Foundation for Inference

DATA 606 - Statistics & Probability for Data Analytics

Jason Bryer, Ph.D. and Angela Lui, Ph.D.

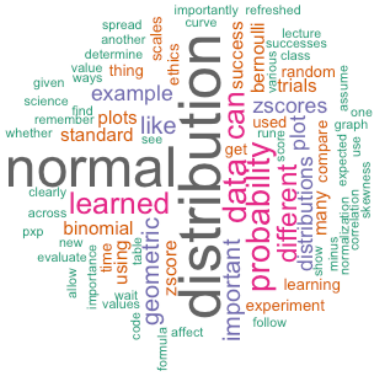
March 9, 2026

Midterm

- There is no meetup next week.
- Will be available March 16th.
- Due March 22nd (by midnight).
- Covers chapters 1 through 5.
- 20 multiple choice questions.
- You may use your notes, textbook, and course site. **Do not consult with anyone else.**

One Minute Paper Results

What was the most important thing you learned during this class?



What important question remains unanswered for you?

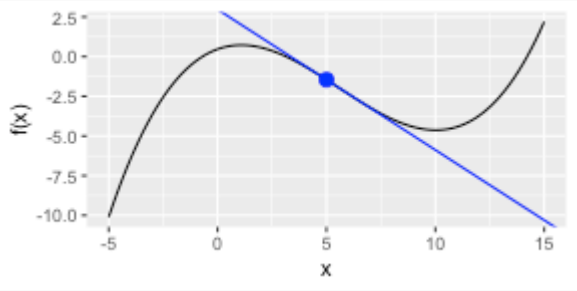
Crash Course in Calculus

Crash Course in Calculus

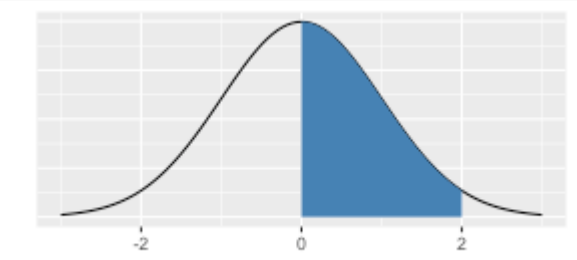
There are three major concepts in calculus that will be helpful to understand:

Limits - the value that a function (or sequence) approaches as the input (or index) approaches some value.

Derivatives - the slope of the line tangent at any given point on a function.

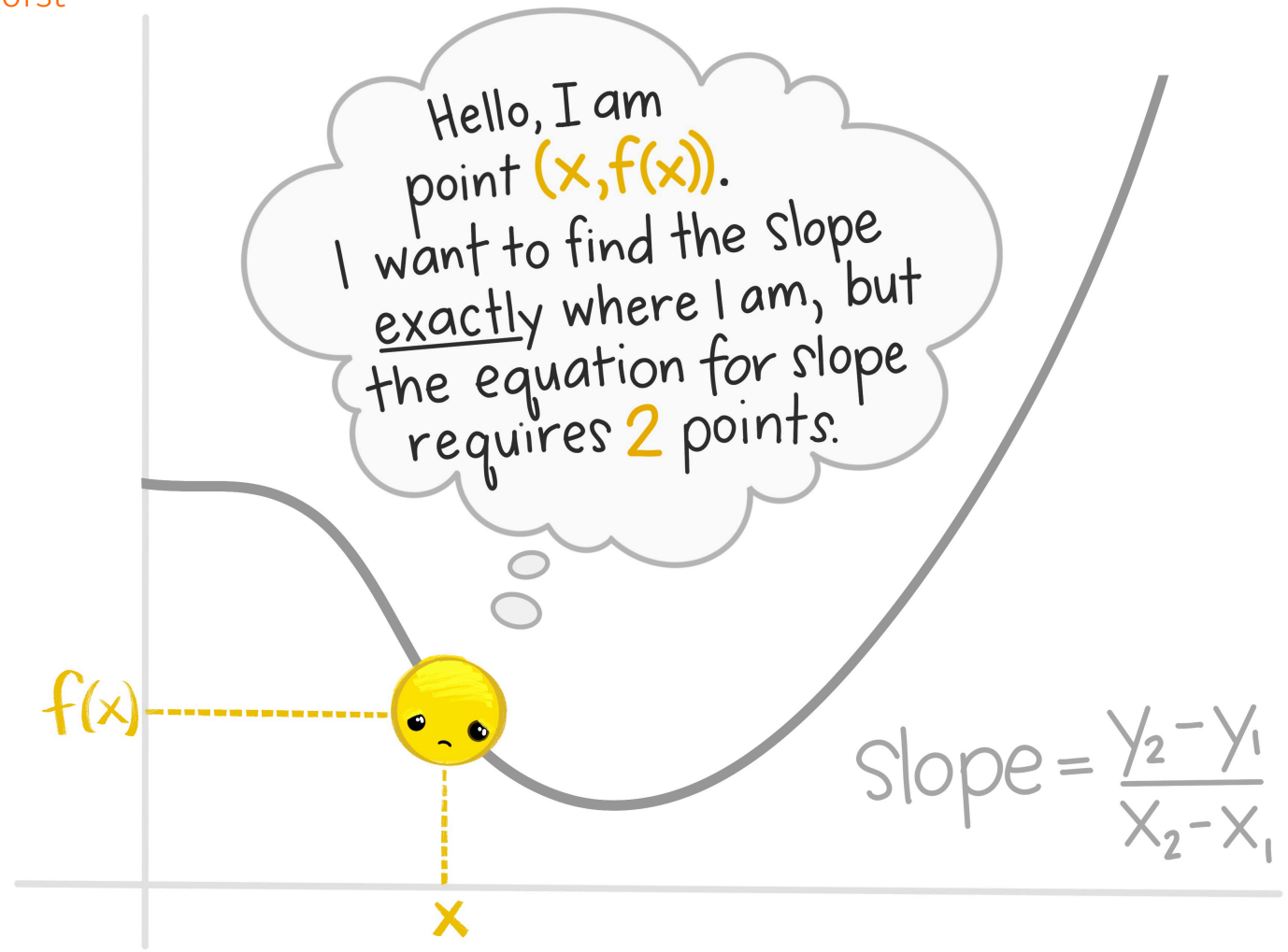


Integrals - the area under the curve.



