

# Foundation for Inference Part 1

DATA 606 - Statistics & Probability for Data Analytics

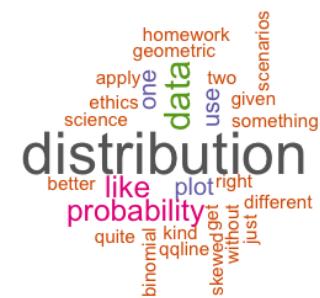
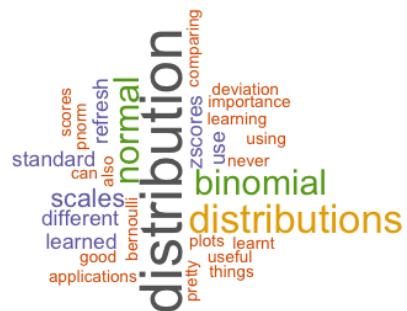
Jason Bryer, Ph.D. and Angela Lui, Ph.D.

September 29, 2021

# One Minute Paper Results

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

# **What important question remains unanswered for you?**



# Homework Presentations

- 4.17 Nick Oliver
- 4.19 William Aiken
- 4.25 Tyler Baker



# Crash Course in Calculus

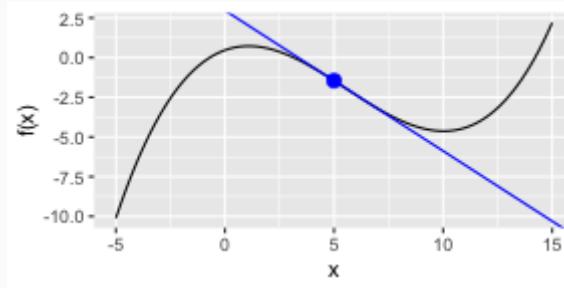


# Crash Course in Calculus

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

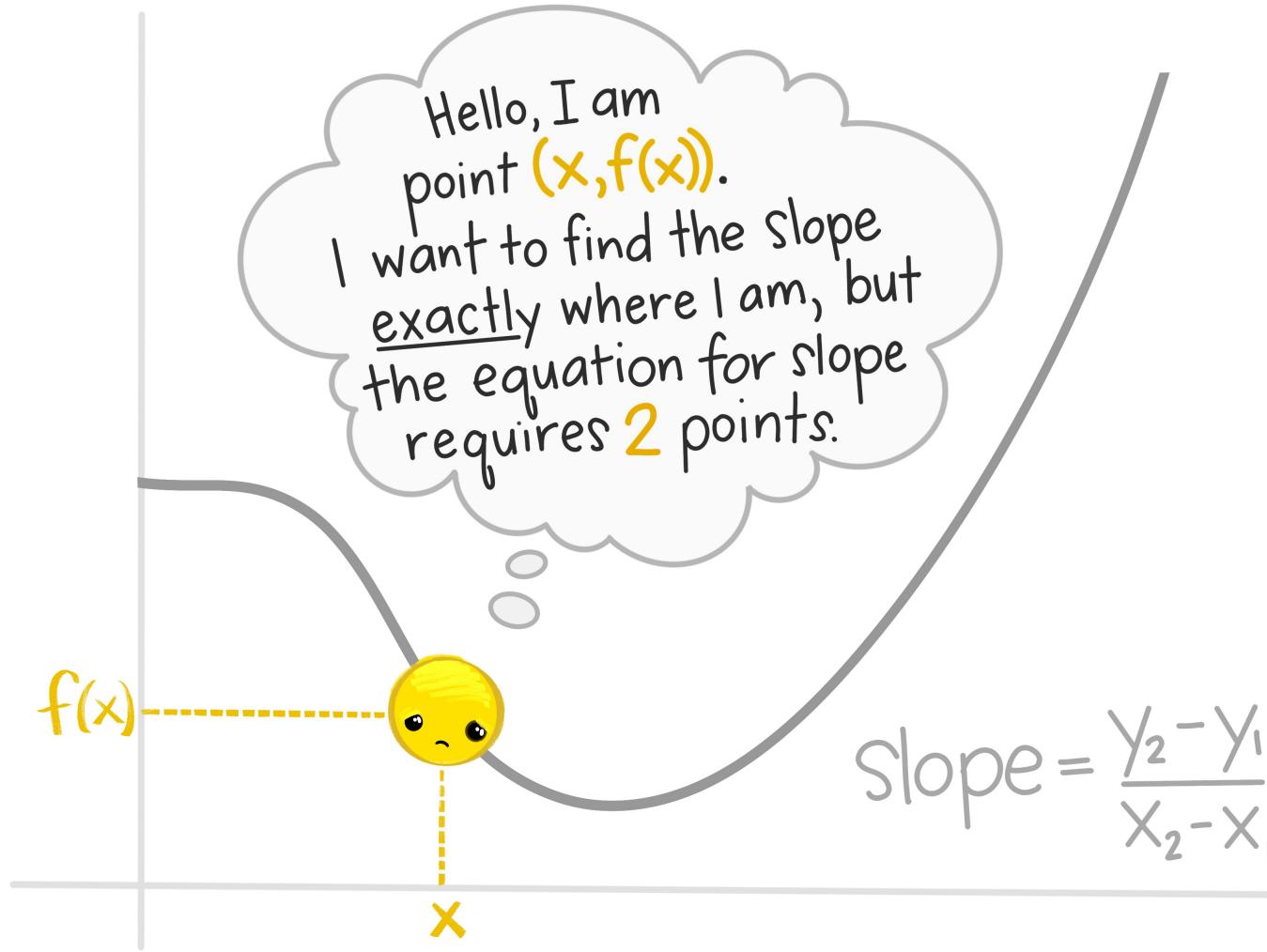
**Limits** - the value that a function (or sequence) approaches as the input (or index) approaches some value.

