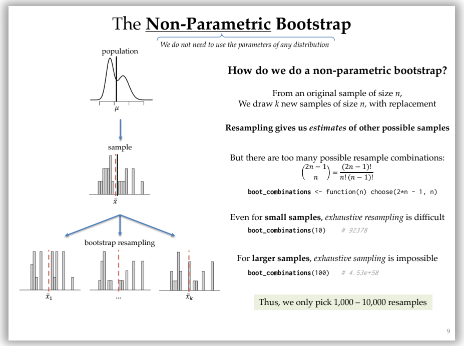
Rescaling Scores



t-Tests Reviewed



Bootstrapped Testing

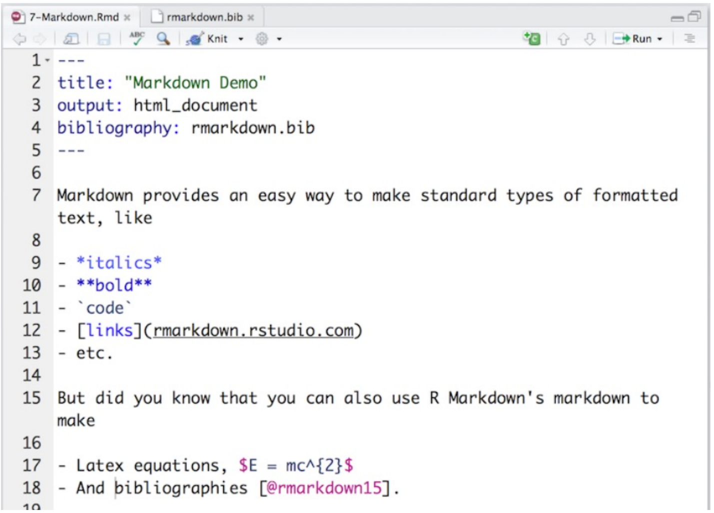


RMarkdown

https://rmarkdown.rstudio.com

Markdown (*.md)

*Easy syntax to add **formatting** to text*

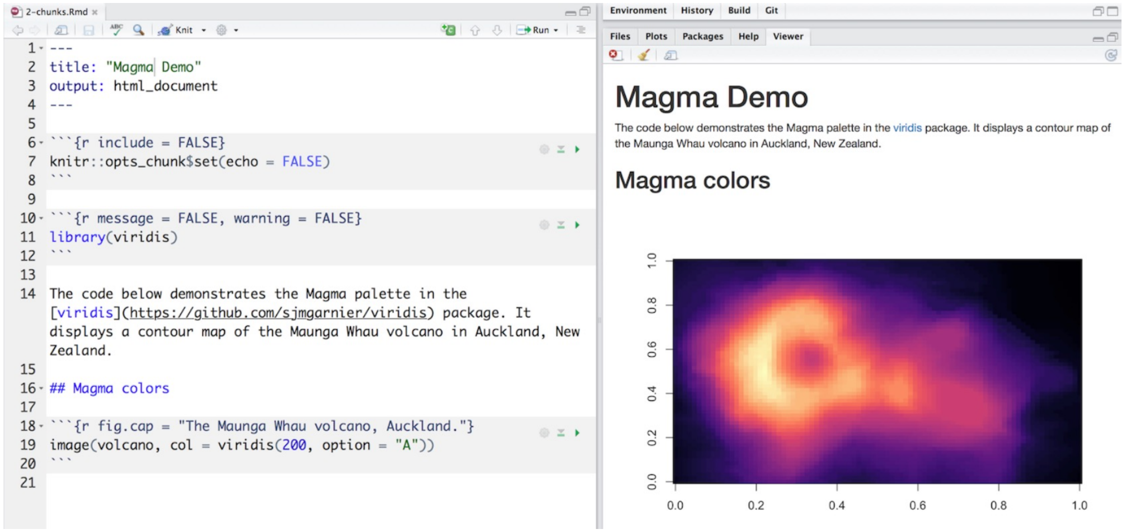


You can use Pandoc's Markdown to make:

- Headers
- Lists
- Links
- Images
- Block quotes
- Latex equations
- Horizontal rules
- Tables
- Footnotes
- Bibliographies and Citations
- Slide breaks
- Italicized text
- Bold text
- Superscripts
- Subscripts
- Strikethrough text

RMarkdown (*.Rmd)

*Easy syntax to add **R output** to Markdown*



The following output formats are available to use with R Markdown.

Documents

- html_notebook - Interactive R Notebooks
- html_document - HTML document w/ Bootstrap CSS
- pdf_document - PDF document (via LaTeX template)
- word_document - Microsoft Word document (docx)
- odt_document - OpenDocument Text document
- rtf_document - Rich Text Format document
- md_document - Markdown document (various flavors)

Presentations (slides)

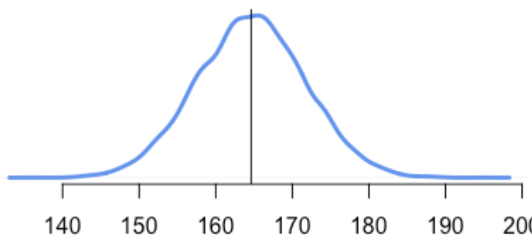
- ioslides_presentation - HTML presentation with ioslides
- revealjs::revealjs_presentation - HTML presentation with reveal.js
- slidy_presentation - HTML presentation with W3C Slidy
- beamer_presentation - PDF presentation with LaTeX Beamer
- powerpoint_presentation: PowerPoint presentation

More

- flexdashboard::flex_dashboard - Interactive dashboards
- tufte::tufte_handout - PDF handouts in the style of Edward Tufte
- tufte::tufte_html - HTML handouts in the style of Edward Tufte
- tufte::tufte_book - PDF books in the style of Edward Tufte
- html_vignette - R package vignette (HTML)
- github_document - GitHub Flavored Markdown document

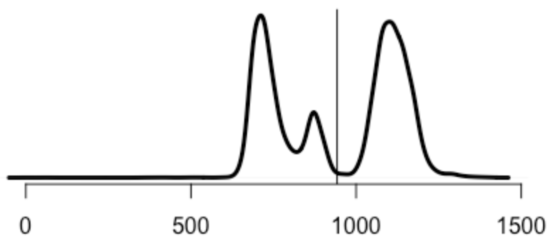
Rescaling

Normal Data




```
plot(density(heights))
head(heights)
[1] 174.39 160.71 167.27 169.17
```

Non-normal Data




```
plot(density(minday))
head(minday)
[1] 1050 1200  720  720 1080  900
```

Original

Normalization *(Min-Max Scaling)*

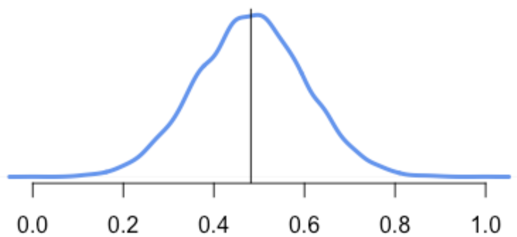
$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} = \frac{\text{(Min difference)}}{\text{range}}$$

```
normalize <- function(x) {
  (x - min(x)) / (max(x) - min(x))
}
```

*Rescaling changes **centrality** and **dispersion***

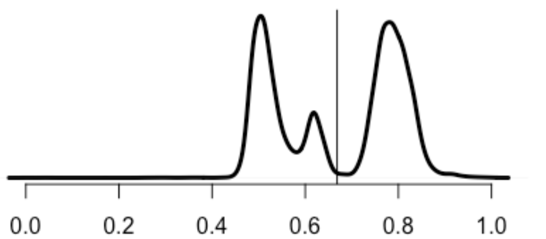
*but does not change **shape** of distribution*

Normalized




```
plot(density(normalize(heights)))
head(normalize(heights))
[1] 0.6486486 0.4586645 0.4445548 ...
```

Normalized




```
plot(density(normalize(minday)))
head(normalize(minday))
[1] 0.7446809 0.8510638 0.5106383 ...
```

Normalized

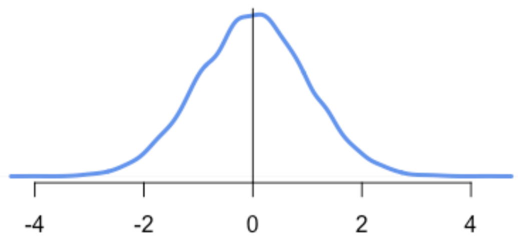
Standardization *(Z-score Normalization)*

(from Original)

$$X_{std} = \frac{X - \bar{X}}{s_x} = \frac{\textit{mean-centered data}}{\textit{standard deviation}}$$

```
standardize <- function(x) {
  (x - mean(x)) / sd(x)
}
```

Standard Normal




```
plot(density(standardize(heights)))
head(standardize(heights))
[1]  1.37346859 -0.54966899  0.37253734
```

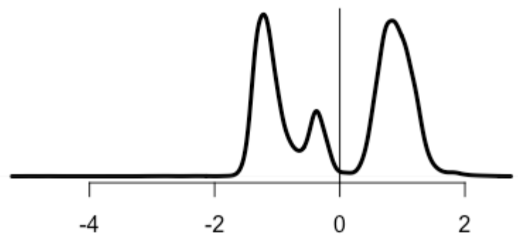
z-scores

```
mean(heights_std)
[1] 8.693189e-17
```

zero

```
sd(heights_std)
[1] 1
```

Standardized



Standardized

```
plot(density(standardize(minday)))
head(standardize(minday))
[1]  0.5668138  1.3576899 -1.1731137 ...
```

z-scores

```
mean(minday_std)
[1] -4.25589e-17
```

zero

```
sd(minday_std)
[1] 1
```

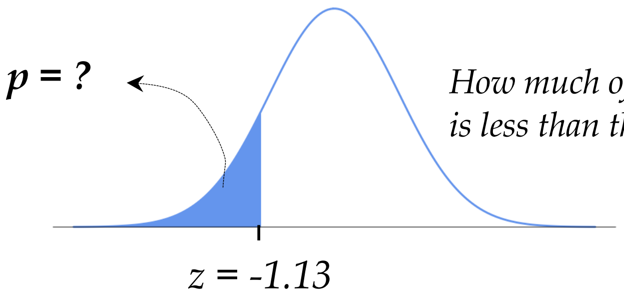
Is the standard normal distribution useful for anything?