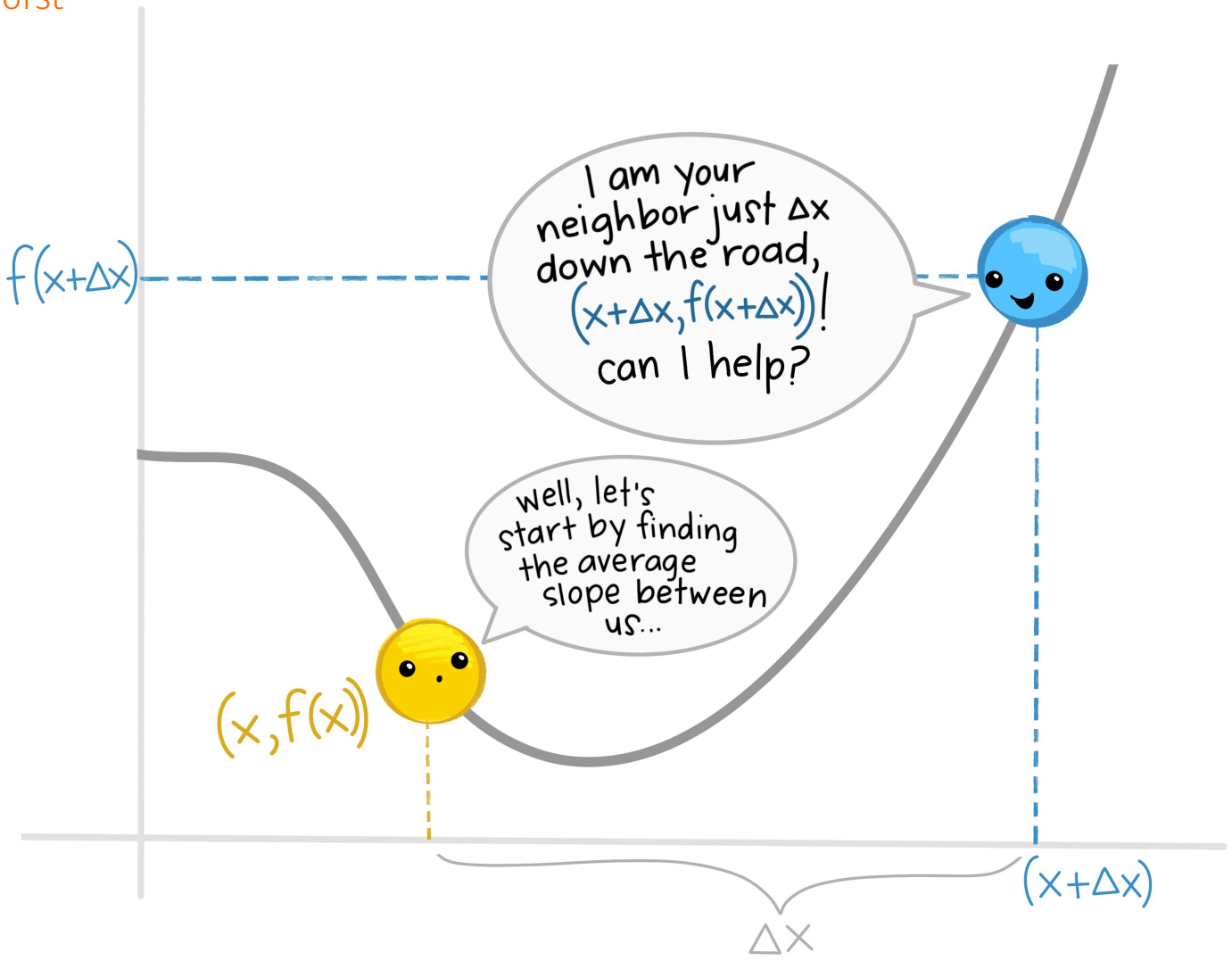
Derivatives

Source: @allison_horst



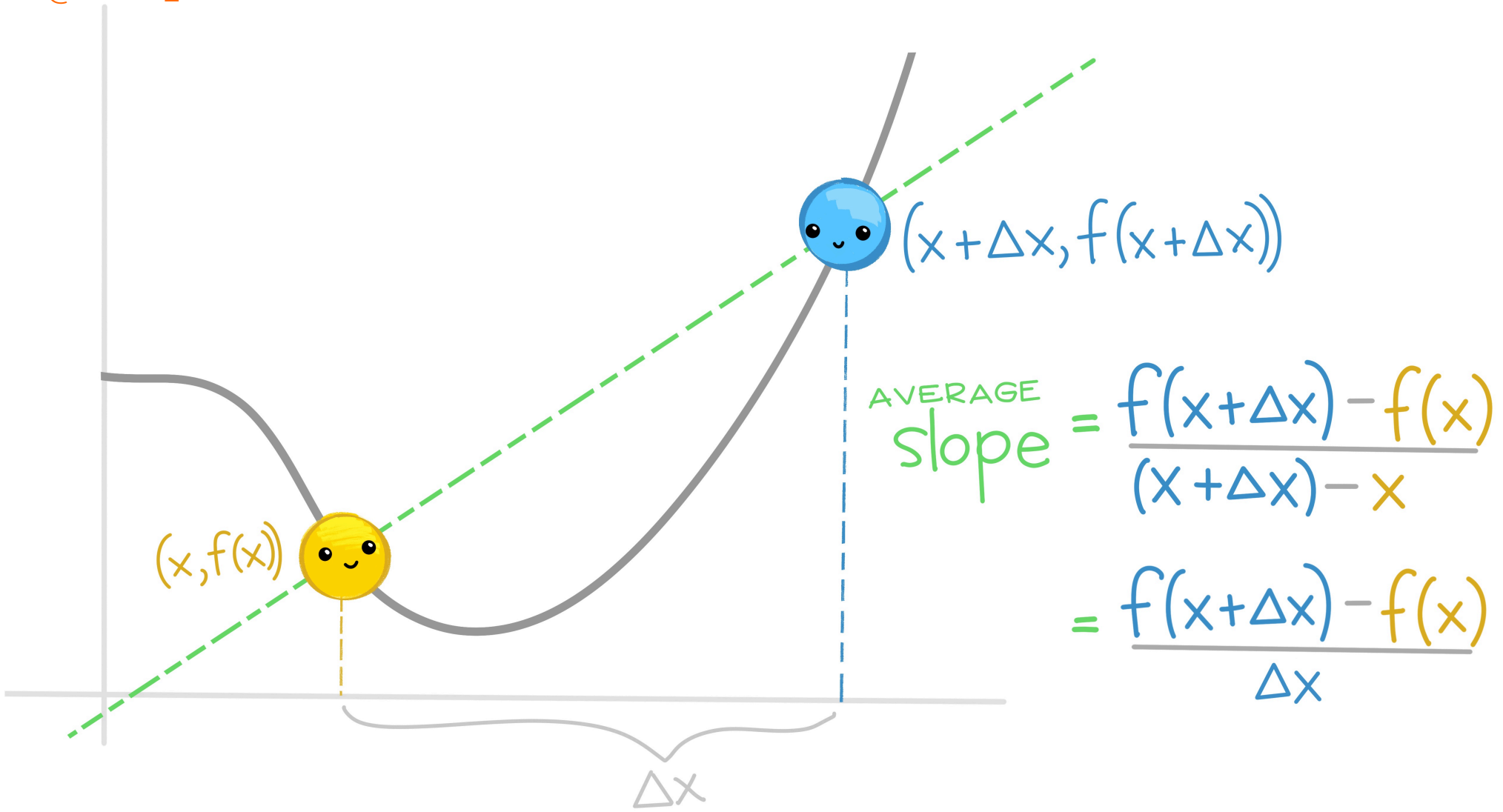
Derivatives

Source: @allison_horst



Derivatives

Source: @allison_horst



Derivatives

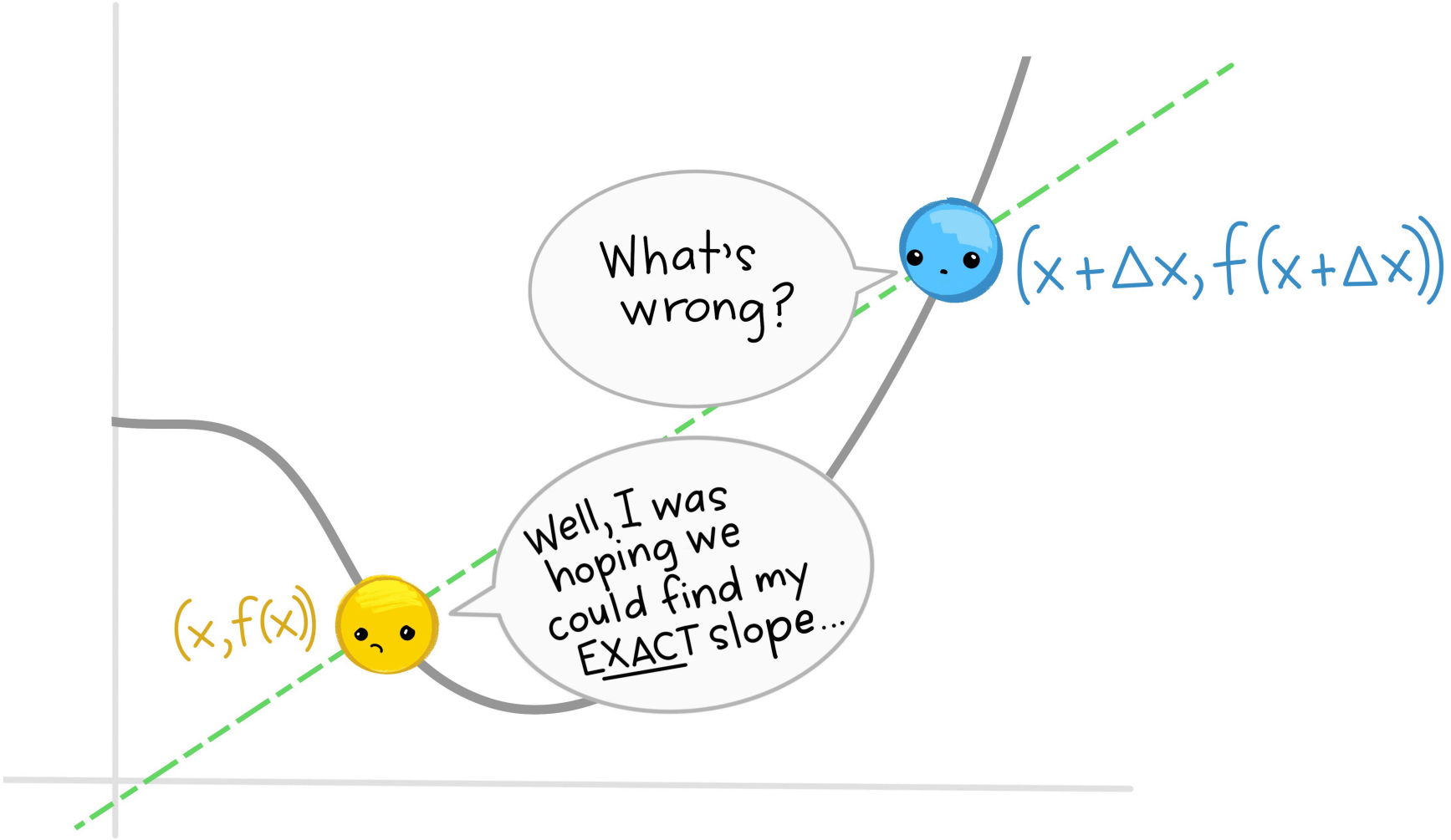
Source: @allison_horst

So: the average slope between ANY 2 POINTS on function $f(x)$ separated by Δx is

$$\boxed{m = \frac{f(x+\Delta x) - f(x)}{\Delta x}}$$

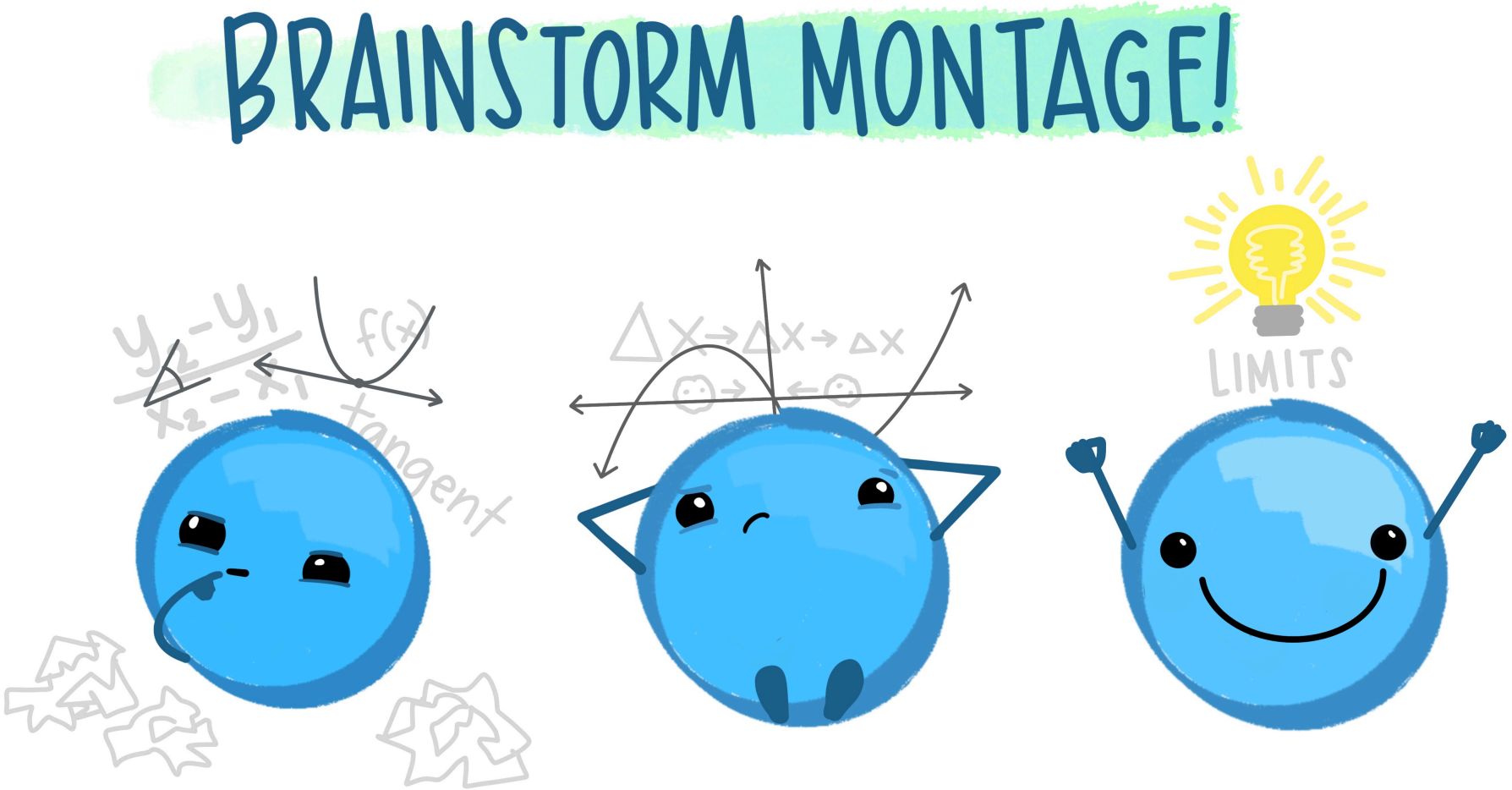
Derivatives

Source: @allison_horst



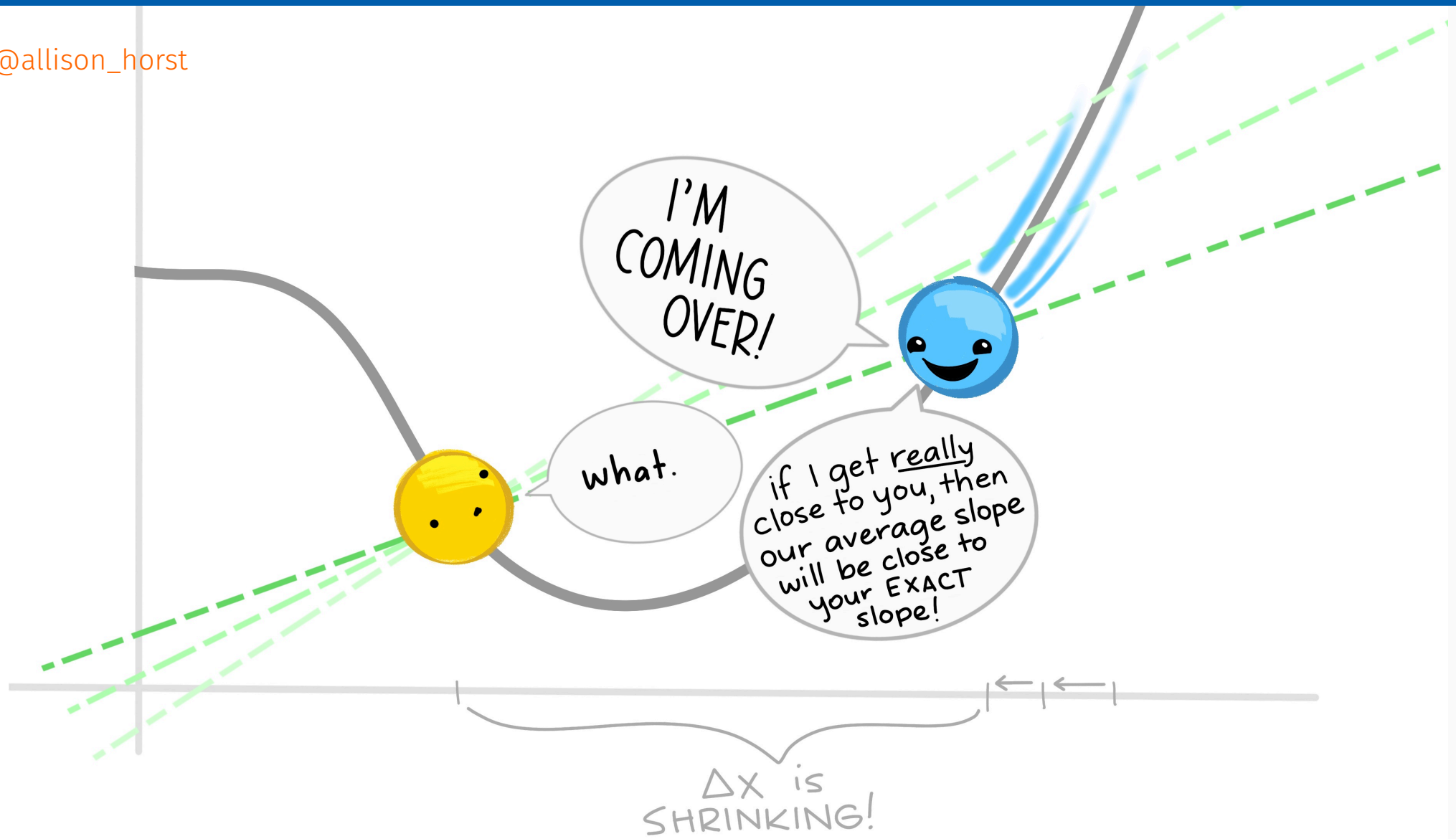
Derivatives

Source: @allison_horst



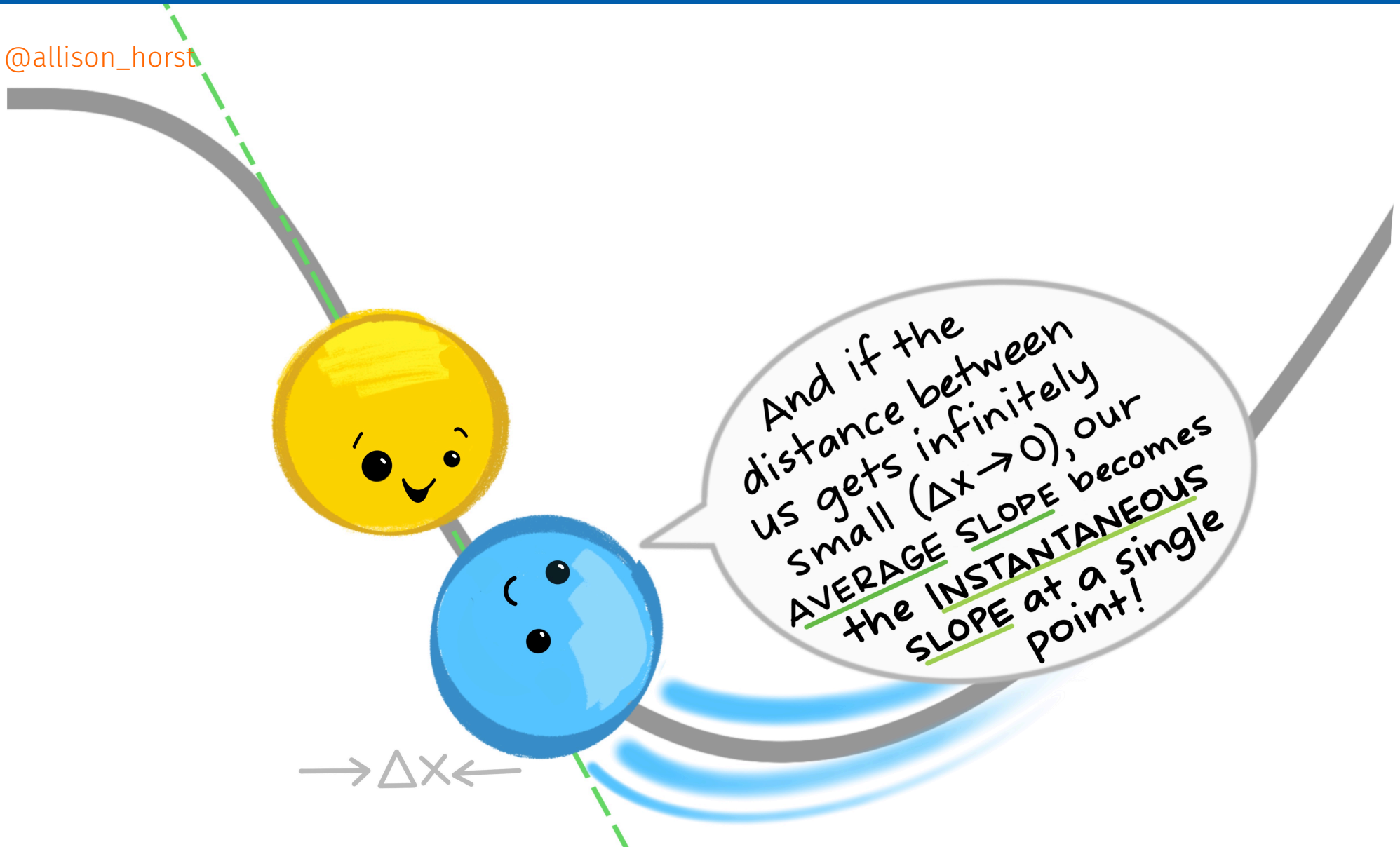
Derivatives

Source: @allison_horst



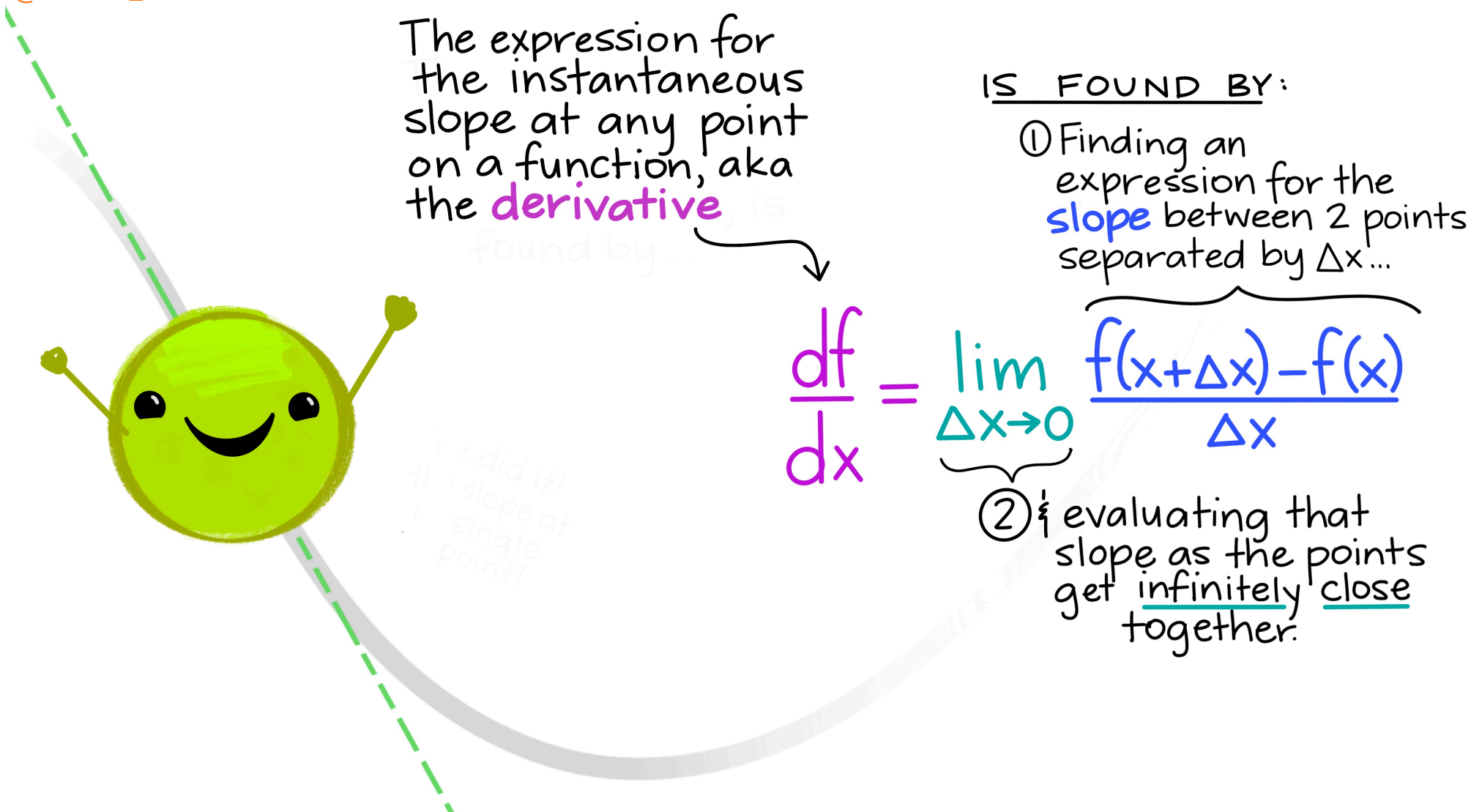
Derivatives

Source: @allison_horst



Derivatives

Source: @allison_horst



Function for Normal Distribution

$$f\left(x|\mu,\sigma\right) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

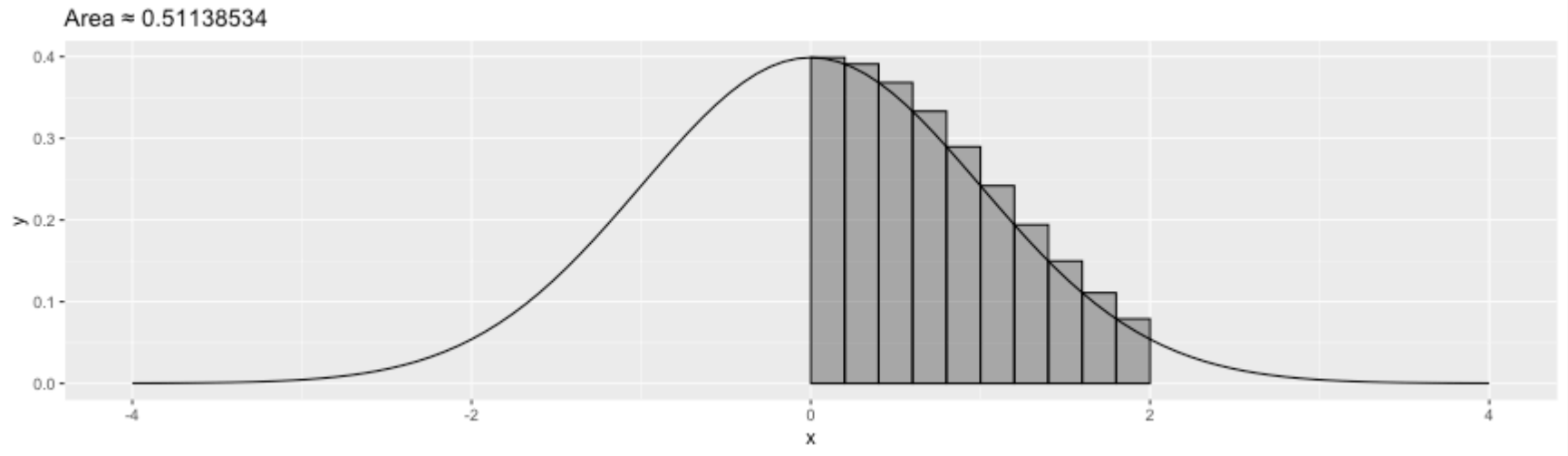
```
f <- function(x, mean = 0, sigma = 1) {
    1 / (sigma * sqrt(2 * pi)) * exp(1)^(-1/2 * ( (x - mean) / sigma )^2)