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

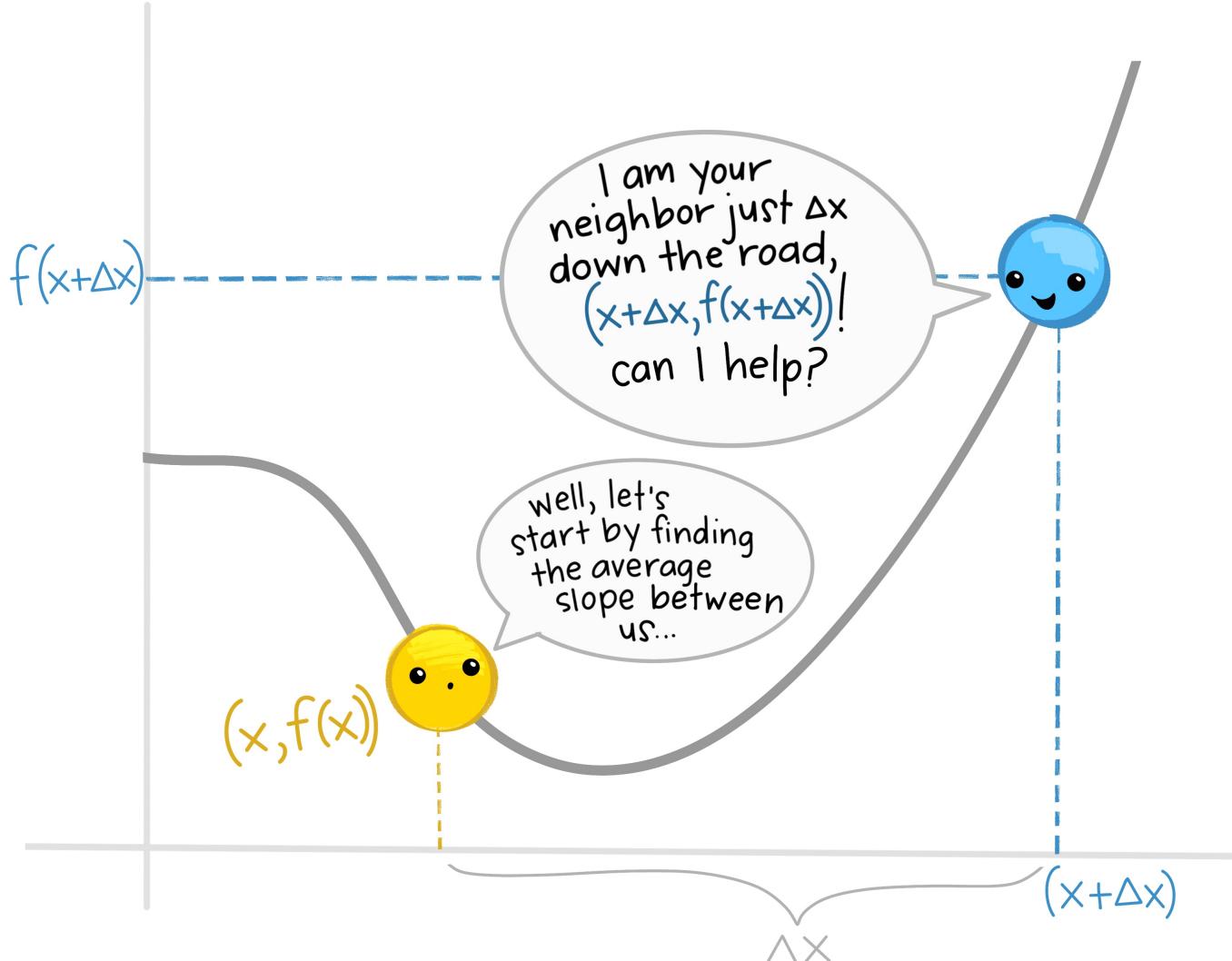


**Integrals** - the