What are the <u>units</u> of standardized data?

Z-scores and probabilities

for standard normal distributions

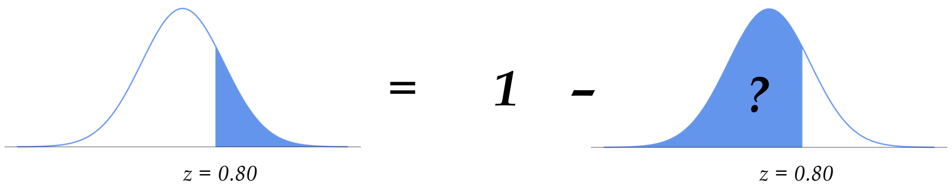
Probability given a z-score



How much of the std normal distribution is less than this z-value?

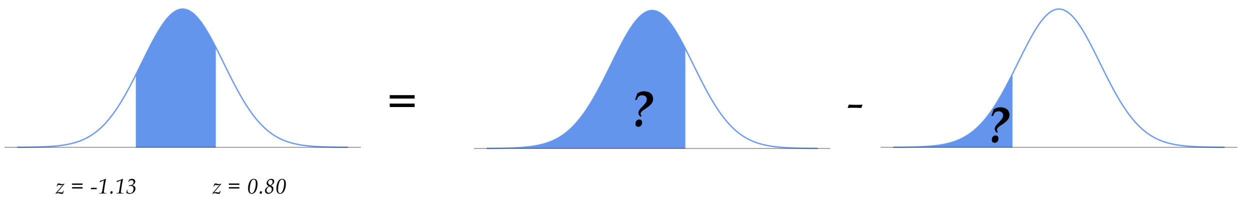
```
pnorm(-1.13)
[1] 0.1292381
```

12.92%



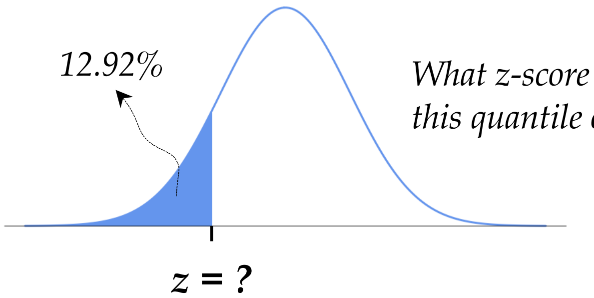

```
pnorm(0.80, lower.tail = FALSE)
[1] 0.2118554

1 - pnorm(0.80)
[1] 0.2118554
```




```
pnorm(0.80) - pnorm(-1.13)
[1] 0.6589065
```

Z-score given a probability (as quantile)



What z-score would be greater than this quantile of the distribution?

```
qnorm(0.1292)
[1] -1.130181
```

Classical One-Sample Test of Means: *t-Test*

sample mean *hypothesized mean*

Test of Interest: $\bar{x} = \mu_0$

sample mean vs. hypothesized population mean
(units specific to domain)

Test Statistic:

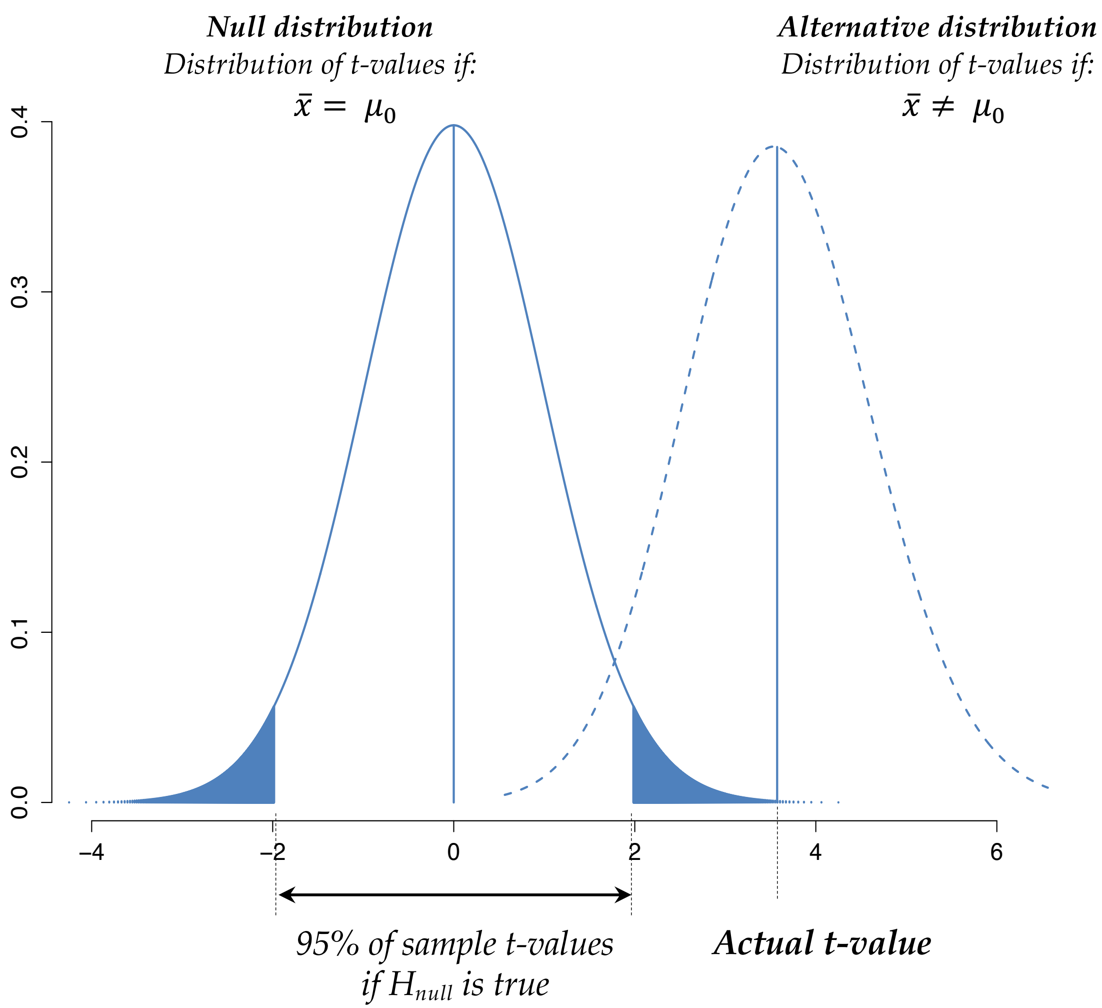
test the difference

$$t = \frac{\left(\bar{x} - \mu_0\right)}{s_{\bar{x}}}$$

standard error
(standard deviation of the sampling mean)

Interpretations of *t-statistic*:

- How many *standard errors* the *sample mean* is away from the *hypothesized mean*
- A *standardized difference* of means



Confidence Interval of μ: *t-distribution*

Sample statistics:

Sample Mean:
(weakly approx. to pop. mean)

$$\bar{x} = \frac{\sum x_i}{n} \sim \mu_x$$

Standard Deviation:

$$s = \sqrt{\frac{\sum(x_i - \bar{x})^2}{n-1}}$$

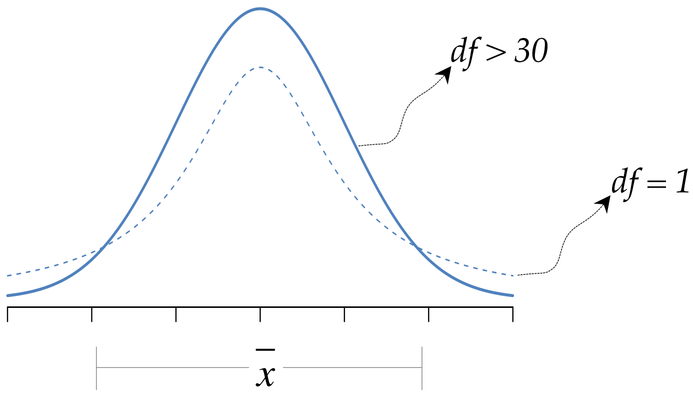
sample size: n

*degrees of freedom (**df**) = n-1*

Standard error of the mean:
(based on one sample)

$$s_{\bar{x}} = {}^{s}\!/\!\sqrt{n}$$

distribution of sample means $\bar{x}_1, \bar{x}_2, \bar{x}_3, \ldots, \bar{x}_{100}$