}
```

```
min <- 0; max <- 2
ggplot() + stat_function(fun = f) + xlim(c(-4, 4)) +
    geom_vline(xintercept = c(min, max), color = 'blue', linetype = 2) + xlab('x')
```

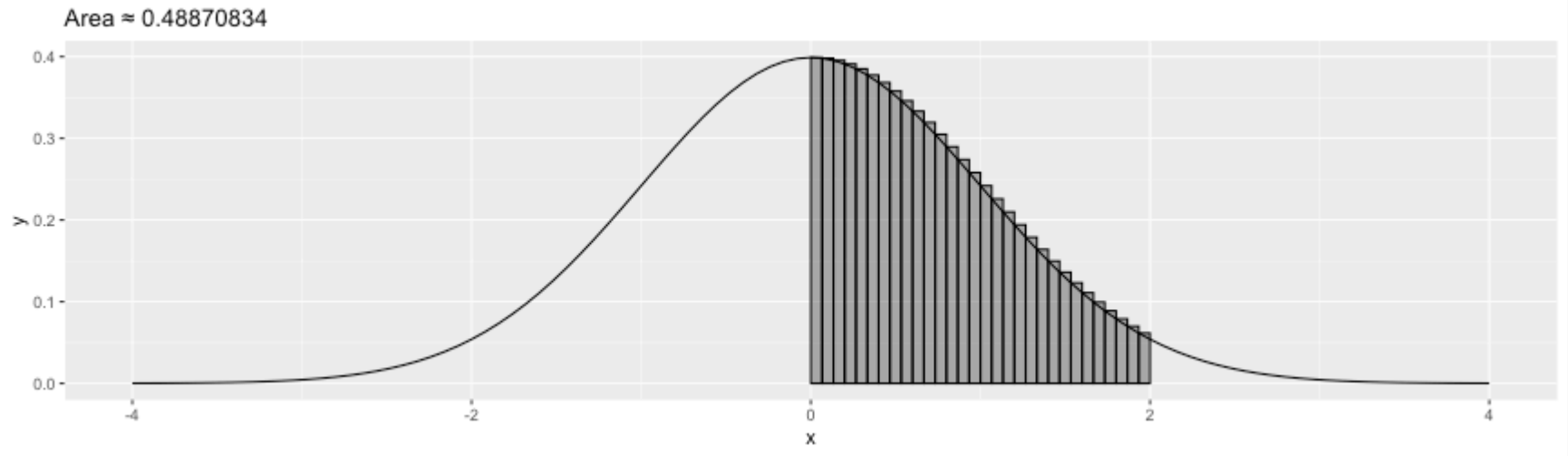
Reimann Sums

One strategy to find the area between two values is to draw a series of rectangles. Given *n* rectangles, we know that the width of each is $\frac{2-0}{n}$ and the height is $f(x)$. Here is an example with 3 rectangles.

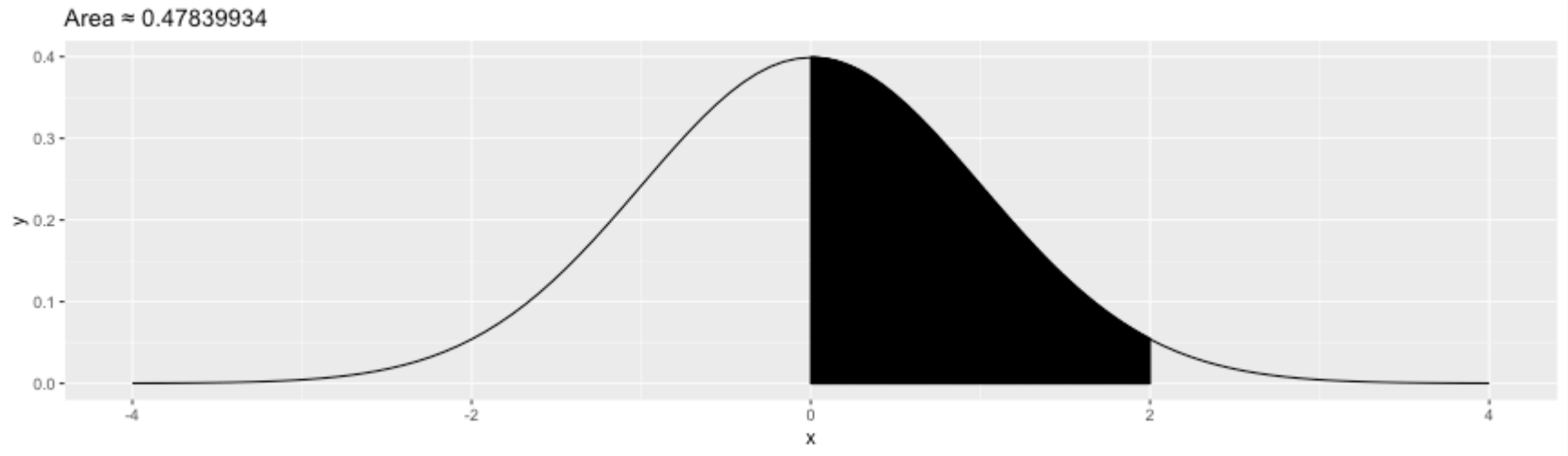
Reimann Sums (10 rectangles)



Reimann Sums (30 rectangles)



Reimann Sums (300 rectangles)



$n \to \infty$

As n approaches infinity we are going to get the *exact* value for the area under the curve. This notion of letting a value get increasingly close to infinity, zero, or any other value, is called the **limit**.

The area under a function is called the integral.

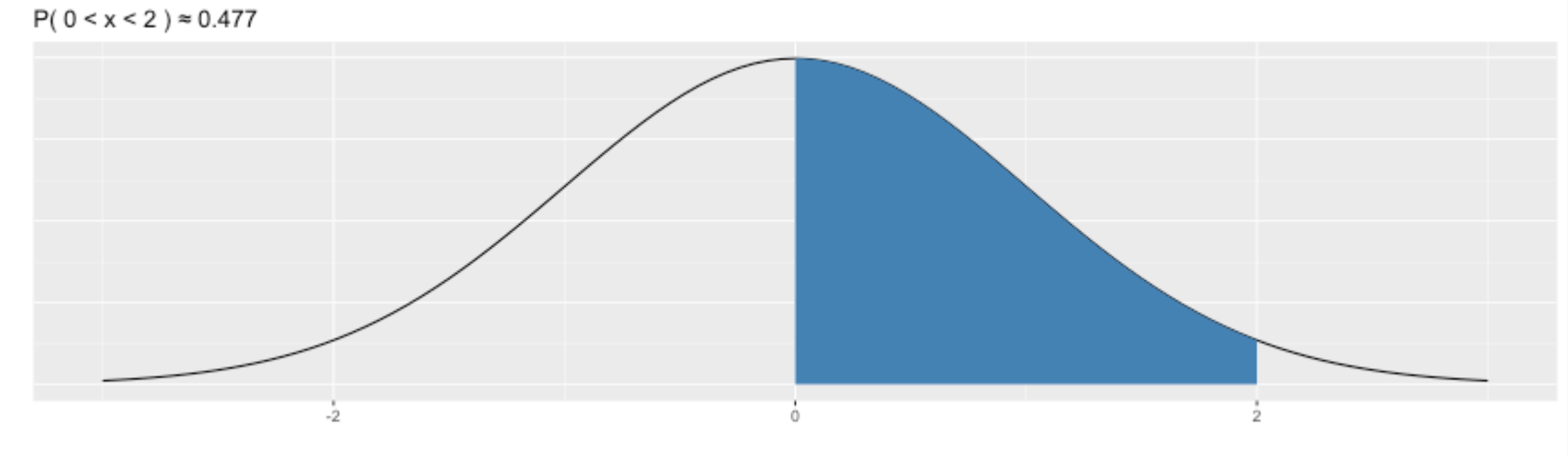
```
integrate(f, 0, 2)
```

```
## 0.4772499 with absolute error < 5.3e-15
```

```
DATA606::shiny_demo('calculus')
```