area under the curve.

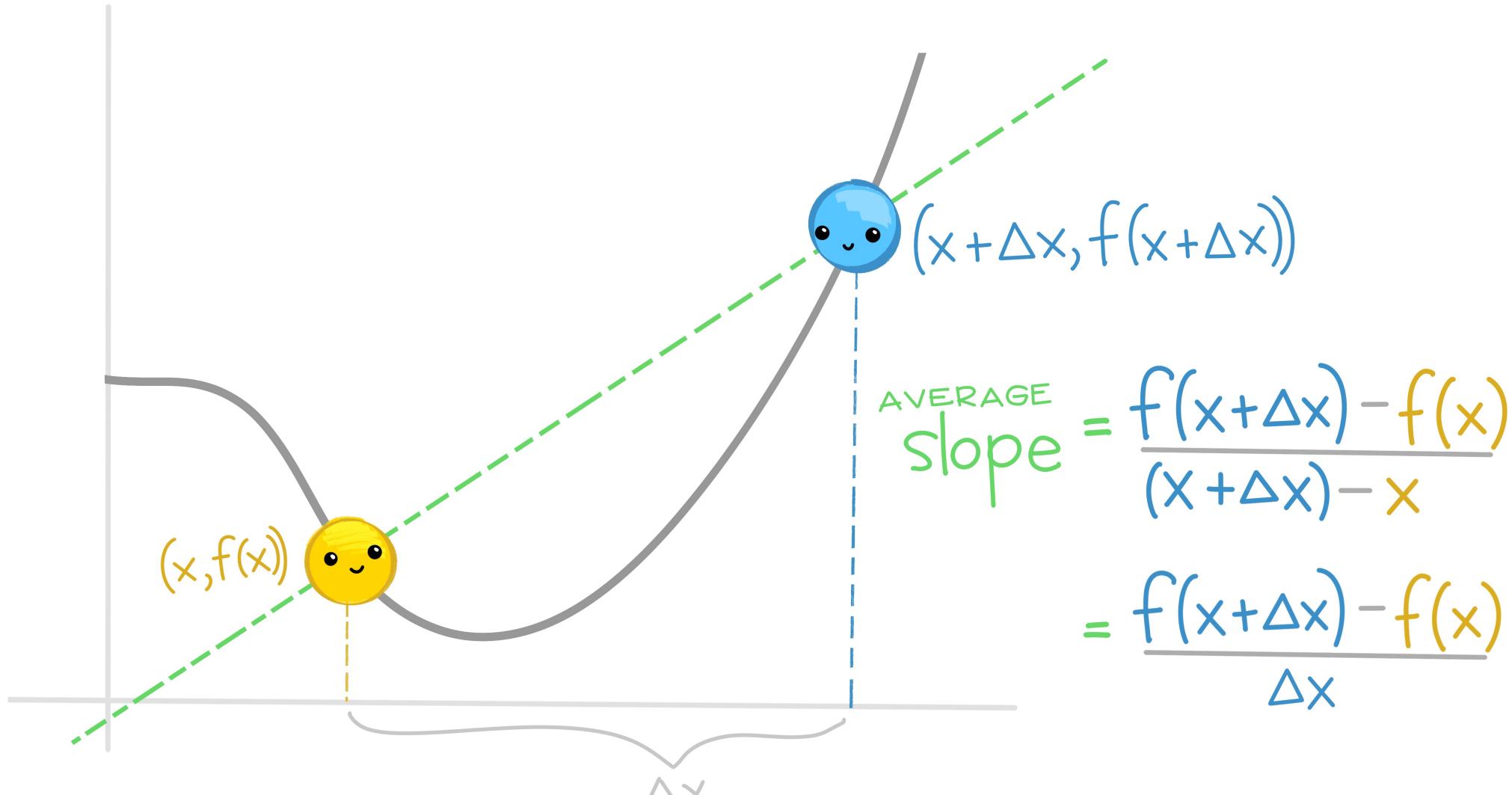
# Derivatives



# Derivatives



# Derivatives