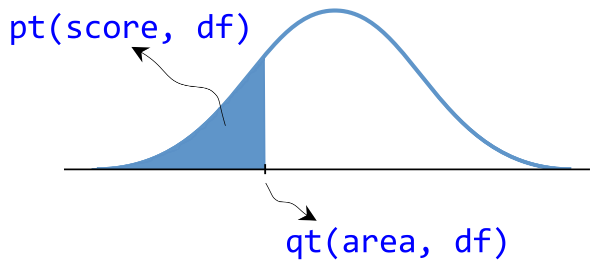
95% Confidence Interval:	$\bar{x} - 1.96\left({}^{s}/\sqrt{n}\right)$	*to*	$\bar{x} + 1.96\left({}^{s}/\sqrt{n}\right)$
99% Confidence Interval:	$\bar{x} - 2.58\left({}^{s}/\sqrt{n}\right)$	*to*	$\bar{x} + 2.58\left({}^{s}/\sqrt{n}\right)$

Confidence Interval of Population Mean (μ_x):

$$\bar{x} \pm t\left({}^{s}/\sqrt{n}\right)$$

Confidence Level	t ($df > 30$)
90.0%	**1.65**
95.0%	**1.96**
99.0%	**2.58**

Probability and scores on t-distributions:

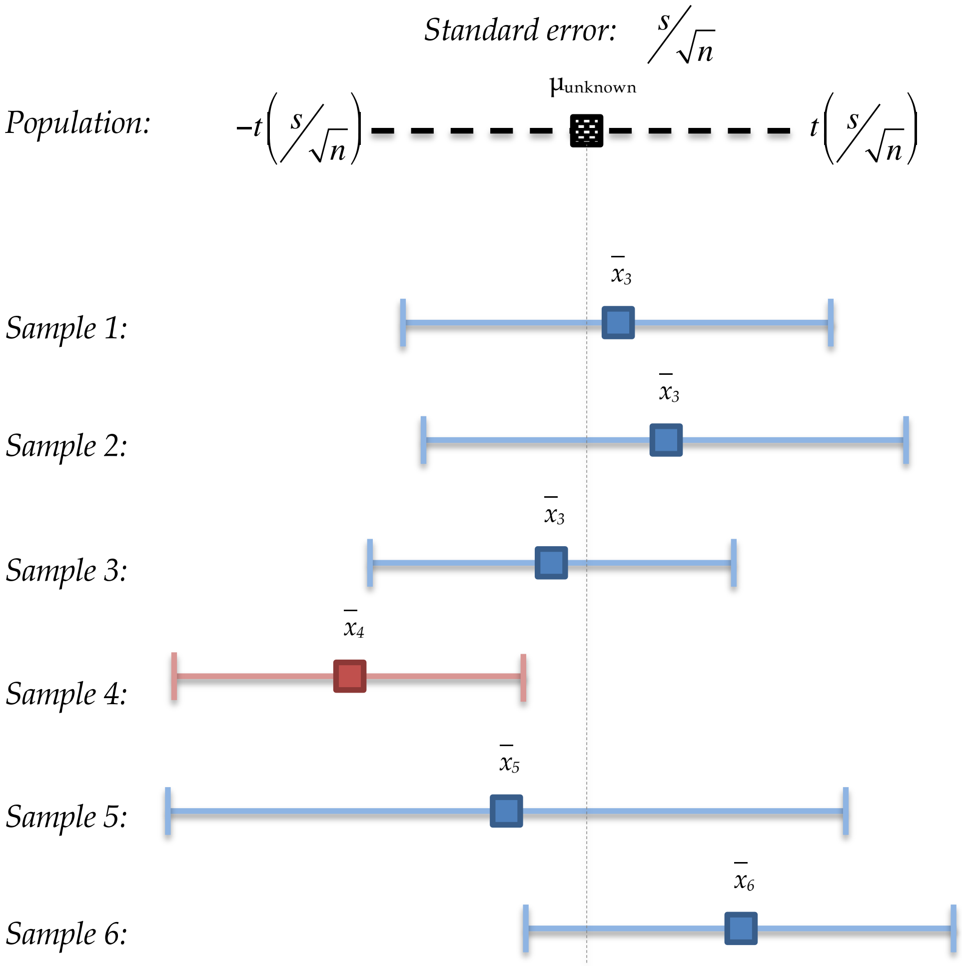


Examples:

```
pt(-1.13, df=100)
[1] 0.1305897

qt(0.13, df=100)
[1] -1.132817
```

Hypothesis Testing with t

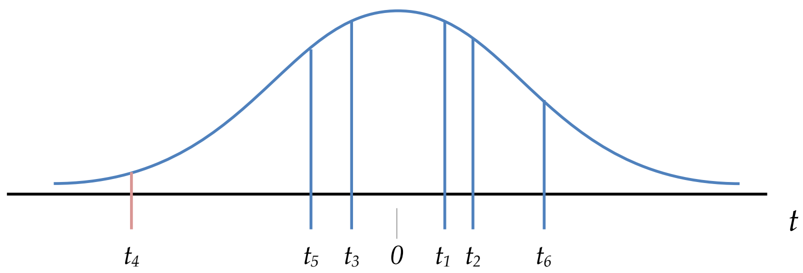


sample mean — *hypothesized population mean*

$$t = \frac{(\bar{x}-\mu_0)}{s_{\bar{x}}} = \frac{(\bar{x}-\mu_0)}{\left(s/\sqrt{n}\right)} \sim t_{n-1}$$

standard error

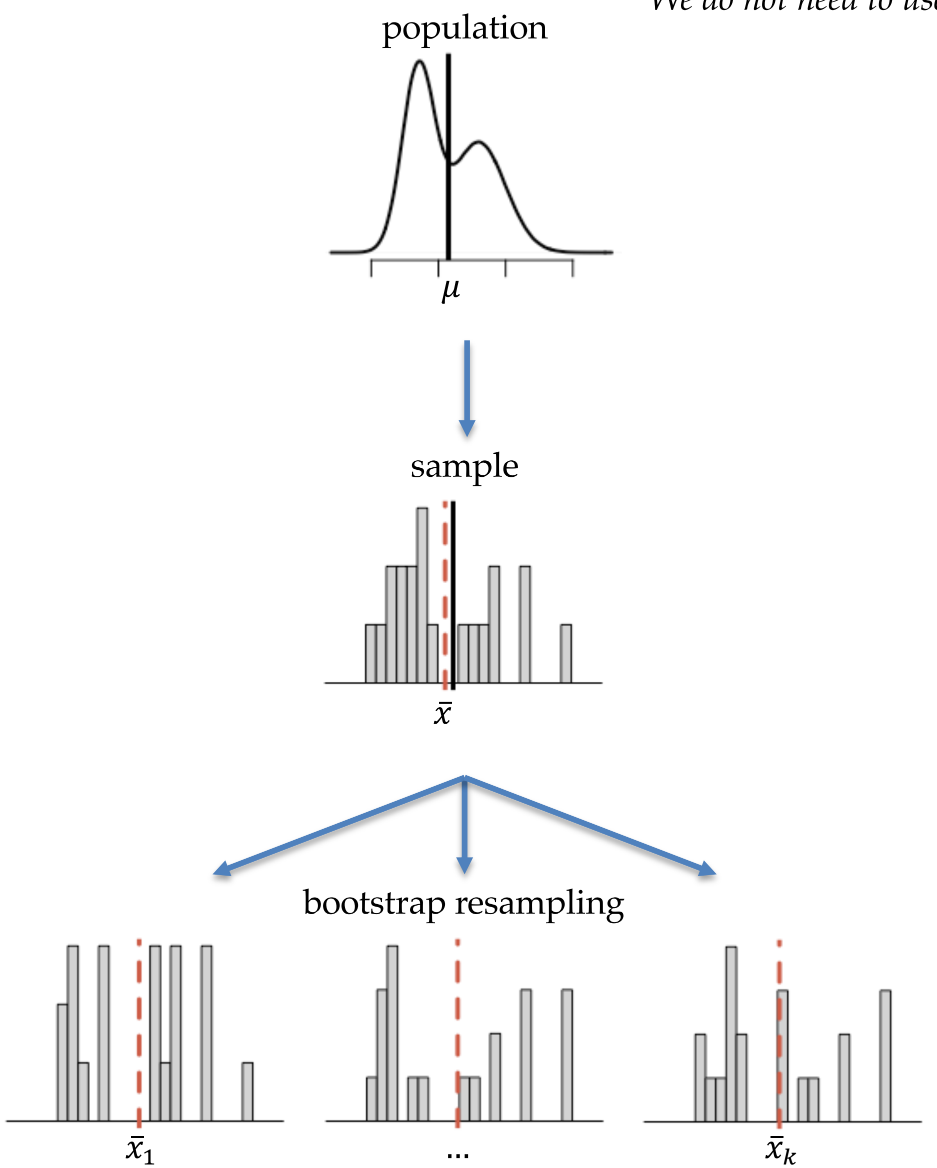
the distance (in standard errors) from the hypothesized mean to sample mean



t-distribution
(assuming H_{null} is correct)

The **Non-Parametric** Bootstrap

We do not need to use the parameters of any distribution



How do we do a non-parametric bootstrap?