Normal Distribution

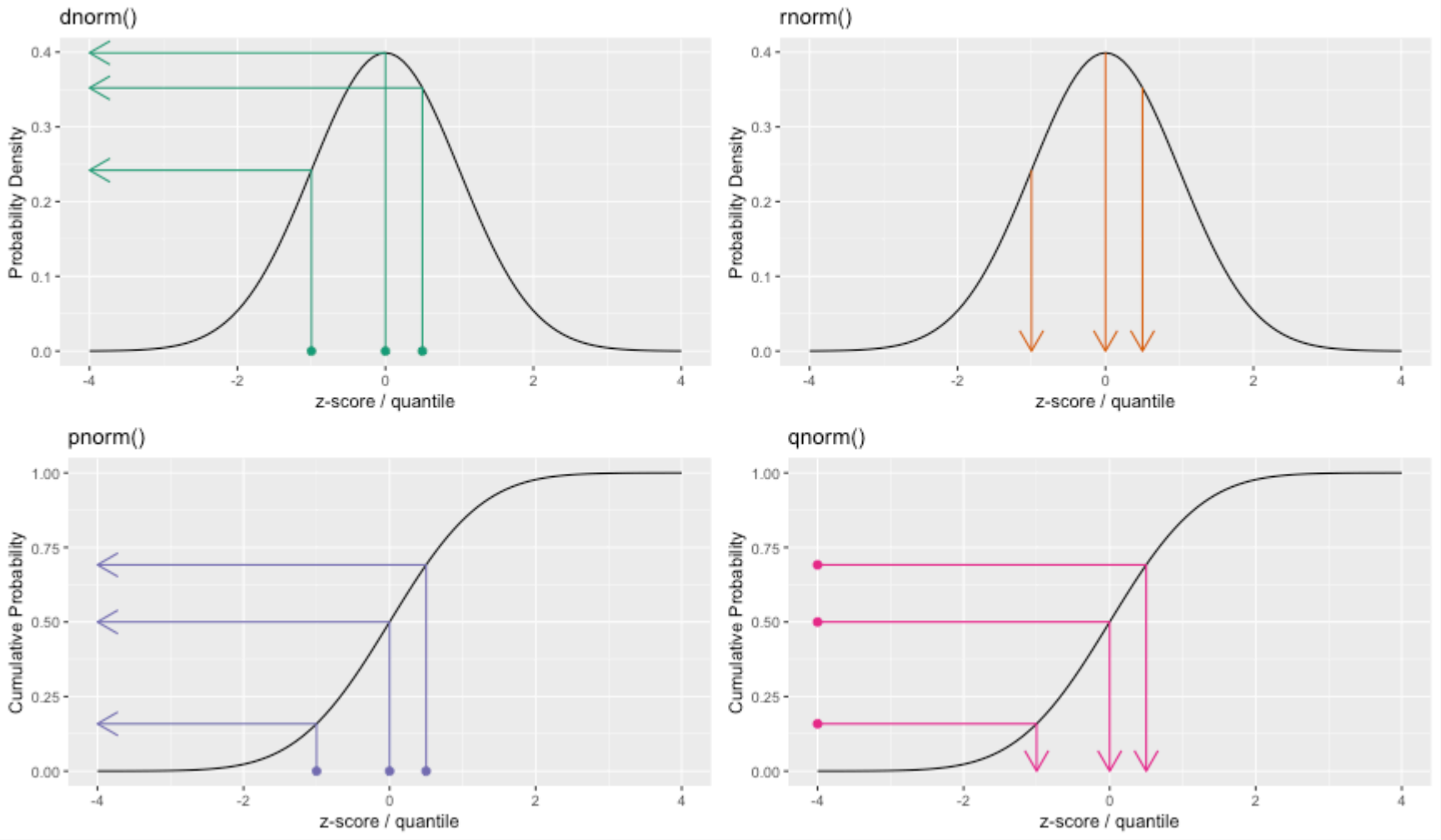
```
normal_plot(cv = c(0, 2))
```




```
pnorm(2) - pnorm(0)
```

```
## [1] 0.4772499
```

R's built in functions for working with distributions



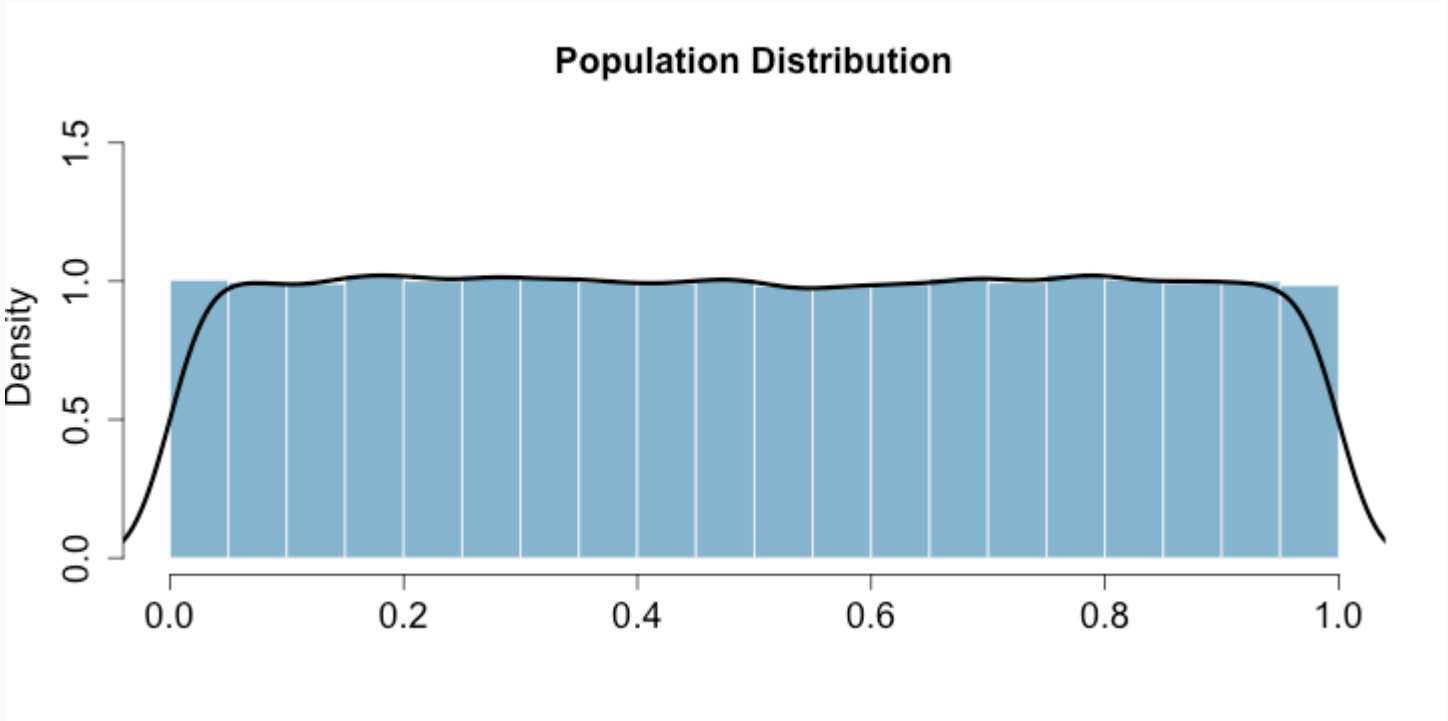

```
VisualStats::distributions_shiny()
```

Foundation for Inference

Population Distribution (Uniform)

```r
n <- 1e5
pop <- runif(n, 0, 1)
mean(pop)
```

```
## [1] 0.4992796
```

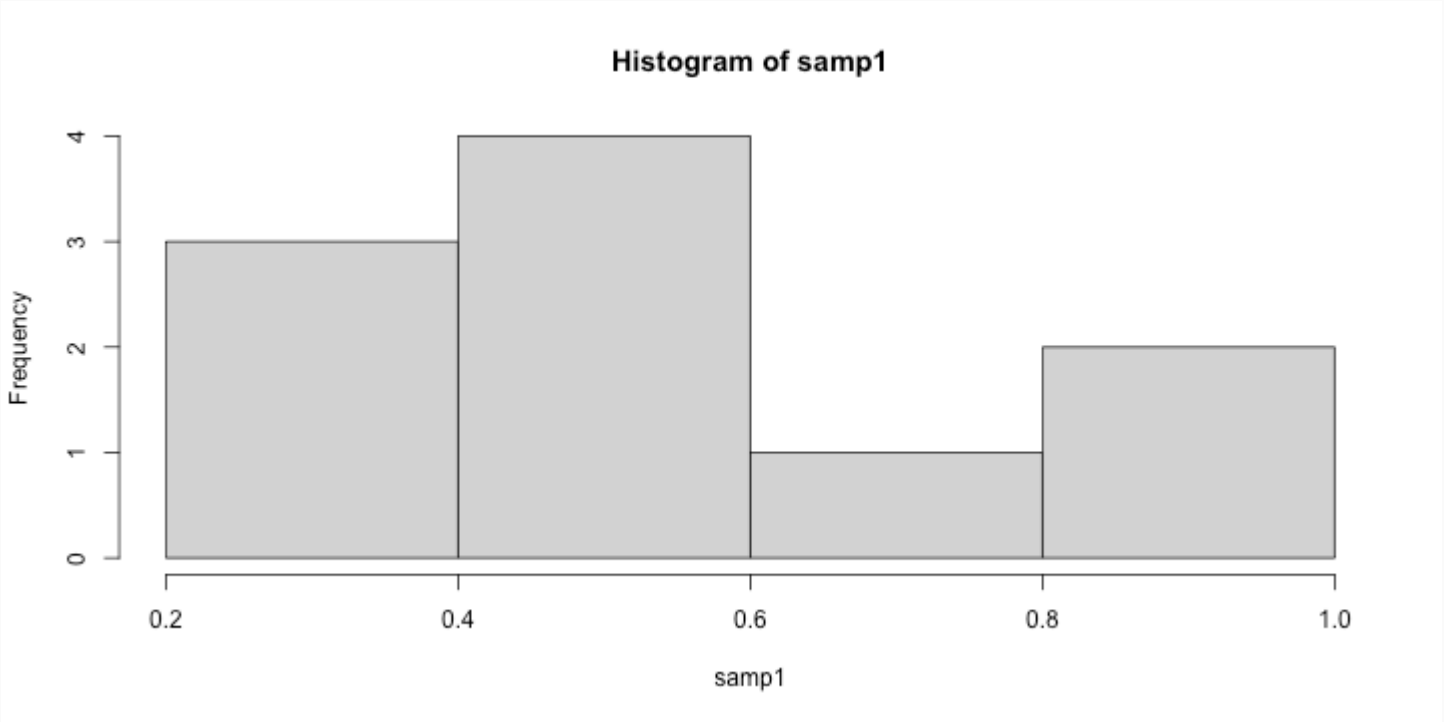


Random Sample (n=10)

```
samp1 <- sample(pop, size=10)
mean(samp1)
```

```
## [1] 0.5379023
```

```
hist(samp1)
```

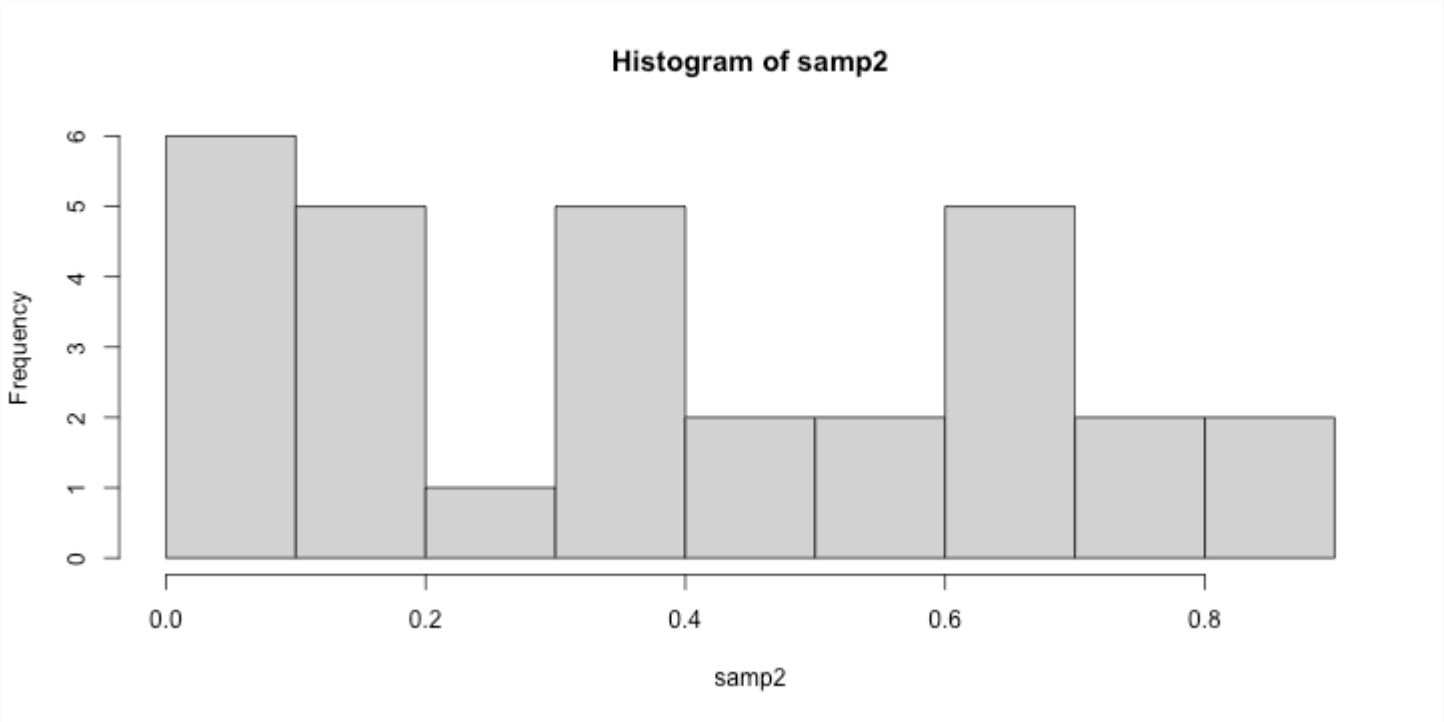


Random Sample (n=30)

```
samp2 <- sample(pop, size=30)
mean(samp2)
```

```
## [1] 0.3745776
```

```
hist(samp2)
```