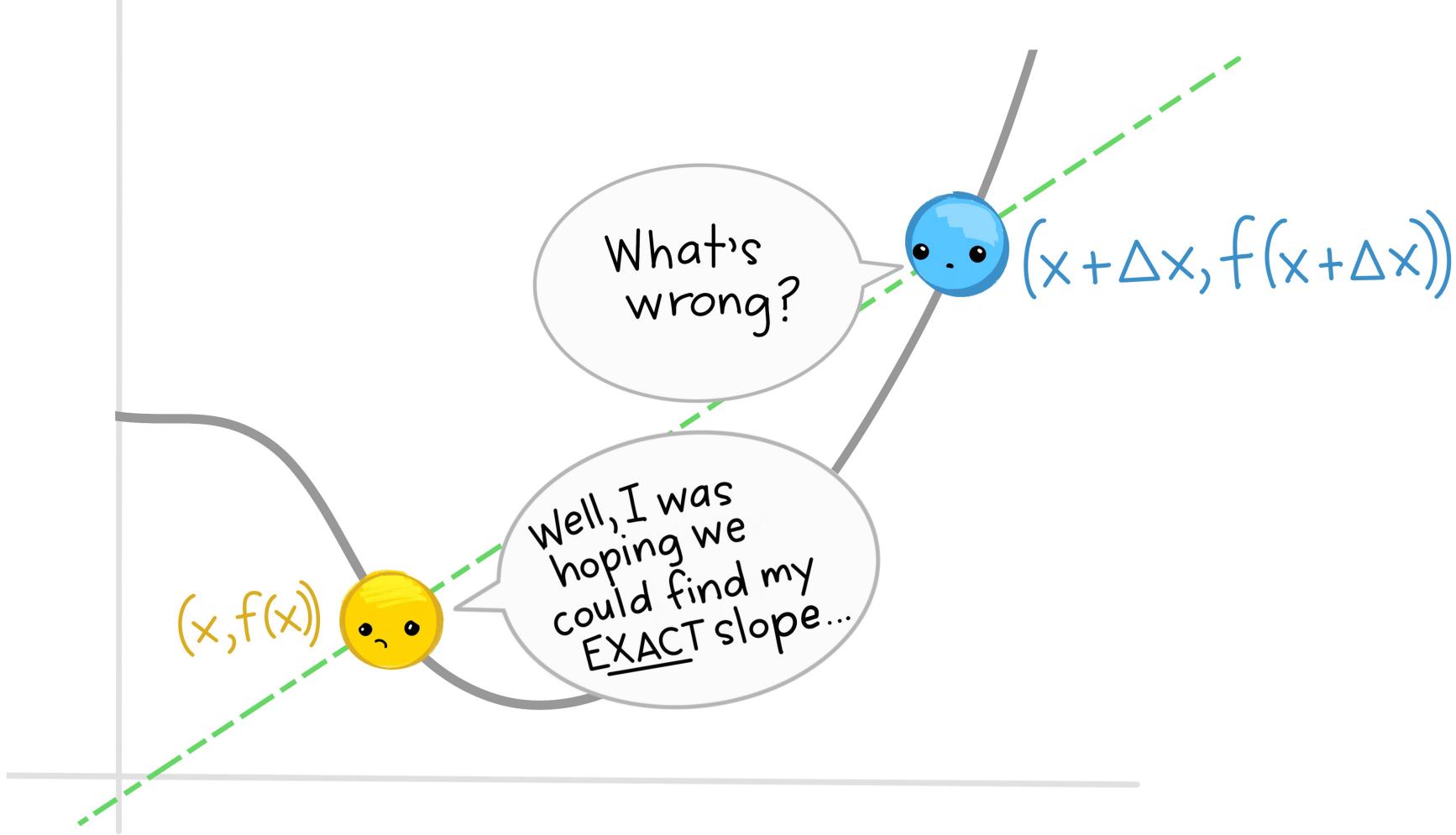
# Derivatives

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