From an original sample of size n,
We draw k new samples of size n, with replacement

Resampling gives us *estimates* of other possible samples

But there are too many possible resample combinations:

$$\binom{2n-1}{n} = \frac{(2n-1)!}{n!\,(n-1)!}$$

```r
boot_combinations <- function(n) choose(2*n - 1, n)
```

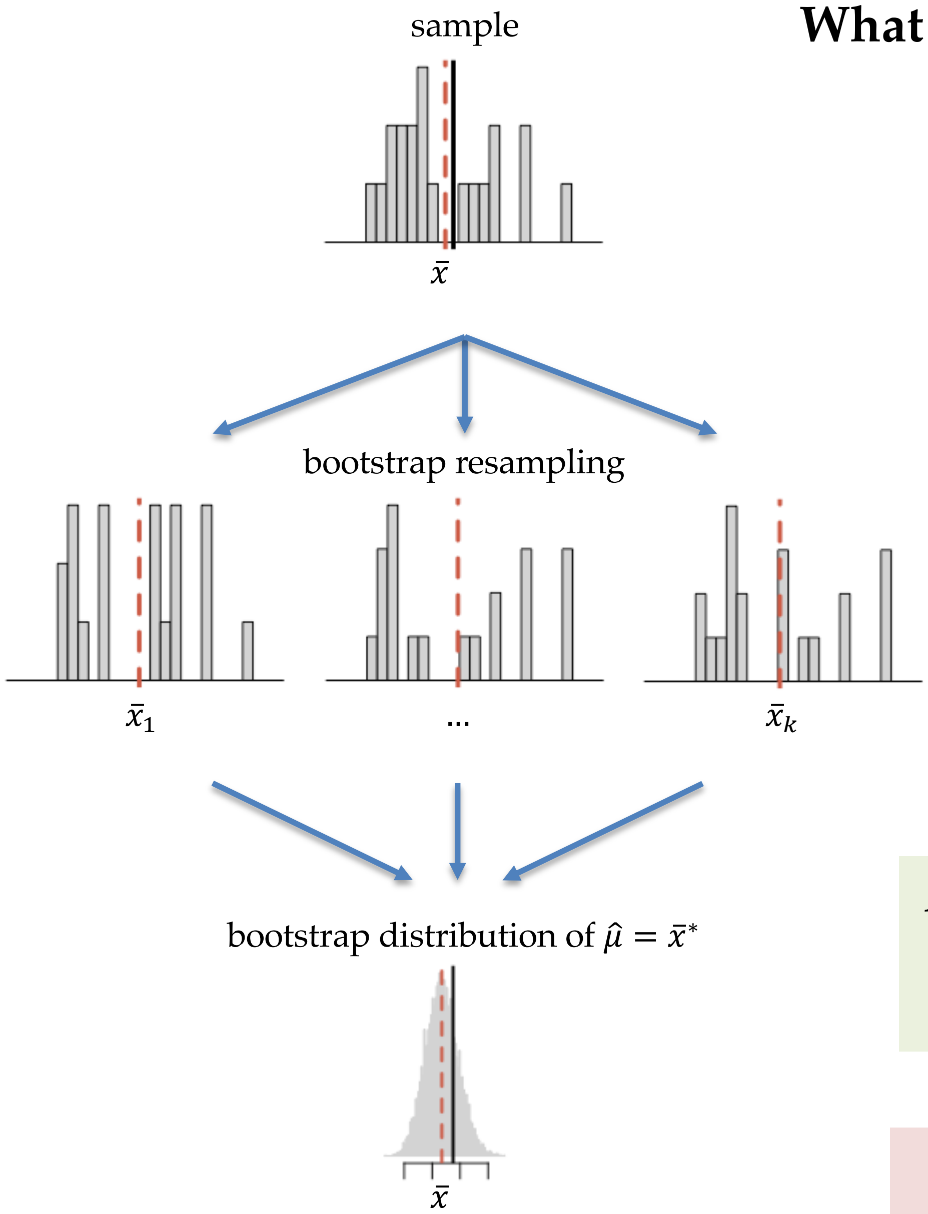
Even for **small samples**, *exhaustive resampling* is difficult

```r
boot_combinations(10)     # 92378
```

For **larger samples**, *exhaustive sampling* is impossible

```r
boot_combinations(100)    # 4.53e+58
```

Thus, we only pick 1,000 – 10,000 resamples



What does non-parametric bootstrapping give us?

Bootstrapped means $\bar{x}_i$ are centered
around sample mean $\bar{x}$,
not around population mean μ

The best estimate of population mean $\boldsymbol{\mu}$
is still the mean of the original sample $\bar{\boldsymbol{x}}$

Bootstrapping does **not** give us a more
accurate estimate of of μ than $\bar{x}$

Bootstrapping only tells us
how precise $\bar{\boldsymbol{x}}$ might be
(confidence interval of $\bar{x}$)

Bootstrap percentile confidence interval

Picking the 2.5% - 97.5% quantiles of $\bar{x}_i$
should give us estimate of the 95% CI of $\bar{x}$

Percentile CI is poor for **small samples** (n ≤ 30)

Randomness in Bootstrapping

Reproducibility

```
set.seed(10)
sample(1:10, replace=TRUE)
[1] 6 4 5 7 1 3 3 3 7 5
sample(1:10, replace=TRUE)
[1] 7 6 2 6 4 5 1 3 4 9
```

We can ***initialize*** *R's randomization algorithm with a given* ***seed*** *value*

```
set.seed(10)
sample(1:10, replace=TRUE)
[1] 6 4 5 7 1 3 3 3 7 5
sample(1:10, replace=TRUE)
[1] 7 6 2 6 4 5 1 3 4 9
```

With the same seed, other researchers can now ***reproduce our research***

Pick a random seed for every project, and use it `set.seed()`

```
# round(runif(1) * 10^9)
set.seed(864721226)
```

Destructive Resampling

How much data is lost in a single resampling with replacement?

```
set.seed(4356781)
resampled <- sample(1:100, replace=TRUE)

100 - length( unique(resampled) )
[1] 37
```

On each random pick of a bootstrap, every number has:

(1/n) probability of being picked, and

(1 - 1/n) probability of not being picked.

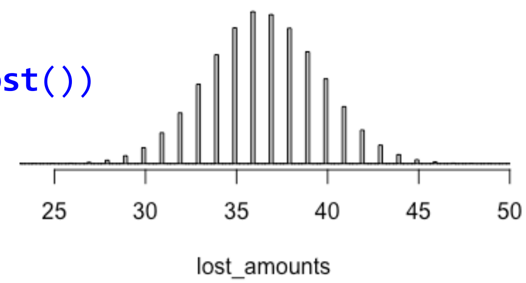
Probability of an item NOT being picked at large n:

$$\lim_{n\to\infty}\left(1-\frac{1}{n}\right)^n = \frac{1}{e} \approx 0.3678794$$

In general, how much data is lost in bootstrapping?

```
lost <- function() {
  100 - length(unique(sample(1:100, replace=TRUE)))
}

lost_amounts <- replicate(100000, lost())
hist(lost_amounts)
mean(lost_amounts)   # [1] 36.58539
```



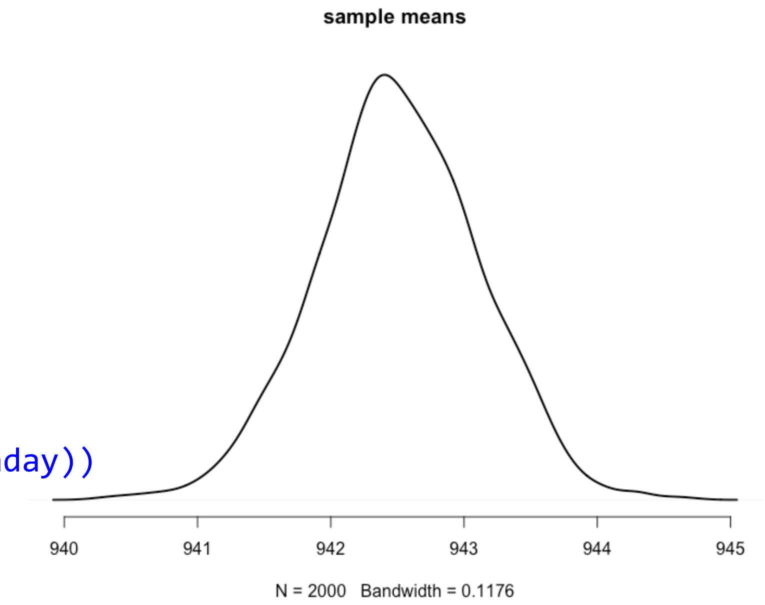
~1/3rd of data is lost in a bootstrap!