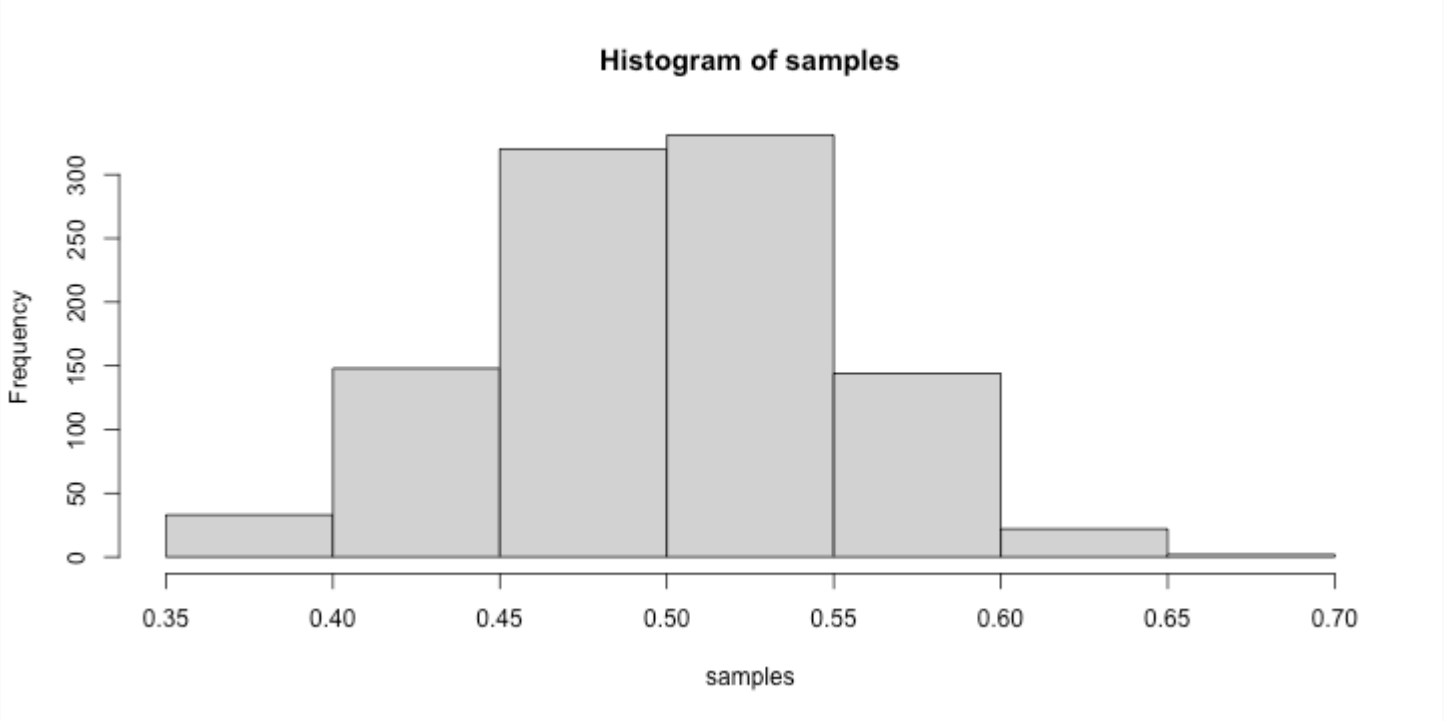
Lots of Random Samples

```r
M <- 1000
samples <- numeric(length=M)
for(i in seq_len(M)) {
    samples[i] <- mean(sample(pop, size=30))
}
head(samples, n=8)
```

```
## [1] 0.4233288 0.4805973 0.4966598 0.4895134 0.5392948 0.4815648 0.5583565
## [8] 0.5591351
```

Sampling Distribution

```
hist(samples)
```



Central Limit Theorem (CLT)

Let X_1, X_2, ..., X_n be independent, identically distributed random variables with mean μ and variance σ^2, both finite. Then for any constant z,

$$\lim_{n \to \infty} P\left(\frac{\bar{X} - \mu}{\sigma / \sqrt{n}} \leq z\right) = \Phi\left(z\right)$$

where Φ is the cumulative distribution function (cdf) of the standard normal distribution.

In other words...

The distribution of the sample mean is well approximated by a normal model:

$$\bar{x} \sim N\left(mean = \mu, SE = \frac{\sigma}{\sqrt{n}}\right)$$

where SE represents the **standard error**, which is defined as the standard deviation of the sampling distribution. In most cases σ is not known, so use s.

CLT Shiny App

```r
library(DATA606)
shiny_demo('sampdist')
shiny_demo('CLT_mean')
```

Standard Error

```r
samp2 <- sample(pop, size=30)
mean(samp2)
```

```
## [1] 0.5250263
```

```r
(samp2.se <- sd(samp2) / sqrt(length(samp2)))
```

```
## [1] 0.056618
```

Confidence Interval

The confidence interval is then $\mu \pm CV \times SE$ where CV is the critical value. For a 95% confidence interval, the critical value is ~1.96 since

$$\int_{-1.96}^{1.96} \frac{1}{\sigma\sqrt{2\pi}} d^{-\frac{(x-\mu)^2}{2\sigma^2}} \approx 0.95$$

```
qnorm(0.025) # Remember we need to consider the two tails, 2.5% to the left, 2.5% to the right.
```

```
## [1] -1.959964
```

```
(samp2.ci <- c(mean(samp2) - 1.96 * samp2.se, mean(samp2) + 1.96 * samp2.se))
```

```
## [1] 0.4140550 0.6359975
```

Confidence Intervals (cont.)

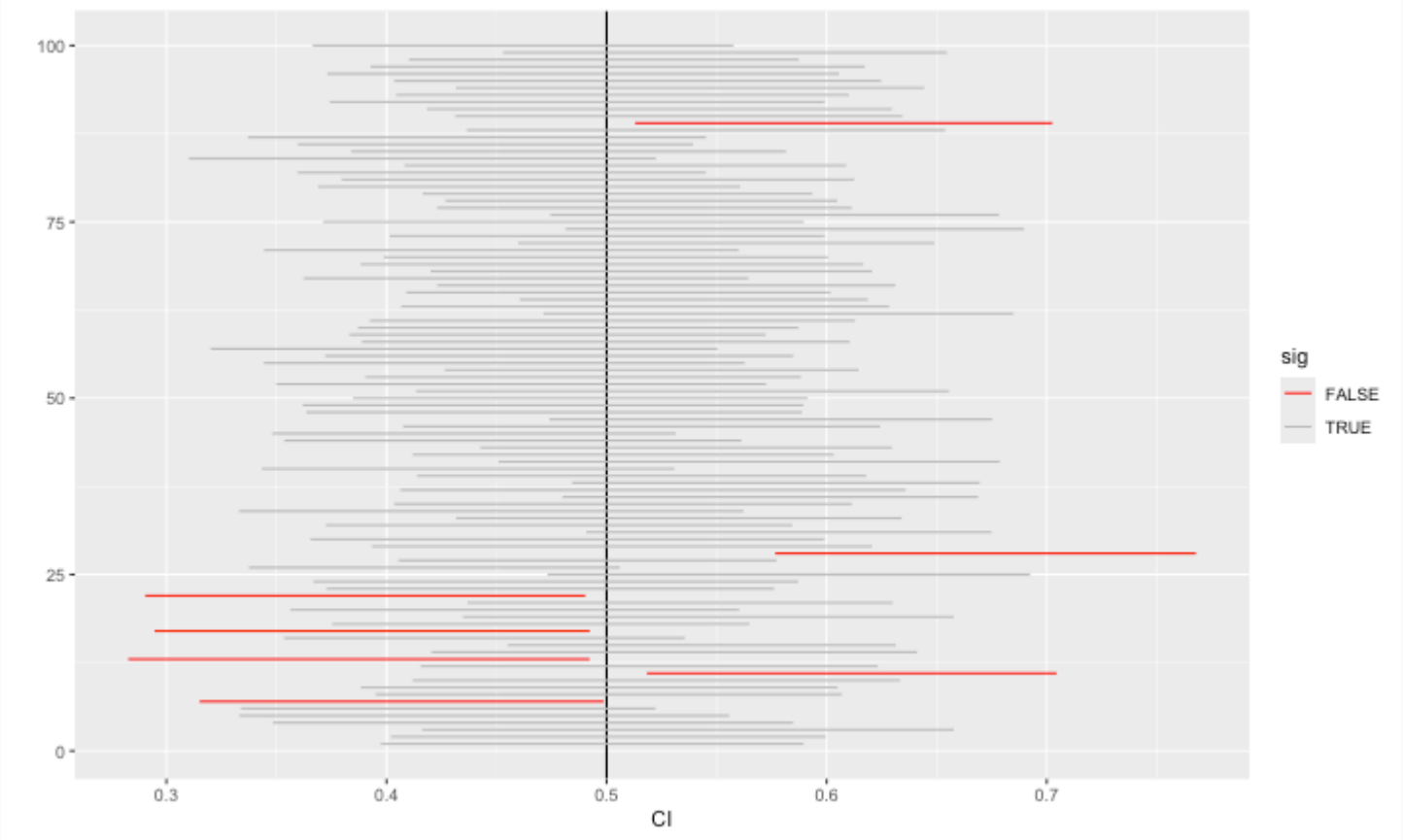
We are 95% confident that the true population mean is between 0.414055, 0.6359975.

That is, if we were to take 100 random samples, we would expect at least 95% of those samples to have a mean within 0.414055, 0.6359975.

```
ci <- data.frame(mean=numeric(), min=numeric(), max=numeric())
for(i in seq_len(100)) {
    samp <- sample(pop, size=30)
    se <- sd(samp) / sqrt(length(samp))
    ci[i,] <- c(mean(samp),
                mean(samp) - 1.96 * se,
                mean(samp) + 1.96 * se)
}
ci$sample <- 1:nrow(ci)
ci$sig <- ci$min < 0.5 & ci$max > 0.5
```

Confidence Intervals

```
ggplot(ci, aes(x=min, xend=max, y=sample, yend=sample, color=sig)) +
    geom_vline(xintercept=0.5) +
    geom_segment() + xlab('CI') + ylab('') +
    scale_color_manual(values=c('TRUE'='grey', 'FALSE'='red'))
```



Null Hypothesis Testing

Hypothesis Testing

- We start with a null hypothesis (H_0) that represents the status quo.
- We also have an alternative hypothesis (H_A) that represents our research question, i.e. what we're testing for.
- We conduct a hypothesis test under the assumption that the null hypothesis is true, either via simulation or traditional methods based on the central limit theorem.
- If the test results suggest that the data do not provide convincing evidence for the alternative hypothesis, we stick with the null hypothesis. If they do, then we reject the null hypothesis in favor of the alternative.

Hypothesis Testing (using CI)

H_0: The mean of `samp2` = 0.5
H_A: The mean of `samp2` $\neq$ 0.5

Using confidence intervals, if the *null* value is within the confidence interval, then we *fail* to reject the *null* hypothesis.

```
(samp2.ci <- c(mean(samp2) - 1.96 * sd(samp2) / sqrt(length(samp2)),
               mean(samp2) + 1.96 * sd(samp2) / sqrt(length(samp2))))
```

```
## [1] 0.4140550 0.6359975
```

Since 0.5 fall within 0.414055, 0.6359975, we *fail* to reject the null hypothesis.

Hypothesis Testing (using p-values)

$$\bar{x} \sim N\left(mean = 0.49, SE = \frac{0.27}{\sqrt{30} = 0.049}\right)$$

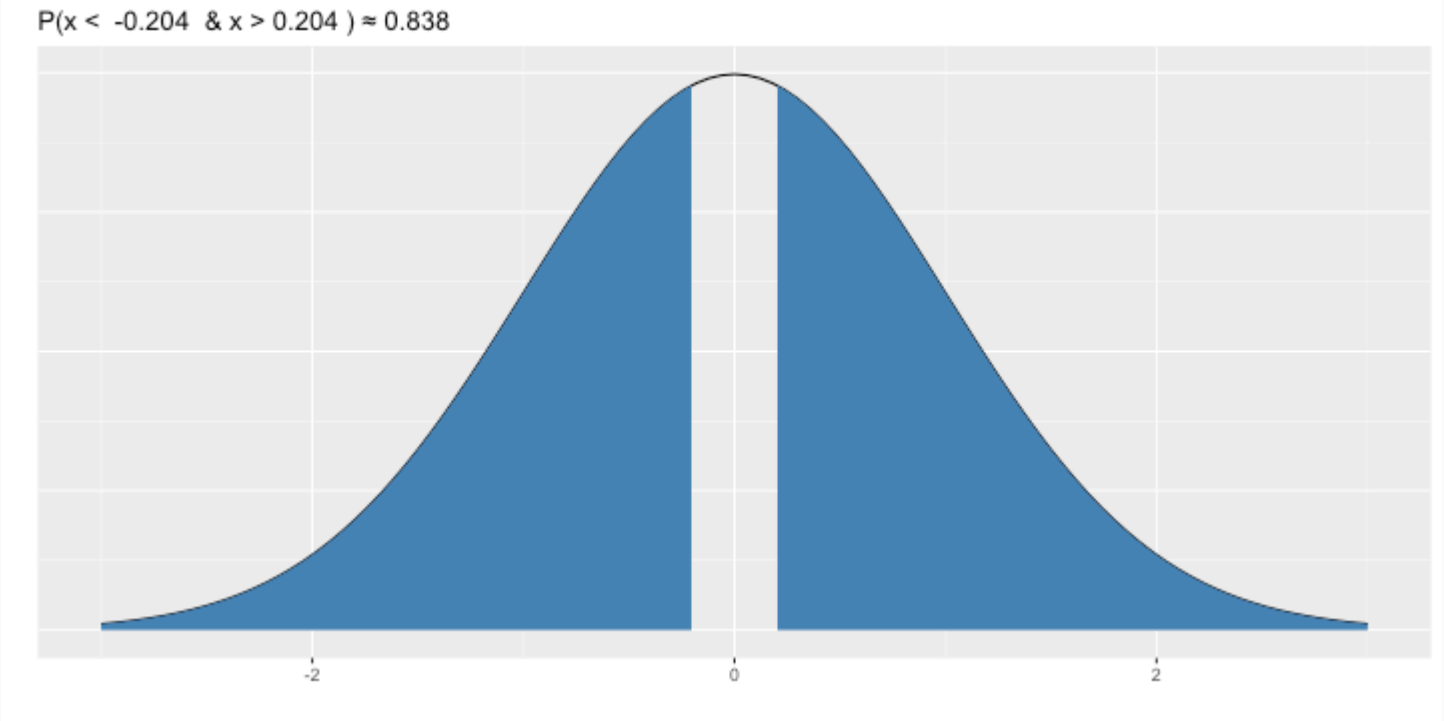
$$Z = \frac{\bar{x} - null}{SE} = \frac{0.49 - 0.50}{0.049} = -.204081633$$

```
pnorm(-.204) * 2
```

```
## [1] 0.8383535
```

Hypothesis Testing (using *p*-values)

```
DATA606::normal_plot(cv = c(.204), tails = 'two.sided')
```



Type I and II Errors

There are two competing hypotheses: the null and the alternative. In a hypothesis test, we make a decision about which might be true, but our choice might be incorrect.

	fail to reject H_0	**reject H_0**
H_0 true	✔	Type I Error
H_A true	Type II Error	✔

- Type I Error: **Rejecting** the null hypothesis when it is **true**.
- Type II Error: **Failing to reject** the null hypothesis when it is **false**.