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

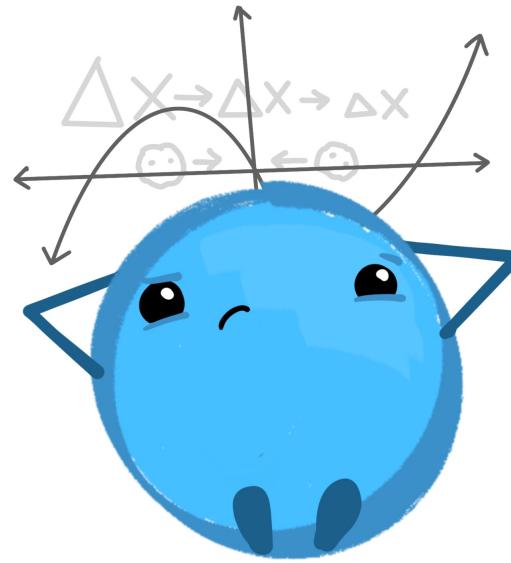


# Derivatives

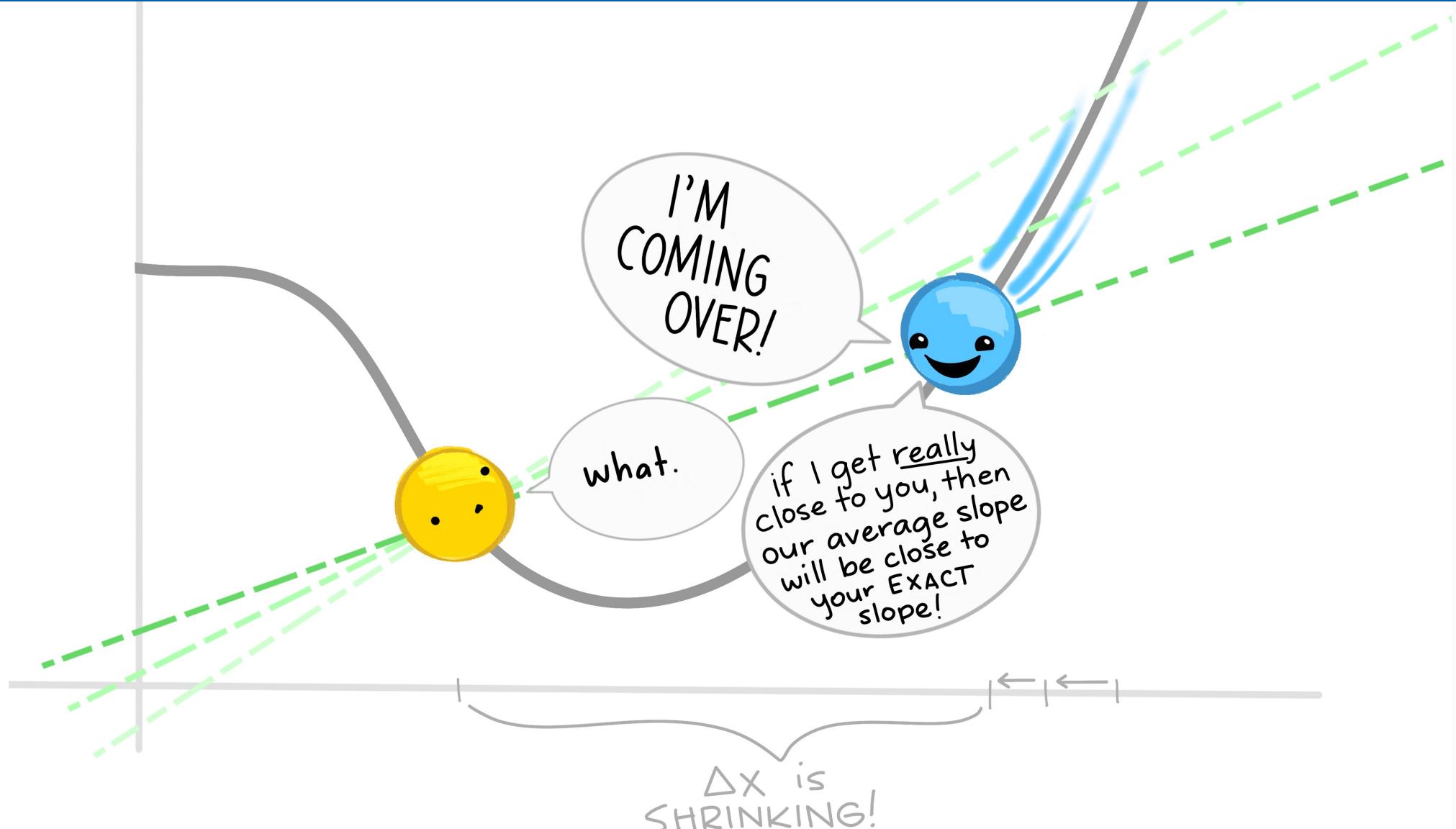


# Derivatives

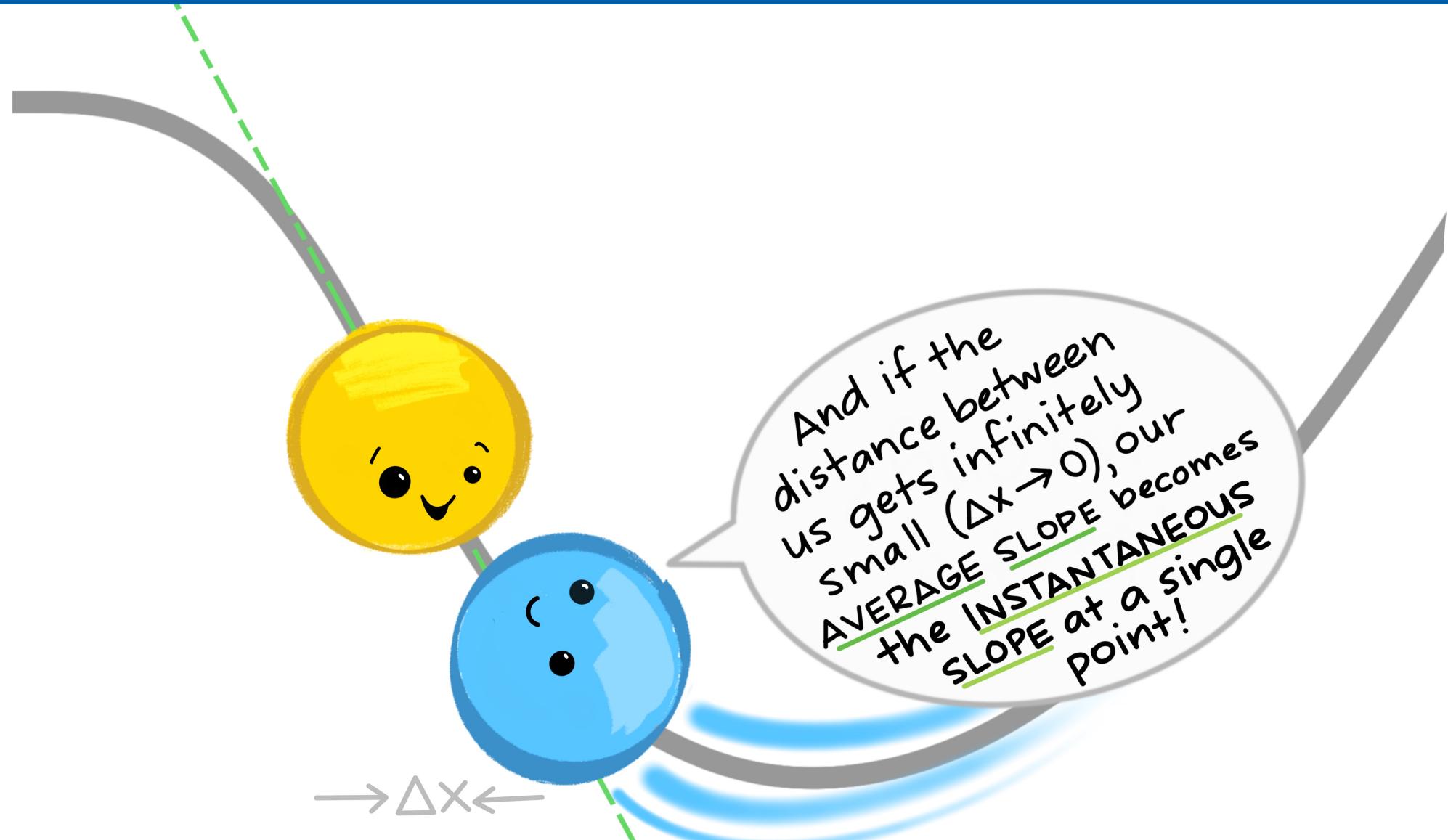
## BRAINSTORM MONTAGE!