Sampling Statistics: Mean vs. Median

```
n = length(minday)
num_boot <- 2000
sample_statistic <- function(stat_function, sample0) {
  resample <- sample(sample0, length(sample0), replace=TRUE)
  stat_function(resample)
}
```

This function takes a ***function parameter****…*

… and runs it whenever needed



Bootstrapped means

```
sample_means <- replicate(num_boot, sample_statistic(mean, minday))
plot(density(sample_means), lwd=2, main="sample means")
quantile(sample_means, probs = c(0.025, 0.975))
  #   2.5%     97.5%
  # 941.33    943.63
```

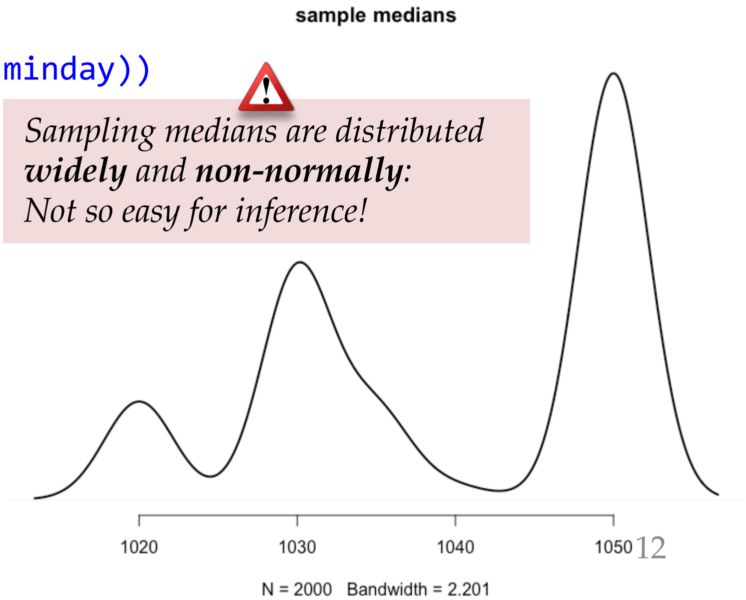
We can estimate the population ***mean*** *within a 2-3 minute interval!*

Bootstrapped medians

```
sample_medians <- replicate(num_boot, sample_statistic(median, minday))
plot(density(sample_medians), lwd=2, main="sample medians")
quantile(sample_medians, probs = c(0.025, 0.975))
  # 2.5% 97.5%
  # 1020  1050
```

The population ***median*** *might be in a 30 minute interval…*

Sampling medians are distributed ***widely*** *and* ***non-normally****:*
Not so easy for inference!



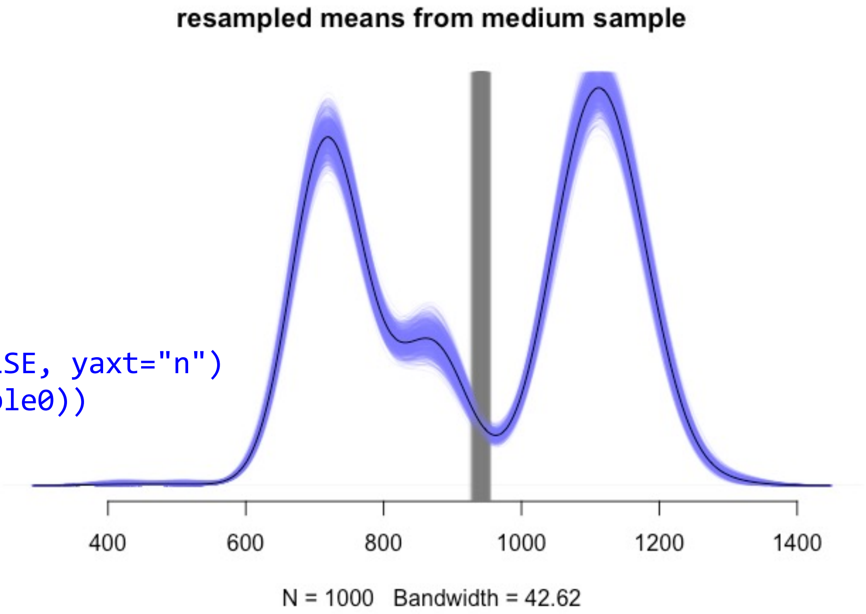
How would you respond to:
"What is the 95% CI of the median?"

Bootstrapping and Standard Error

Sample size and Standard error

```r
plot_resample <- function(stat, sample0) {
  resample <- sample(sample0, length(sample0), replace=TRUE)
  lines(density(resample), col=rgb(0.5,0.5,1, 0.1))
  resample_stat <- stat(resample)
  abline(v=resample_stat, col=rgb(0.5, 0.5, 0.5, 0.1))
}

show_resample_width <- function(sample0, title) {
  num_bootstraps = 1000
  plot(density(sample0), lwd=0, title, ylab="", frame.plot=FALSE, yaxt="n")
  sample_means <- replicate(num_bootstraps, plot_resample(sample0))
  lines(density(sample0), lwd=1, col="black")
}
```

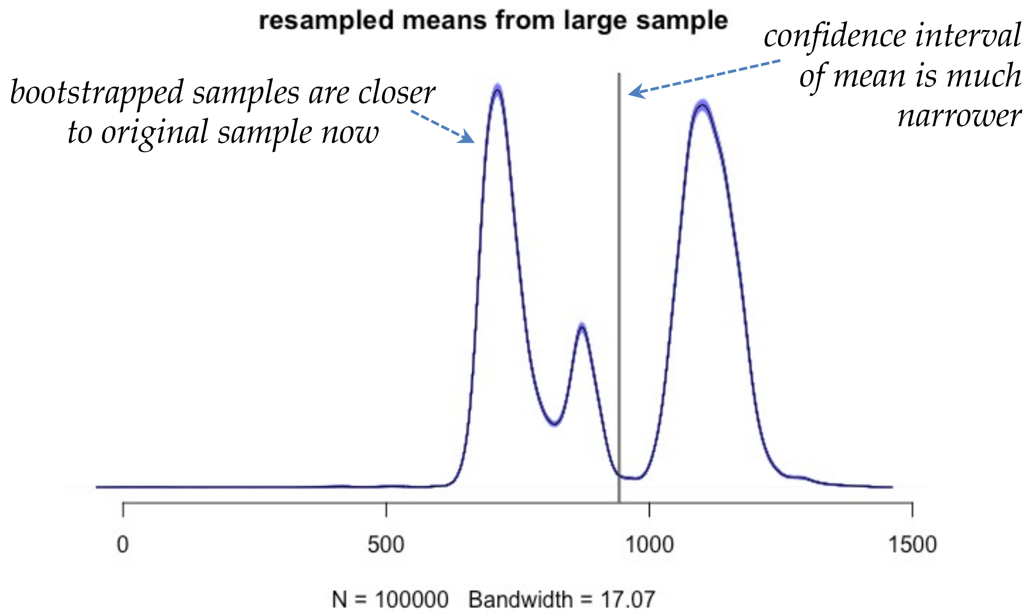


Bootstrapping from a medium-sized sample

```r
show_resample_width(mean, sample(minday, 1000))
```

Bootstrapping from a large sample

```r
show_resample_width(mean, minday)
```



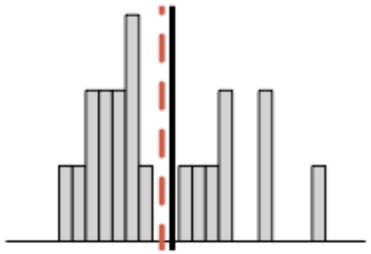
This demonstrates
Standard error:
(standard deviation of sampling means)

$$s_{\bar{x}} = s / \sqrt{n}$$

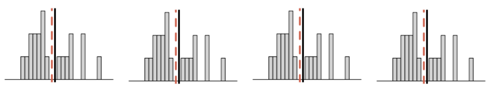
Bootstrapping

Non-parametric Bootstrap

Start from <u>Sample Data</u>



resample data with replacement
from original sample

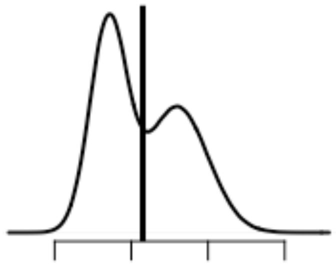


Uses best known estimate
of unknown population distribution:
the sample distribution!

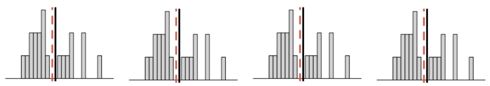
Not the best choice for
small samples, normal distributions

Parametric Bootstrap

Assume <u>Population Distribution</u>



simulate new samples
from distribution parameters

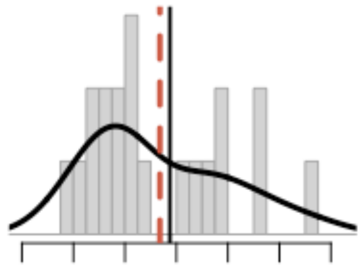


Very precise when population
distribution is well known
(e.g., residuals)

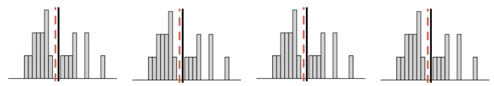
Must have strong reason to
know population distribution

Smoothed Bootstrap

Use <u>Density Function of Sample Data</u>



simulate new samples
from sample's density function



Compromise between
non-parametric and parametric

Does not generalize well
to multivariate or categorical data

Classical Hypothesis Testing: *t-values, p-values*

The *credit manager* of a department store **claims** that their average credit balance of their account holders customers is **$410**.
An *independent auditor* wants to confirm that the credit manager is keeping accurate records.

The auditor carefully **examines 180 accounts at random**, and calculates they have a **mean balance of $507.47**, with **standard deviation of $177.84**.

At 95% confidence, should the auditor believe that the credit manager's estimate is accurate?