Hypothesis Test

If we again think of a hypothesis test as a criminal trial then it makes sense to frame the verdict in terms of the null and alternative hypotheses:

$$H_0 : \text{Defendant is innocent}$$
$$H_A : \text{Defendant is guilty}$$

Which type of error is being committed in the following circumstances?

- Declaring the defendant innocent when they are actually guilty

Type 2 error

- Declaring the defendant guilty when they are actually innocent

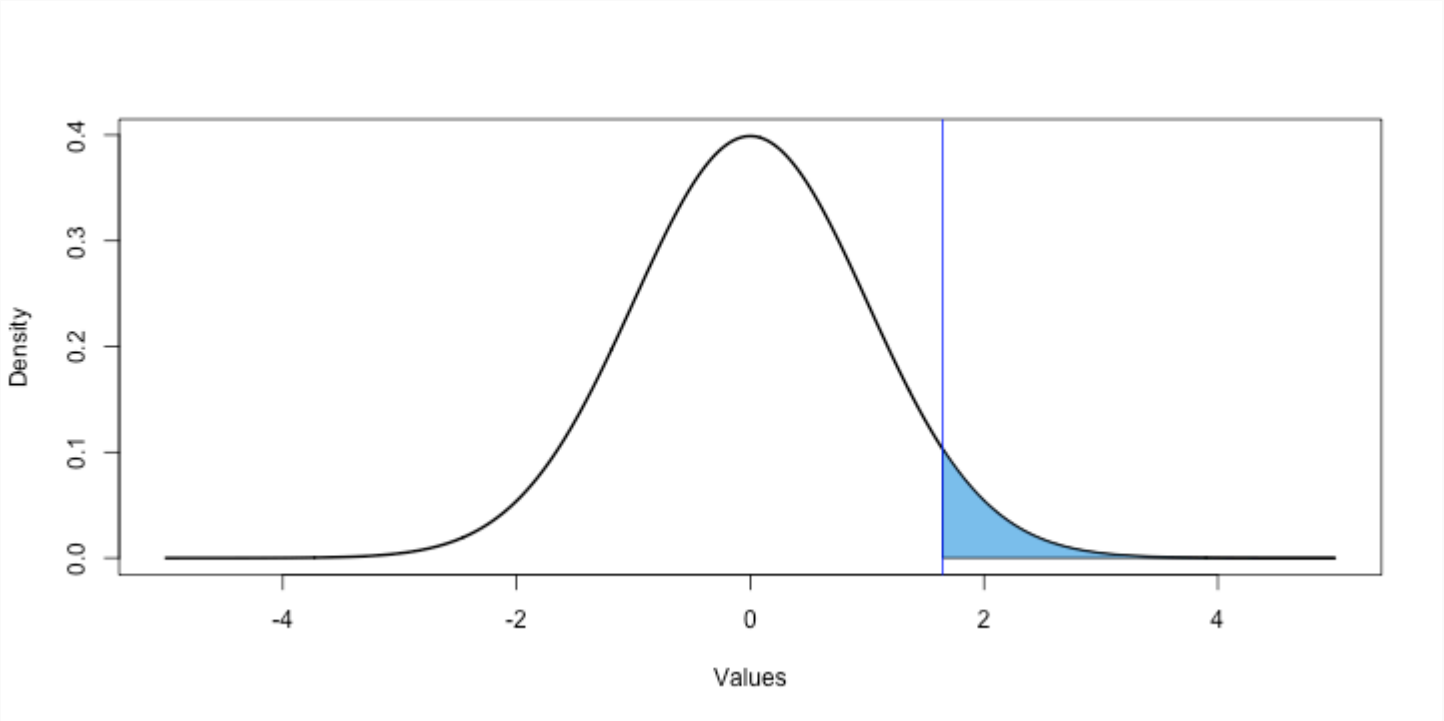
Type 1 error

Which error do you think is the worse error to make?

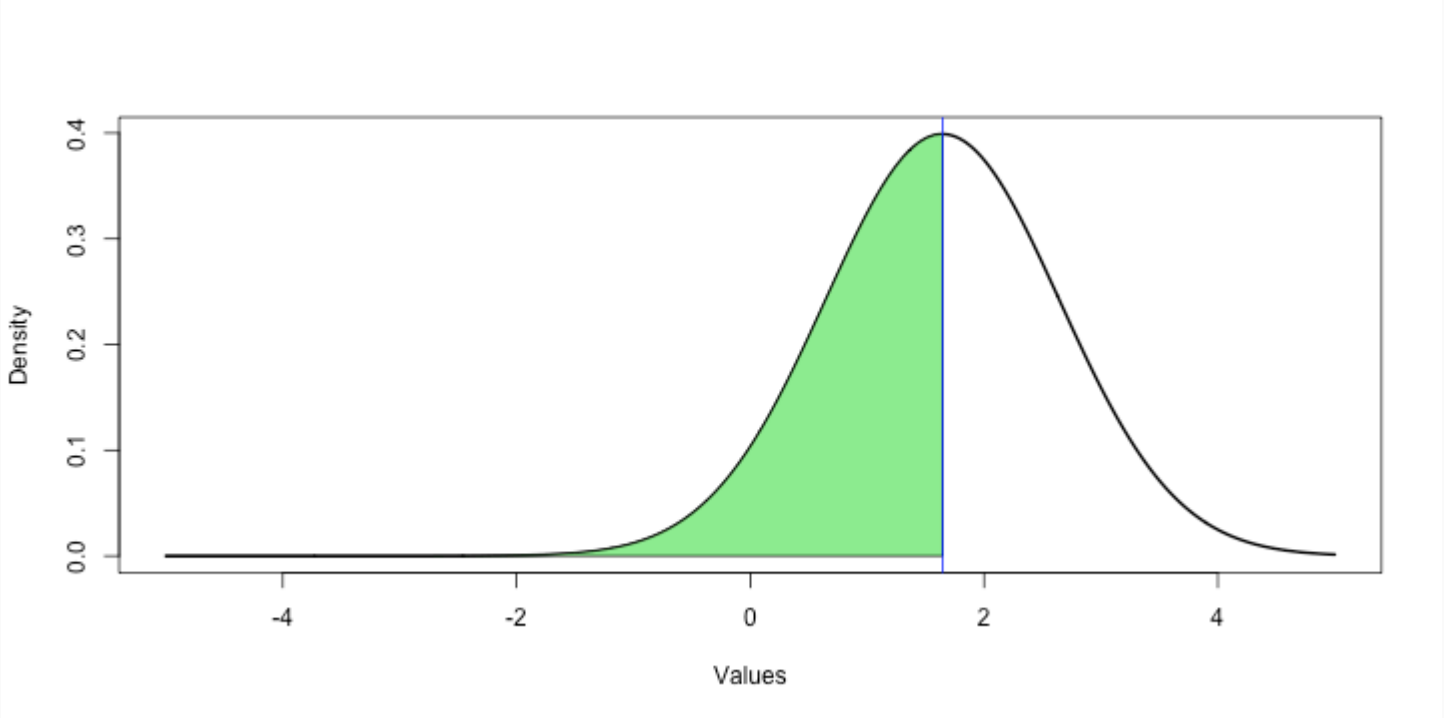
Null Distribution

```r
(cv <- qnorm(0.05, mean=0, sd=1, lower.tail=FALSE))
```

```
## [1] 1.644854
```



Alternative Distribution



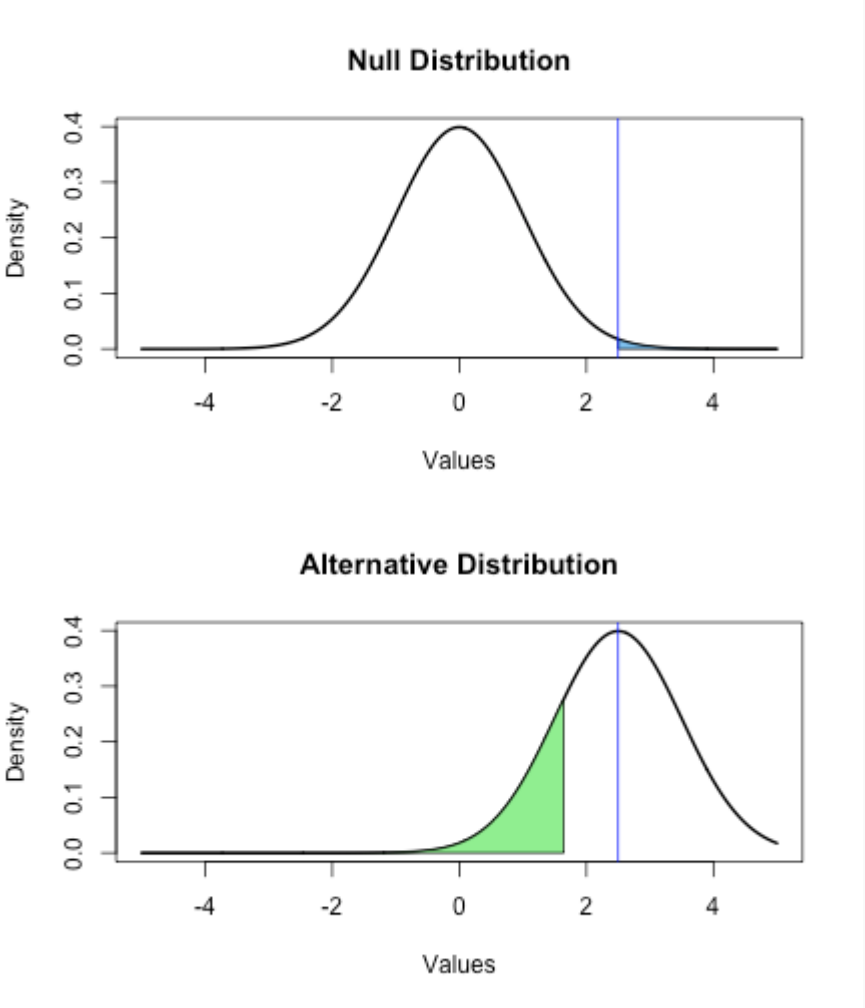

```
pnorm(cv, mean=cv, lower.tail = FALSE)
```

```
## [1] 0.5
```

Another Example (mu = 2.5)

```r
mu <- 2.5
(cv <- qnorm(0.05,
             mean=0,
             sd=1,
             lower.tail=FALSE))
```

```
## [1] 1.644854
```



Numeric Values

Type I Error

```
pnorm(mu, mean=0, sd=1, lower.tail=FALSE)
```

```
## [1] 0.006209665
```

Type II Error

```
pnorm(cv, mean=mu, lower.tail = TRUE)
```

```
## [1] 0.1962351
```

Shiny Application

Visualizing Type I and Type II errors: https://bcdudek.net/betaprob/

Why p < 0.05?

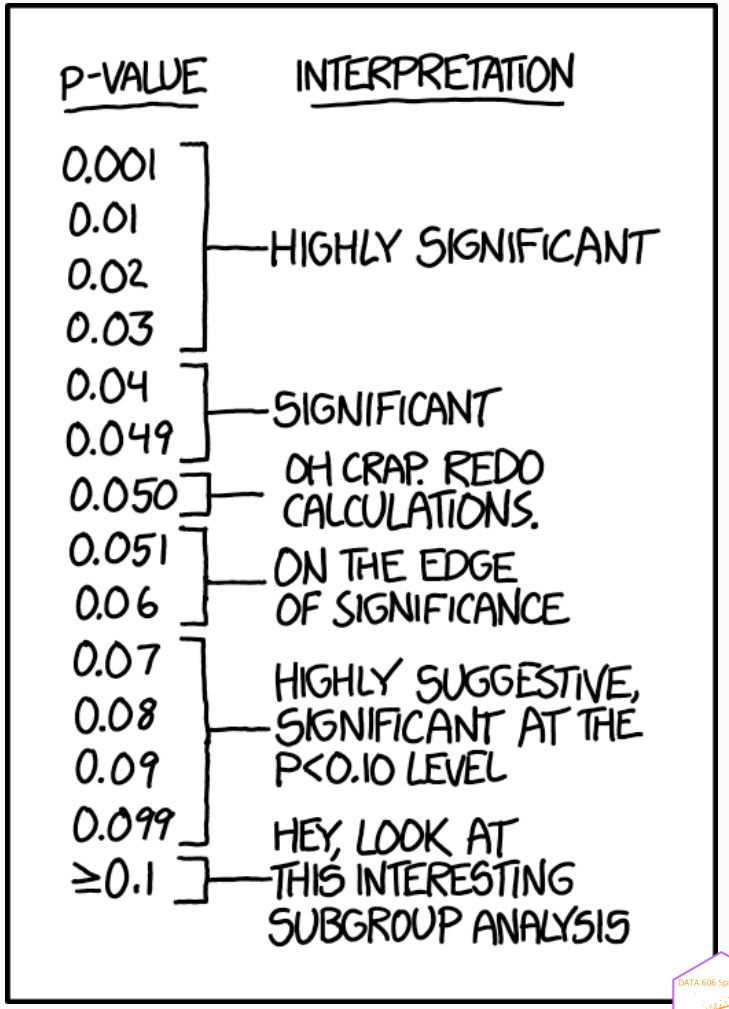
Check out this page: https://r.bryer.org/shiny/Why05/

See also:

Kelly M. *Emily Dickinson and monkeys on the stair Or: What is the significance of the 5% significance level?* Significance 10:5. 2013.