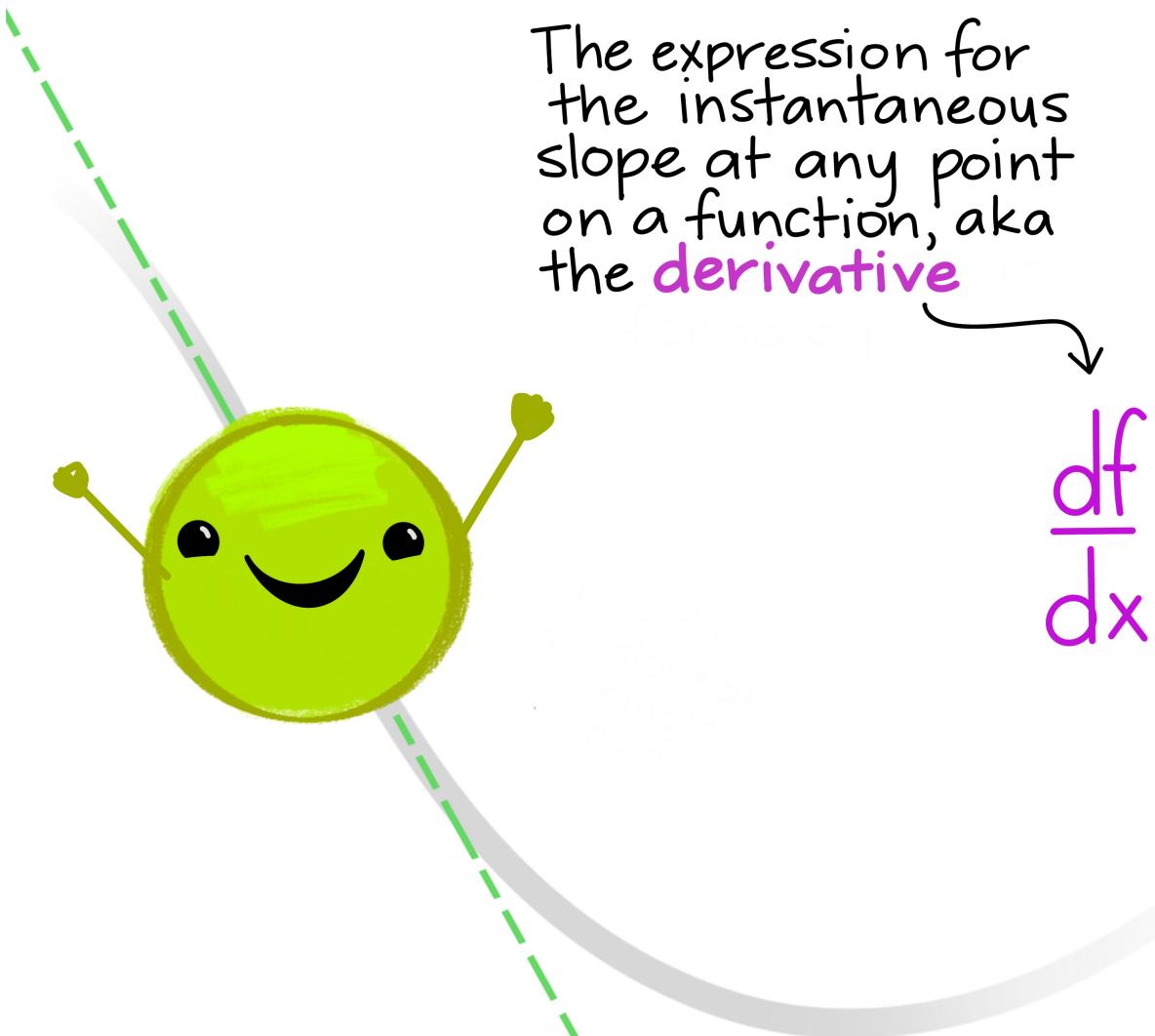
# Derivatives



# Derivatives



# Derivatives



The expression for  
the instantaneous  
slope at any point  
on a function, aka  
the **derivative**

$$\frac{df}{dx} = \lim_{\Delta x \rightarrow 0} \frac{f(x+\Delta x) - f(x)}{\Delta x}$$

IS FOUND BY:

- ① Finding an expression for the **slope** between 2 points separated by  $\Delta x$ ...

$$\underbrace{\frac{f(x+\Delta x) - f(x)}{\Delta x}}$$

- ② Evaluating that slope as the points get infinitely close together.