Manager's (Hypothesized) Population Claim:

```
hypmanager_hyp <- 410
```

Auditor's sample:

```
auditor_sample <- read.csv("audit.txt")$audit
sample_size <- length(auditor_sample)          # 180
sample_mean <- mean(auditor_sample)            # 507.47
sample_sd   <- sd(auditor_sample)              # 177.84
```

The Test

Standard Error:

```
se <- (sample_sd /sqrt(sample_size))
[1] 13.25578
```

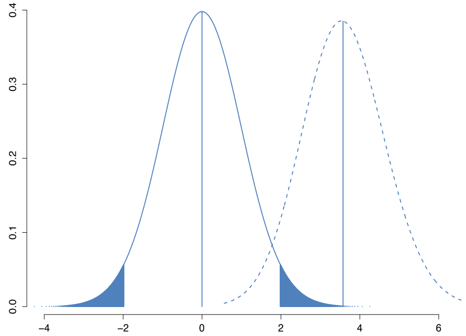
T-statistic

```
t <- (sample_mean - manager_hyp) / se
[1] 7.352796
```

p-value: Probability of t

```
df <- sample_size - 1
p <- 1 - pt(t, df)
[1] 3.349099e-12
```

t-intervals (and p-values) are poor for skewed data
Use the bootstrapped intervals instead



The p-value is less than 5%, so we can reject the manager's estimate at 95% confidence

Bootstrapping the *mean differences*

Let's create some fictional data for our earlier problem:

```
set.seed(50)
pop <- rnorm(100000, mean=511, sd=183)
# Manager's hypothesis: μ=140
```

Claim and sample

```
manager_hyp <- 410
mean(auditor_sample)   # [1] 507.47
```

Difference between auditor's mean and manager's claim

(should be close to zero if they agree!)

```
mean(auditor_sample) - manager_hyp
# [1] 97.46706
```

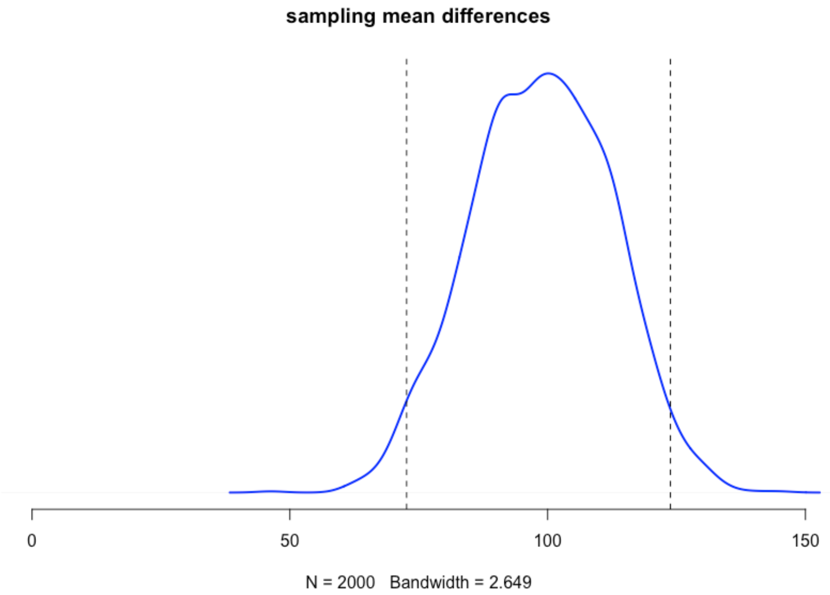
Bootstrapping the *95% CI of the Difference of Means*

```
boot_mean_diffs <- function(sample0, mean_hyp) {
  resample <- sample(sample0, length(sample0), replace=TRUE)
  return( mean(resample) - mean_hyp )
}

set.seed(42379878)
num_boots <- 2000
mean_diffs <- replicate(
  num_boots,
  boot_mean_diffs(auditor_sample, manager_hyp)
)

diff_ci_95 <- quantile(mean_diffs, probs=c(0.025, 0.975))
#      2.5%      97.5%
# 71.37672  124.00195
```

*The **95% CI of the difference** does not contain zero:*
we can reject the manager's claim




```
plot(density(mean_diffs), xlim=c(0,150))
abline(v=diff_ci_95, lty="dashed")
```

test difference

$$t = \frac{(\bar{x} - \mu_0)}{s_{\bar{x}}}$$

Bootstrapping the *t-interval*

Bootstrapping the *standardized difference (t-statistic)*

```
boot_t_stat <- function(sample0, mean_hyp) {
  resample <- sample(sample0, length(sample0), replace=TRUE)
  diff <- mean(resample) - mean_hyp
  se <- sd(resample)/sqrt(length(resample))
  return( diff / se )
}

set.seed(2346786)
num_boots <- 2000
t_boots <- replicate(num_boots, boot_t_stat(auditor_sample, manager_hyp))

mean(t_boots)
# [1] 7.417279
```

Bootstrapped t-statistic

test difference

$$t = \frac{(\bar{x} - \mu_0)}{s_{\bar{x}}}$$

standard error
(standard deviation of the sampling mean)

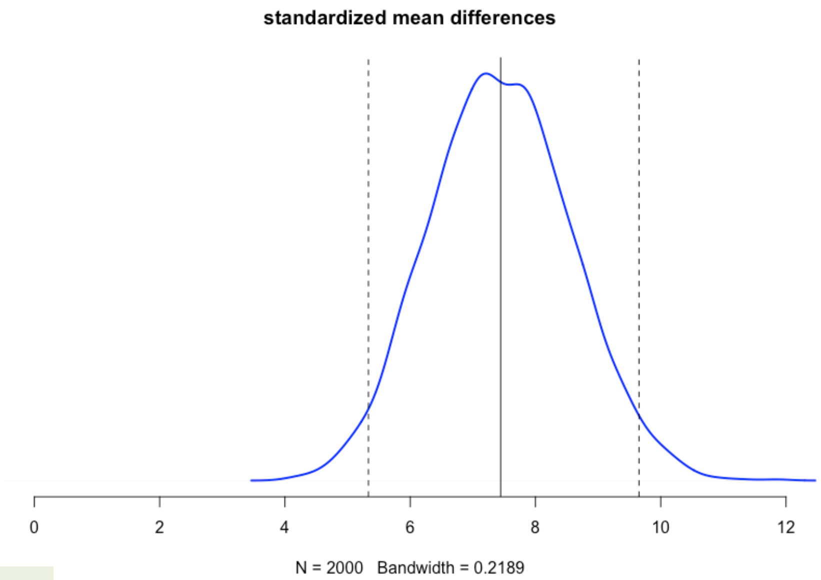
Visualizing the bootstrapped *standardized difference*

```
plot(density(t_boots), xlim=c(0,12), col="blue", lwd=2)


t_ci_95 <- quantile(t_boots, probs=c(0.025, 0.975))
#      2.5%     97.5%
# 5.485465 9.346336


abline(v=mean(t_boots))
abline(v=t_ci_95, lty="dashed")
```

95% CI of t-statistic does not contain zero



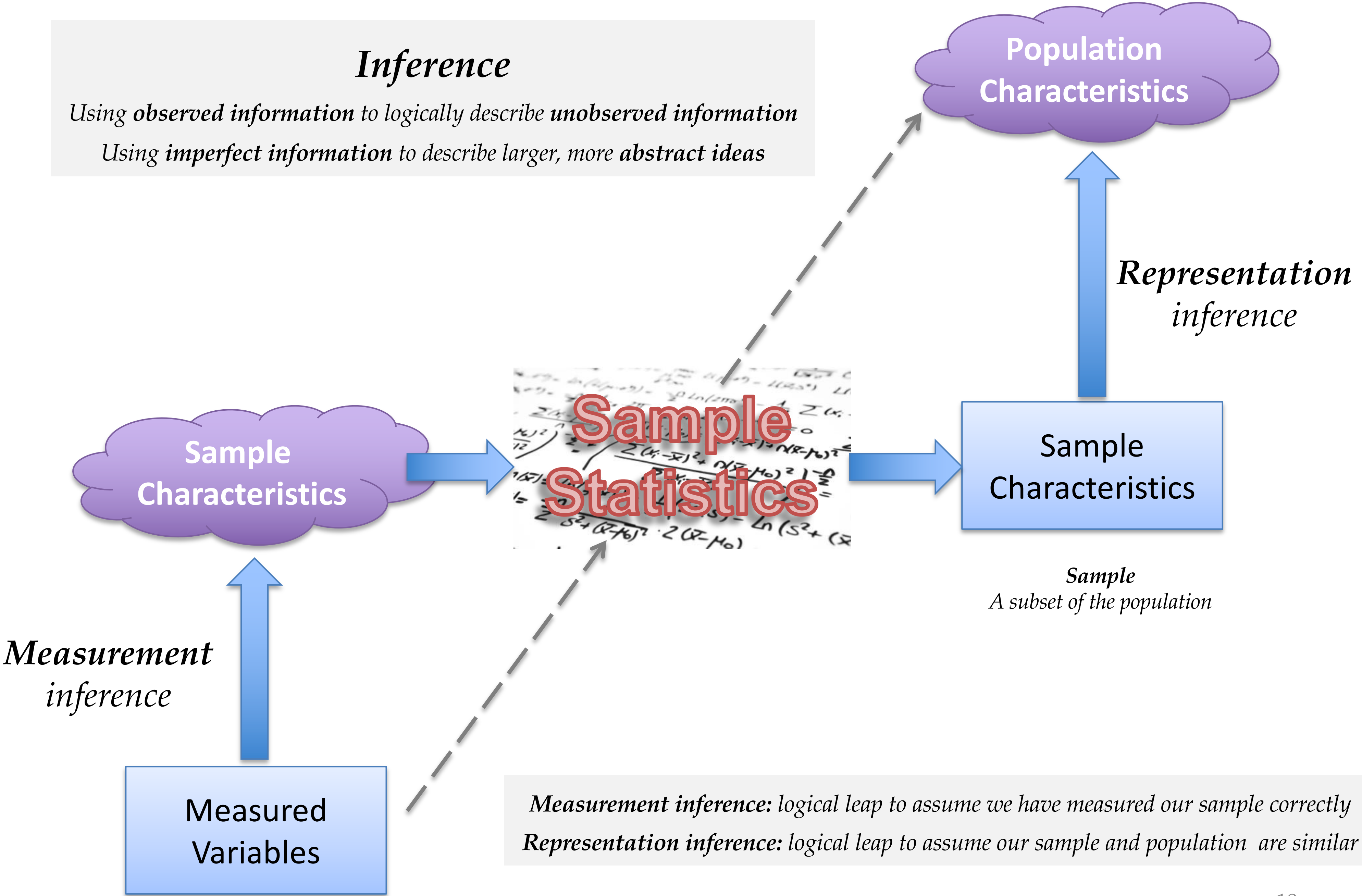
When doing one-sample tests of means,
bootstrapped t-intervals are better than bootstrapped differences,
especially for ***small samples****!*