Statistical vs. Practical Significance

- Real differences between the point estimate and null value are easier to detect with larger samples.
- However, very large samples will result in statistical significance even for tiny differences between the sample mean and the null value (effect size), even when the difference is not practically significant.
- This is especially important to research: if we conduct a study, we want to focus on finding meaningful results (we want observed differences to be real, but also large enough to matter).
- The role of a statistician is not just in the analysis of data, but also in planning and design of a study.



Bootstrapping

Bootstrapping

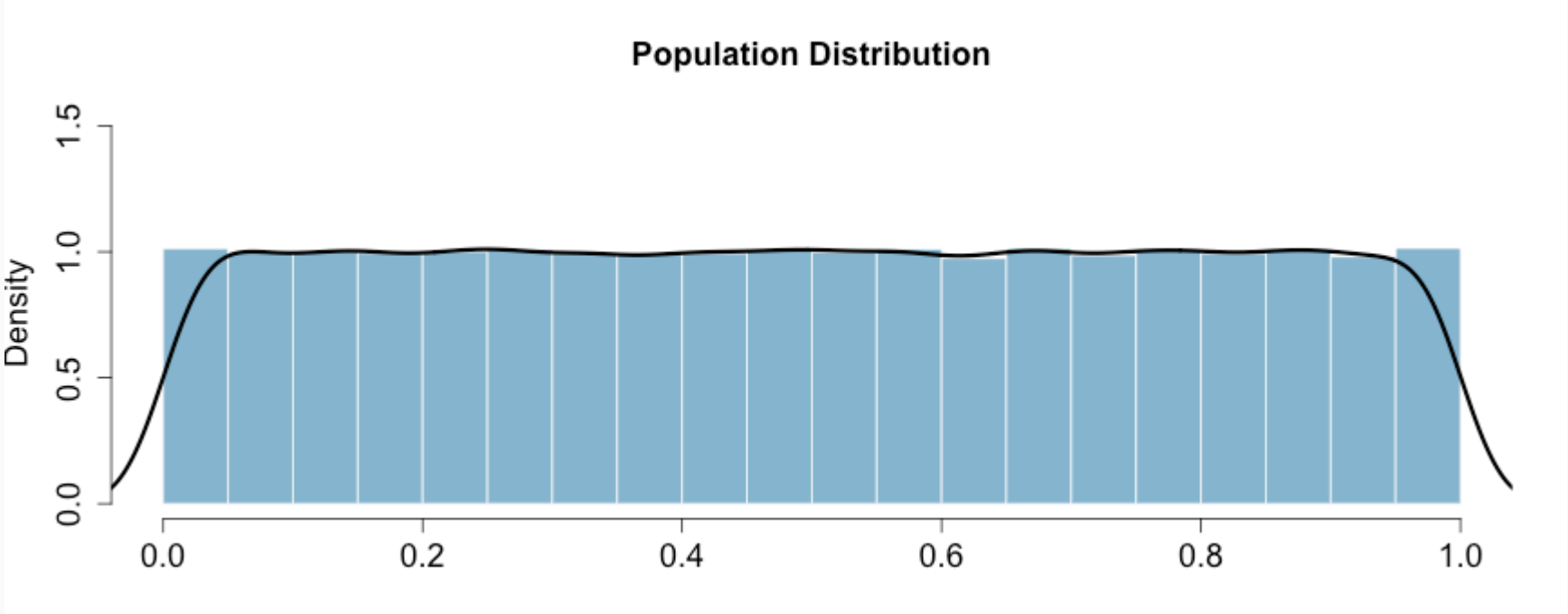
- First introduced by Efron (1979) in *Bootstrap Methods: Another Look at the Jackknife*.
- Estimates confidence of statistics by resampling *with* replacement.
- The *bootstrap sample* provides an estimate of the sampling distribution.
- The `boot` R package provides a framework for doing bootstrapping: https://www.statmethods.net/advstats/bootstrapping.html

Bootstrapping Example (Population)

Define our population with a uniform distribution.

```
n <- 1e5
pop <- runif(n, 0, 1)
mean(pop)
```

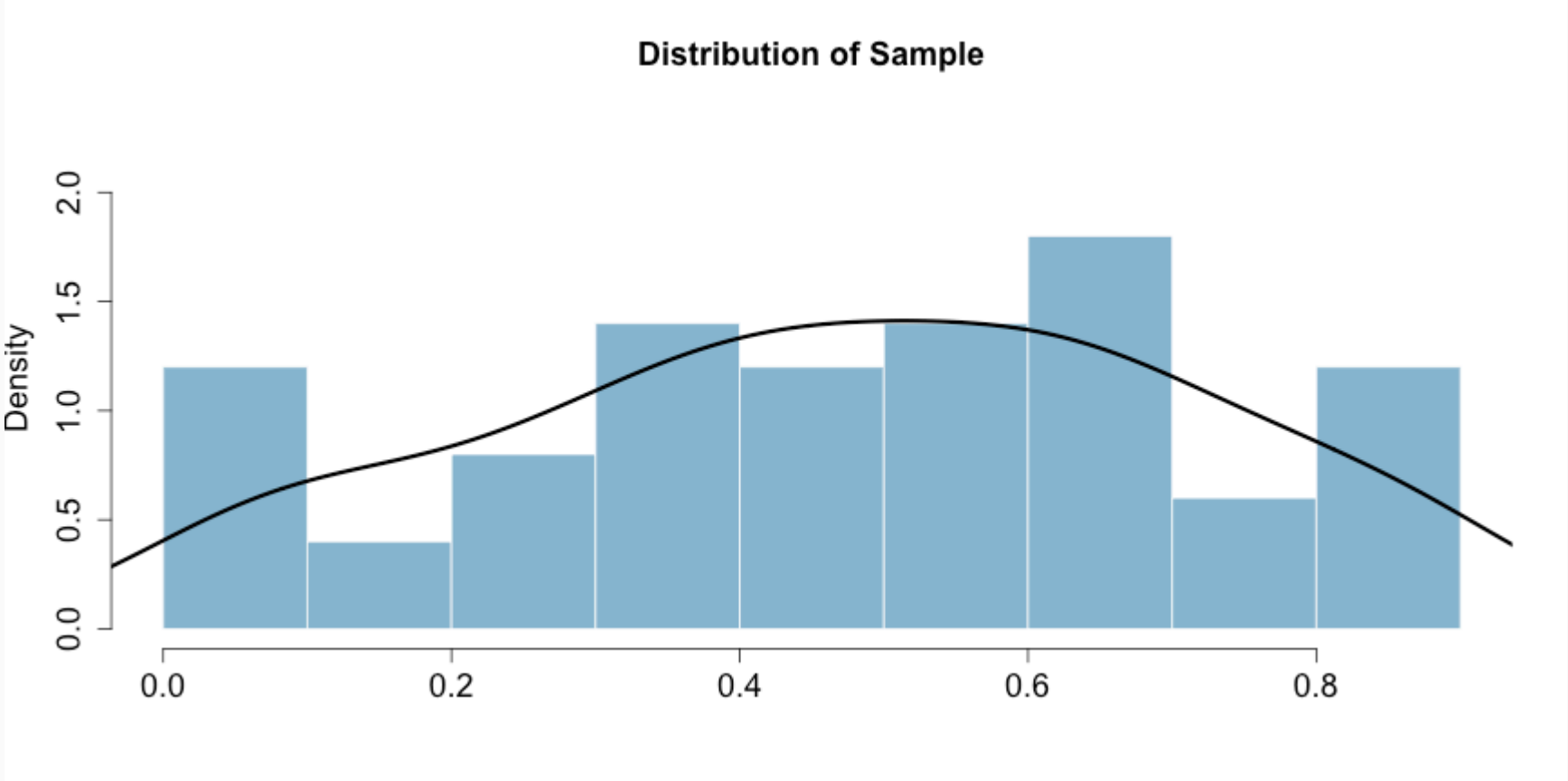
```
## [1] 0.4998327
```



Bootstrapping Example (Sample)

We observe one random sample from the population.

```
samp1 <- sample(pop, size = 50)
```



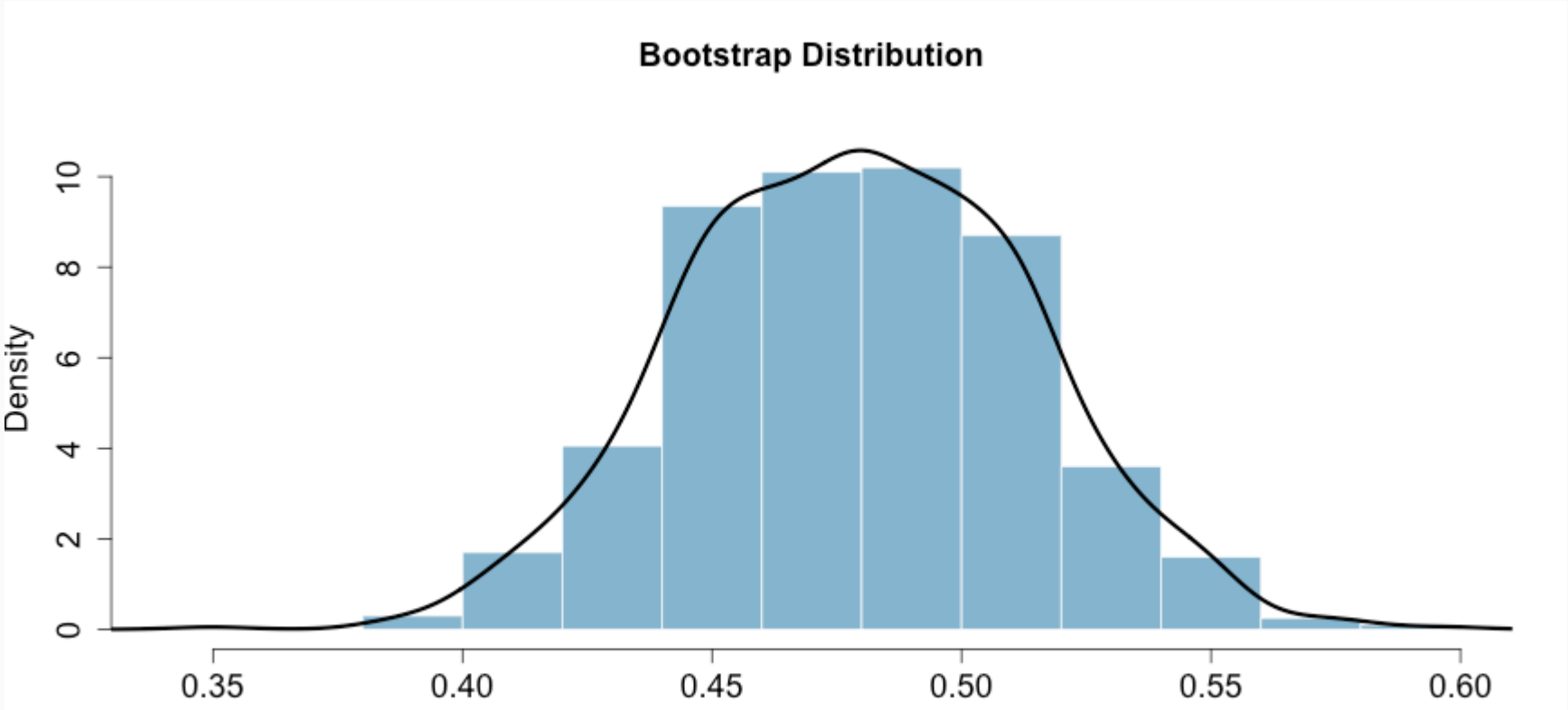
Bootsrapping Example (Estimate)

```r
boot.samples <- numeric(1000) # 1,000 bootstrap samples
for(i in seq_along(boot.samples)) {
    tmp <- sample(samp1, size = length(samp1), replace = TRUE)
    boot.samples[i] <- mean(tmp)
}
head(boot.samples)
```

```
## [1] 0.4603470 0.4375637 0.4953504 0.4700643 0.4509329 0.5534673
```

Bootsrapping Example (Distribution)

```
d <- density(boot.samples)
h <- hist(boot.samples, plot=FALSE)
hist(boot.samples, main='Bootstrap Distribution', xlab="", freq=FALSE,
     ylim=c(0, max(d$y, h$density)+.5), col=COL[1,2], border = "white",
     cex.main = 1.5, cex.axis = 1.5, cex.lab = 1.5)
lines(d, lwd=3)
```



95% confidence interval

```
c(mean(boot.samples) - 1.96 * sd(boot.samples),
  mean(boot.samples) + 1.96 * sd(boot.samples))
```

```
## [1] 0.4115592 0.5464879
```

Bootstrapping is not just for means!

```
boot.samples.median <- numeric(1000) # 1,000 bootstrap samples
for(i in seq_along(boot.samples.median)) {
    tmp <- sample(samp1, size = length(samp1), replace = TRUE)
    boot.samples.median[i] <- median(tmp) # NOTICE WE ARE NOW USING THE median FUNCTION!
}
head(boot.samples.median)
```

```
## [1] 0.3679071 0.6229986 0.4834771 0.4943032 0.4943032 0.5058774
```