Statistical Inference: *generalizing from* ***sample to population***

Inference

Using ***observed information*** *to logically describe* ***unobserved information***

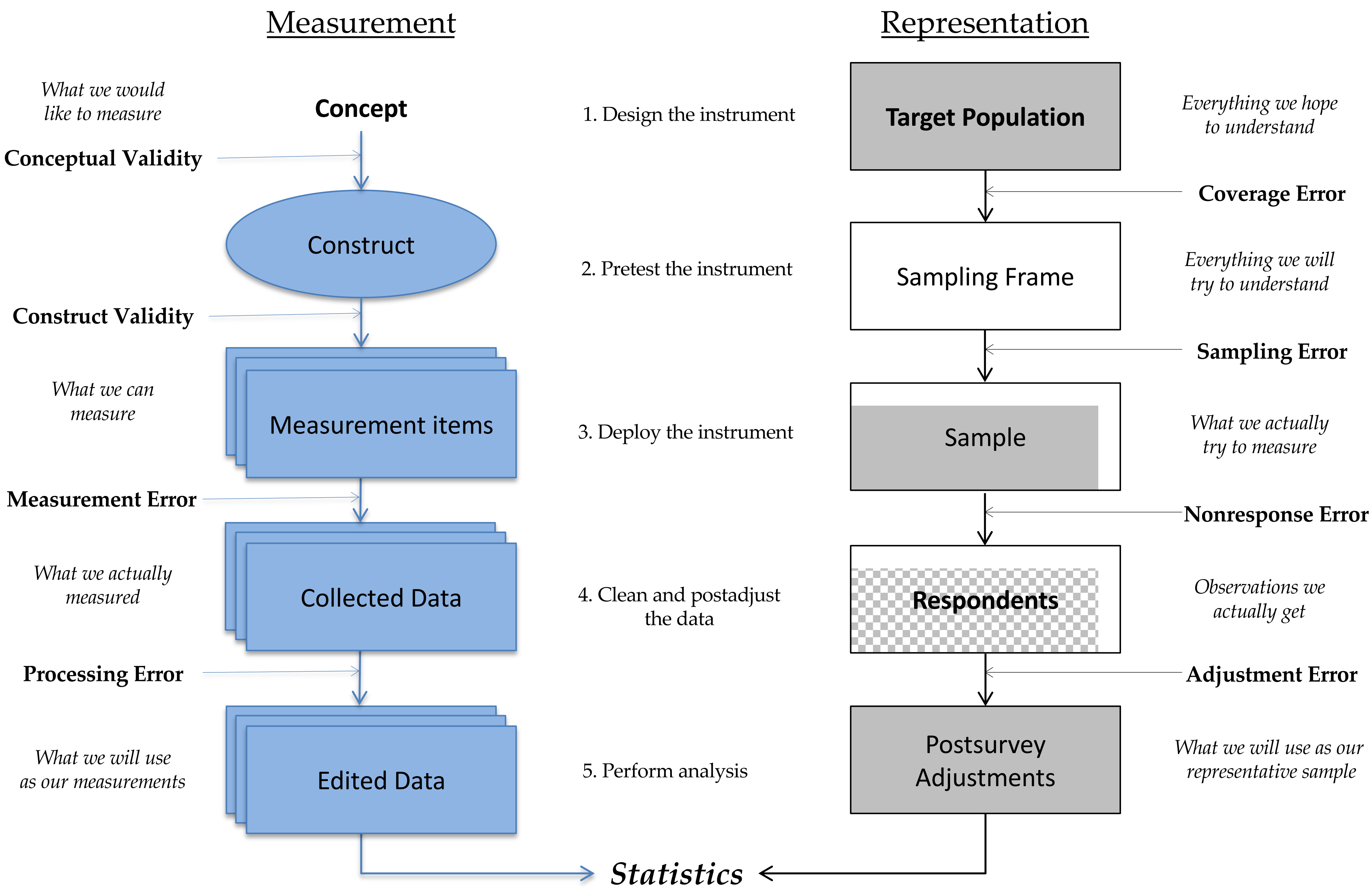
Using ***imperfect information*** *to describe larger, more* ***abstract ideas***



Measurement inference: *logical leap to assume we have measured our sample correctly*

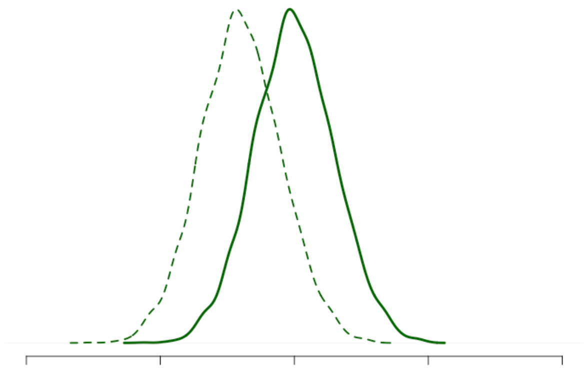
Representation inference: *logical leap to assume our sample and population are similar*

Inference & Error

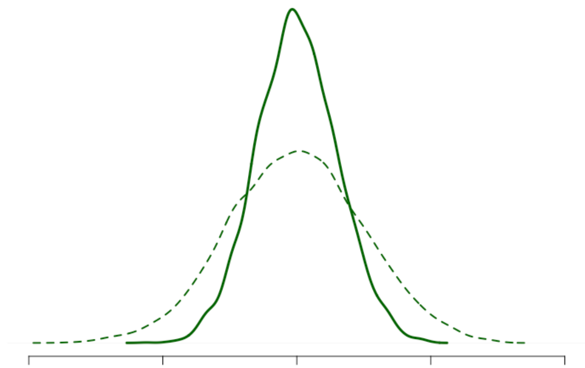


Source: Groves, R. M., Floyd J Fowler, J., Couper, M. P., Lepkowski, J. M., Singer, E., and Tourangeau, R. 2011. Survey Methodology, Wiley.

Error: Bias vs. Variance



$$Y_i \quad = \quad \mu \quad + \quad \varepsilon_i$$



Imagine monitoring usage data of ***users of a mobile game app****,*
for a ***24-hour*** *period*

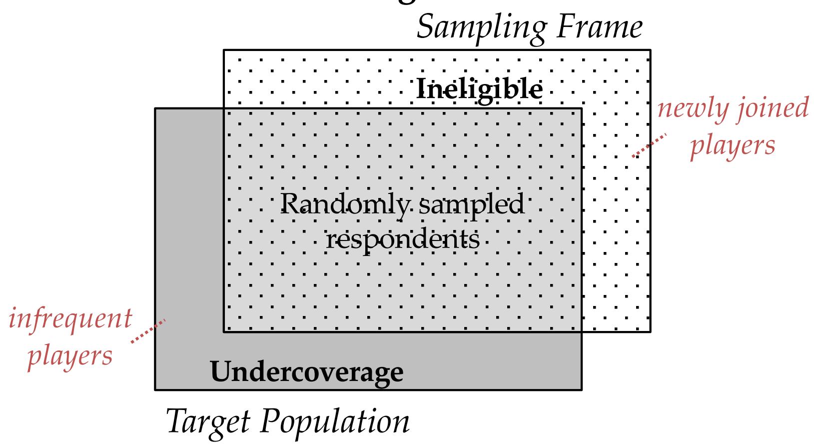
Systematic error ➔ ***Bias***

(errors that tend to agree)

Random error ➔ ***Variance***

(errors that tend to disagree)

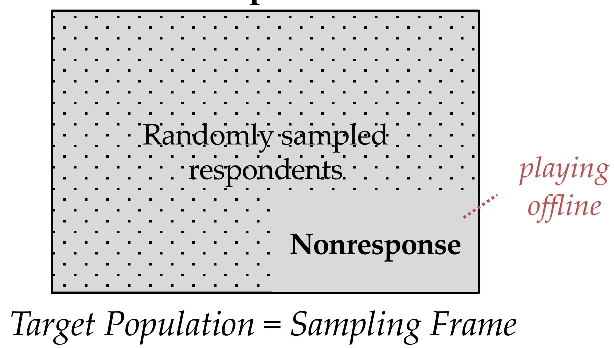
Coverage Error



???

???

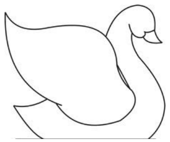
Nonresponse Error



???

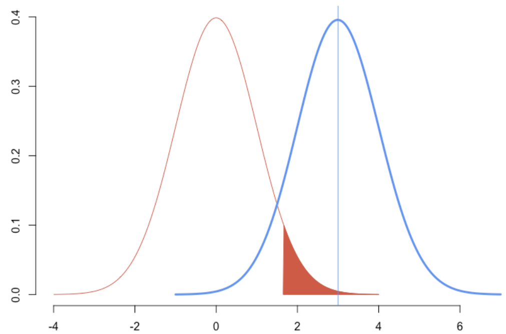
???