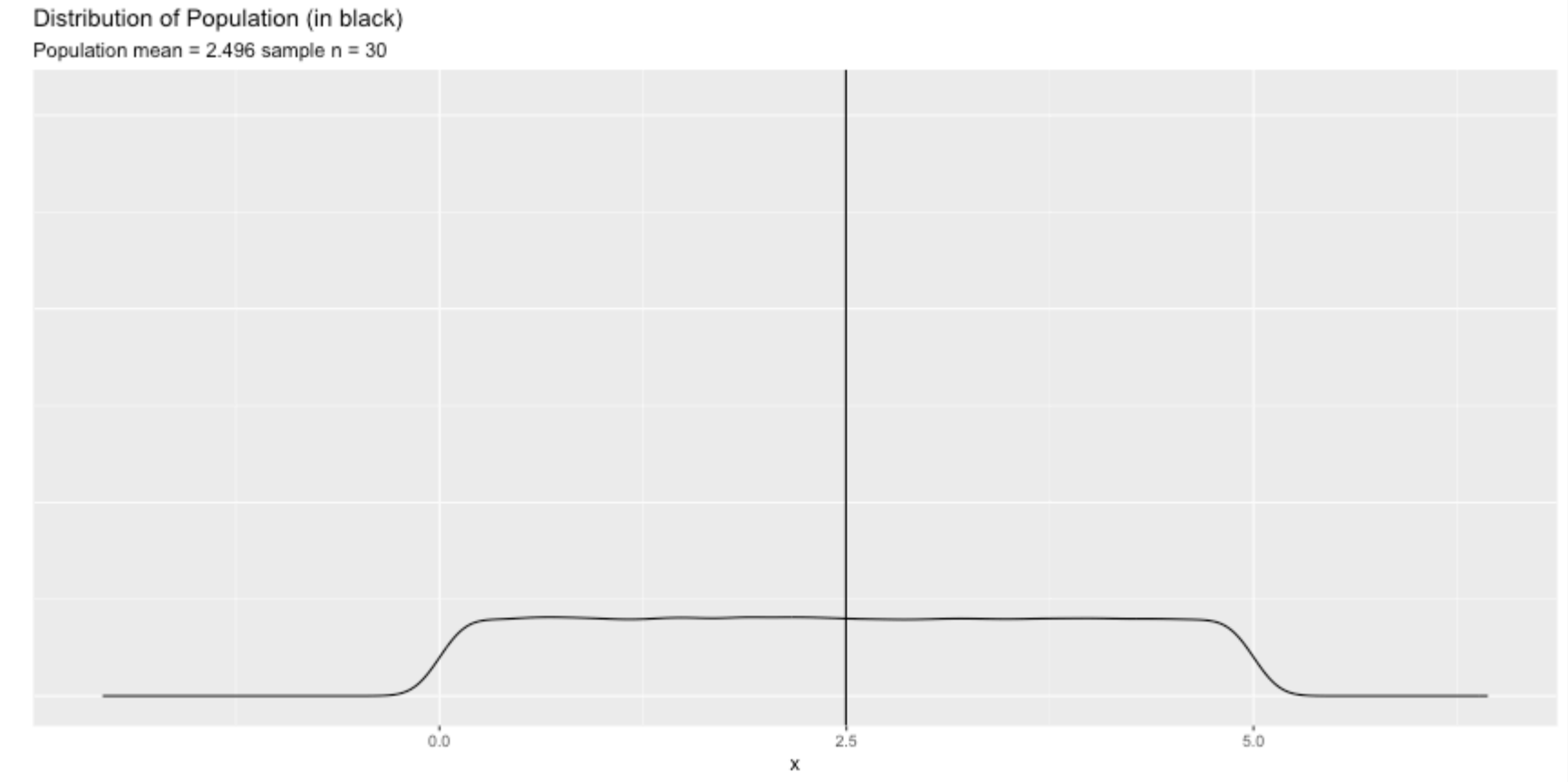
95% confidence interval for the median

```
c(mean(boot.samples.median) - 1.96 * sd(boot.samples.median),
  mean(boot.samples.median) + 1.96 * sd(boot.samples.median))
```

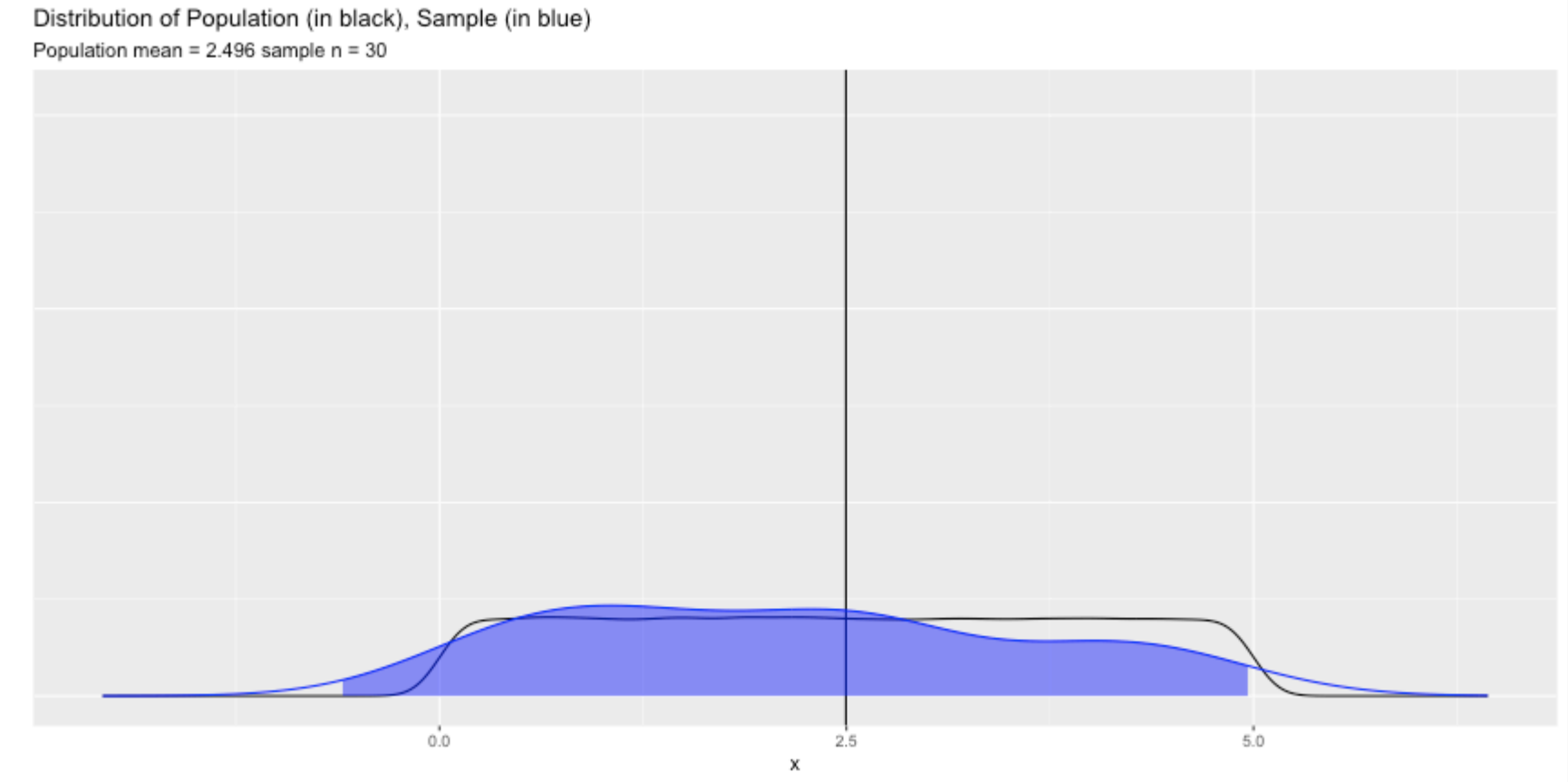
```
## [1] 0.3923681 0.5829224
```

Review

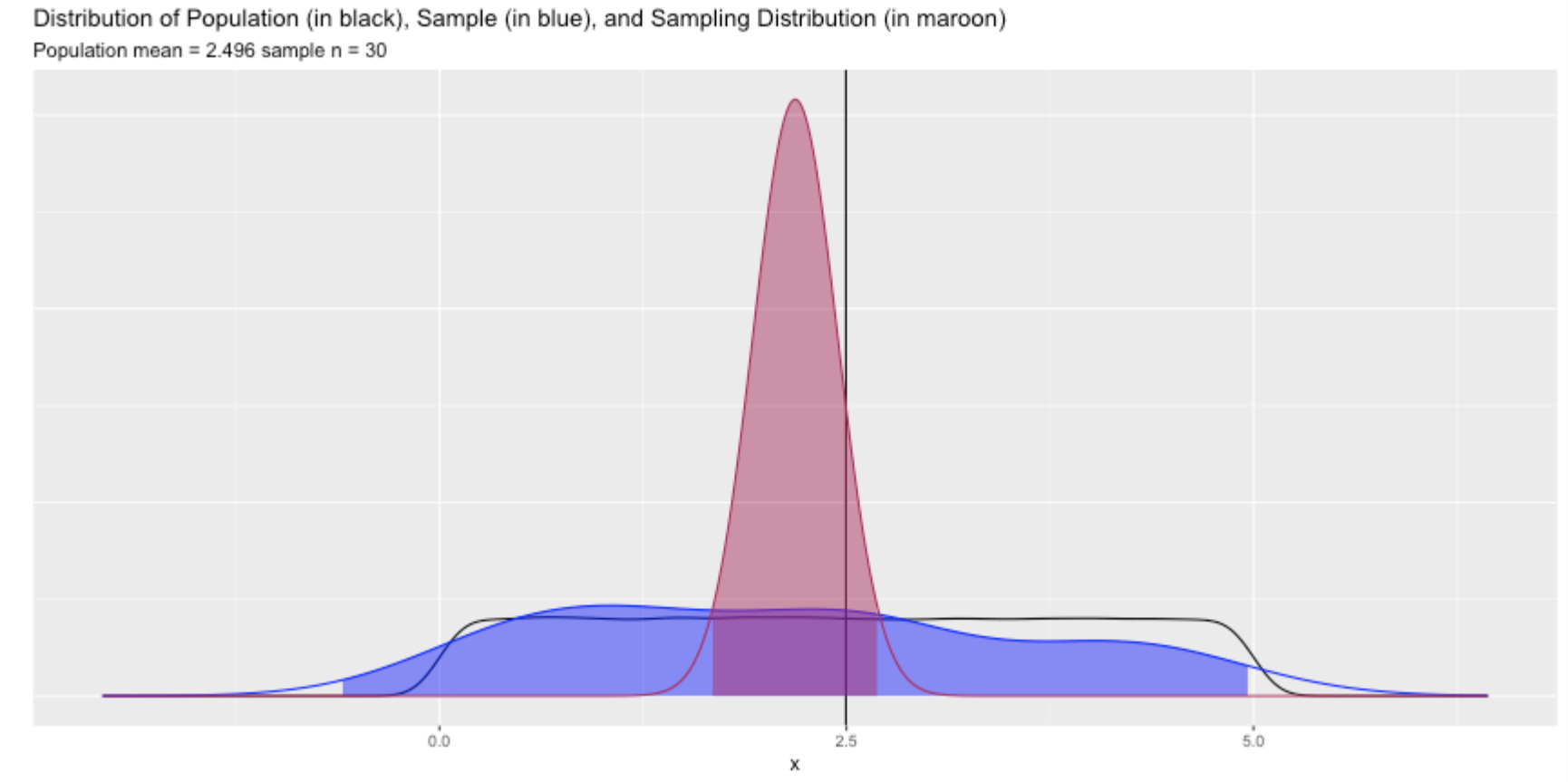
Review: Sampling Distribution



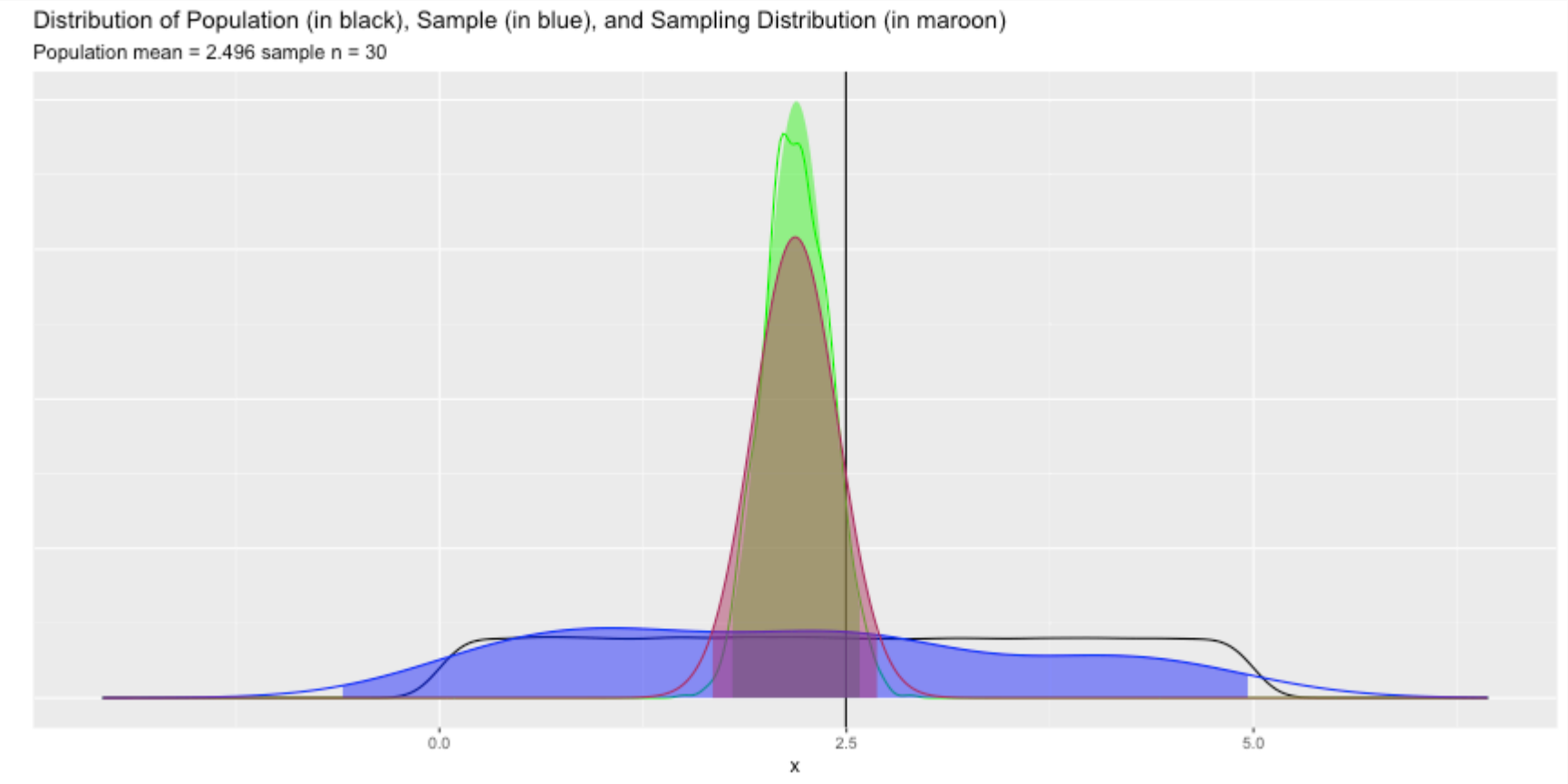
Review: Sampling Distribution



Review: Sampling Distribution



Review: Add Bootstrap Distribution



One Minute Paper

1. What was the most important thing you learned during this class?
2. What important question remains unanswered for you?

https://forms.gle/Ze19MooQHvZmQE2ZA