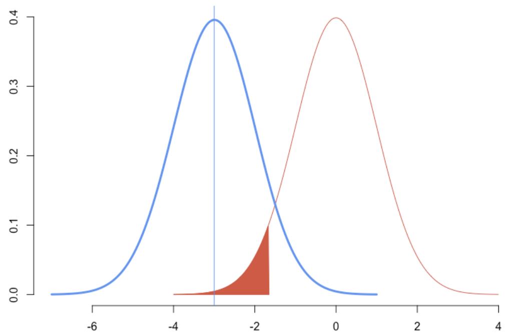
Statistical Inference & **Hypothesis Testing**

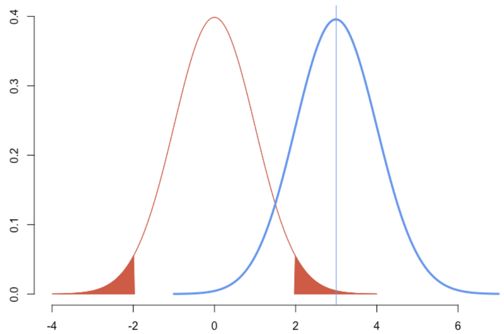


Hypothesis:	**The defendant is guilty!**	If you reject H_{null}	If you cannot reject H_{null}
H_{null}	The defendant is presumed to be innocent		You **find support for H_{null}**
H_{alt}	The defendant is guilty	You **find support for H_{alt}**	

$$t = \frac{\left(\bar{x} - \mu_0\right)}{s_{\bar{x}}}$$





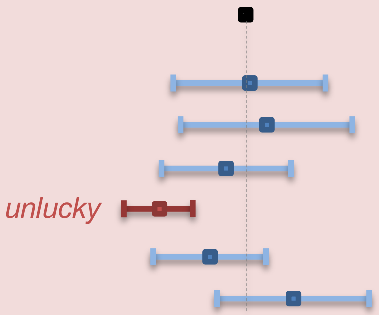
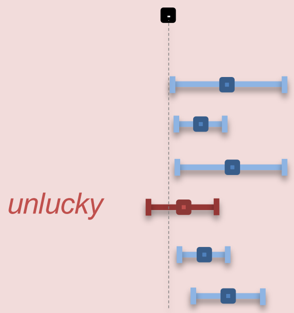


H_{null}	*The average professors spends 8 or fewer hours in the office*	$\mu \leq 8$ hrs
H_{alt}	*The average professor spends more than 8 hours in the office*	$\mu > 8$ hrs

H_{null}	*On average, the factory fills bottles with at least 1 liter*	$\mu \geq 1.00$L
H_{alt}	*On average, the factory fills bottles with less than a liter*	$\mu < 1.00$L

H_{null}	*The average credit balance is $410*	$\mu = 410$
H_{alt}	*The average credit balance is not $410*	$\mu \neq 410$

Type I and Type II Errors

	If H_{null} is really *True*	If H_{null} is really *False*
If evidence says ***reject*** H_{null}	**Type I Error** Probability: α *"Significance Level"* 	**Correct!** Probability: (1-β) ***"Power of the Test"***
If evidence says ***cannot reject*** H_{null}	**Correct!** Probability: 1-α ***"Confidence Level"***	**Type II Error** Probability: β 

🤔

"95% confidence"
Do confidence intervals help unlucky researchers know they are wrong?

`p-value = 0.05`
Does p-value mean you have p% chance of being wrong?

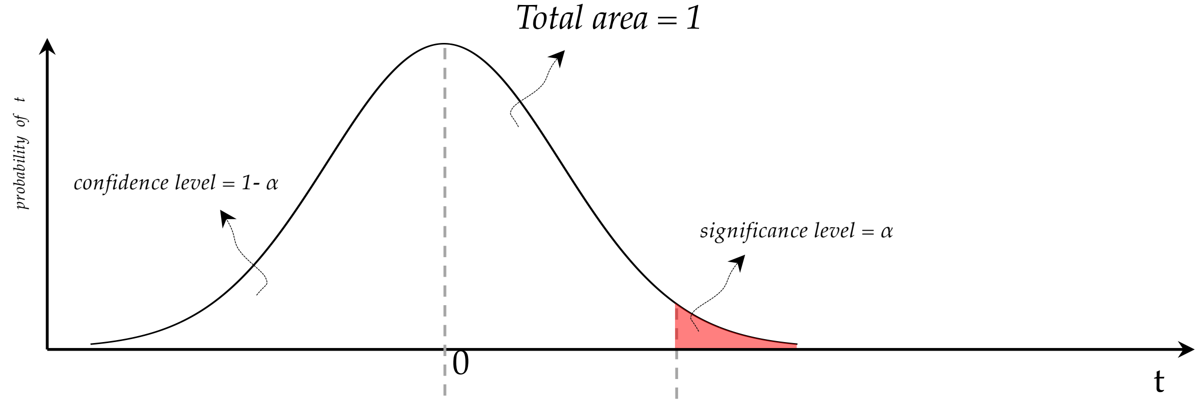
Prior information
What must you know before-hand, to correctly interpret p-value?

Distribution of sampling means

Null hypothesis (H_0): $\mu_0 \leq a$

e.g., average number of people per restaurant booking is 3 persons or less

$$t = \frac{(\bar{x} - \mu_0)}{s_{\bar{x}}} = \frac{(\bar{x} - \mu_0)}{\left(s/\sqrt{n}\right)}$$



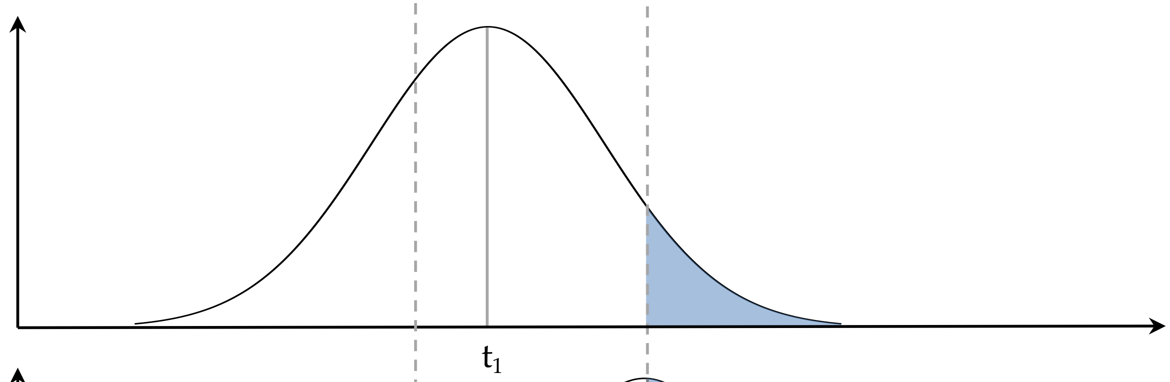
Sample 1:

if we reject H_{null}

$P(\alpha) = P($**type I error**$) =$ $P(1\text{-}\beta) =$ **Power**=

if we don't reject H_{null}

$P(1\text{-}\alpha) = P($"**correct**"$)=$ $P(\beta) = P($**type II error**$) =$



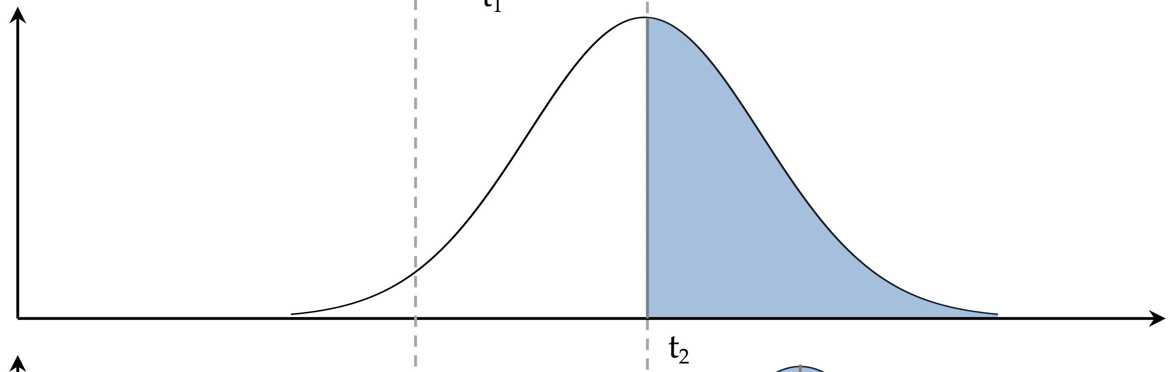
Sample 2:

if we reject H_{null}

$P(\alpha) = P($**type I error**$) =$ $P(1\text{-}\beta) =$ **Power**=

if we don't reject H_{null}

$P(1\text{-}\alpha) = P($"**correct**"$)=$ $P(\beta) = P($**type II error**$) =$



Sample 3:

if we reject H_{null}

$P(\alpha) = P($**type I error**$) =$ $P(1\text{-}\beta) =$ **Power**=

if we don't reject H_{null}

$P(1\text{-}\alpha) = P($"**correct**"$)=$ $P(\beta) = P($**type II error**$) =$