# Function for Normal Distribution

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

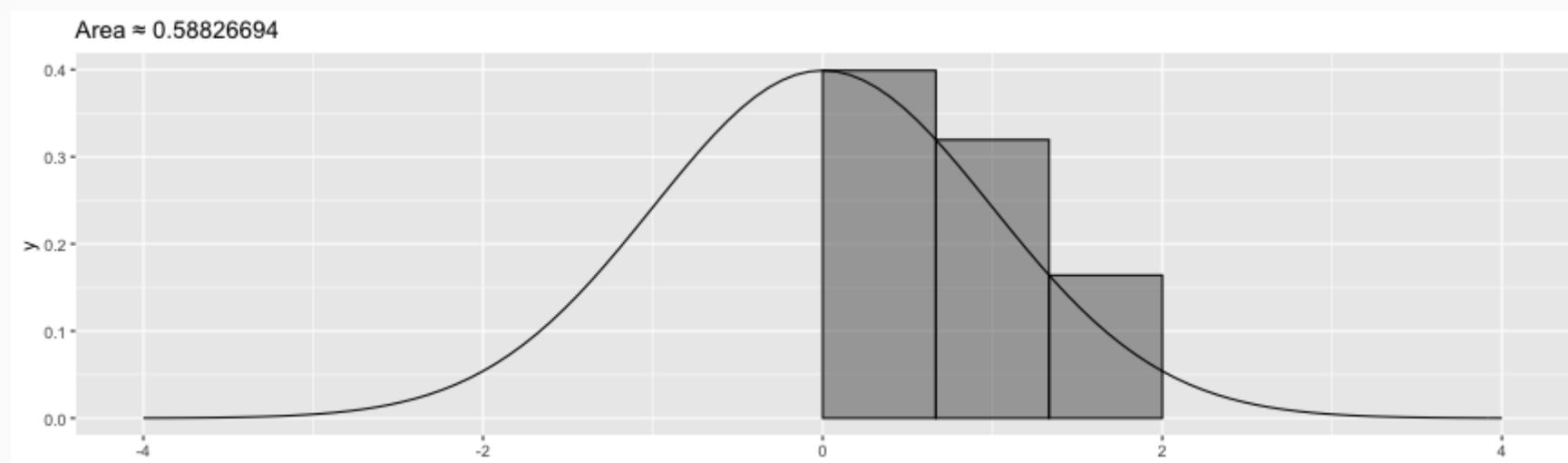
```
f <- function(x, mean = 0, sigma = 1) {  
  1 / (sigma * sqrt(2 * pi)) * exp(1)^(-1/2 * ( (x - mean) / sigma )^2)  
}
```

```
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')
```

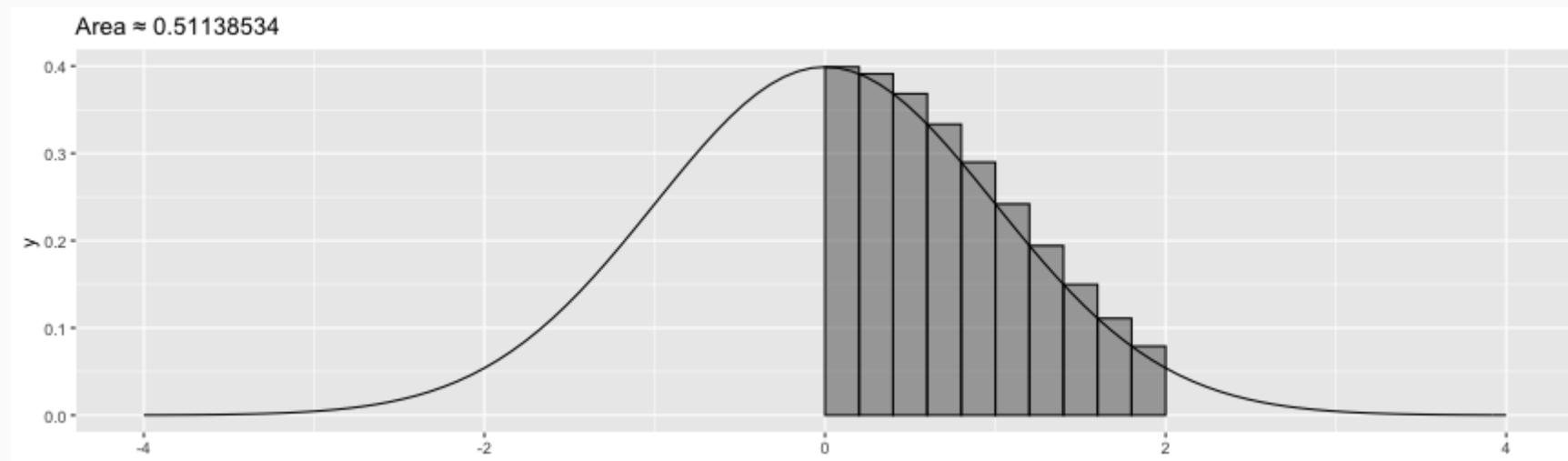


# Reimann Sums

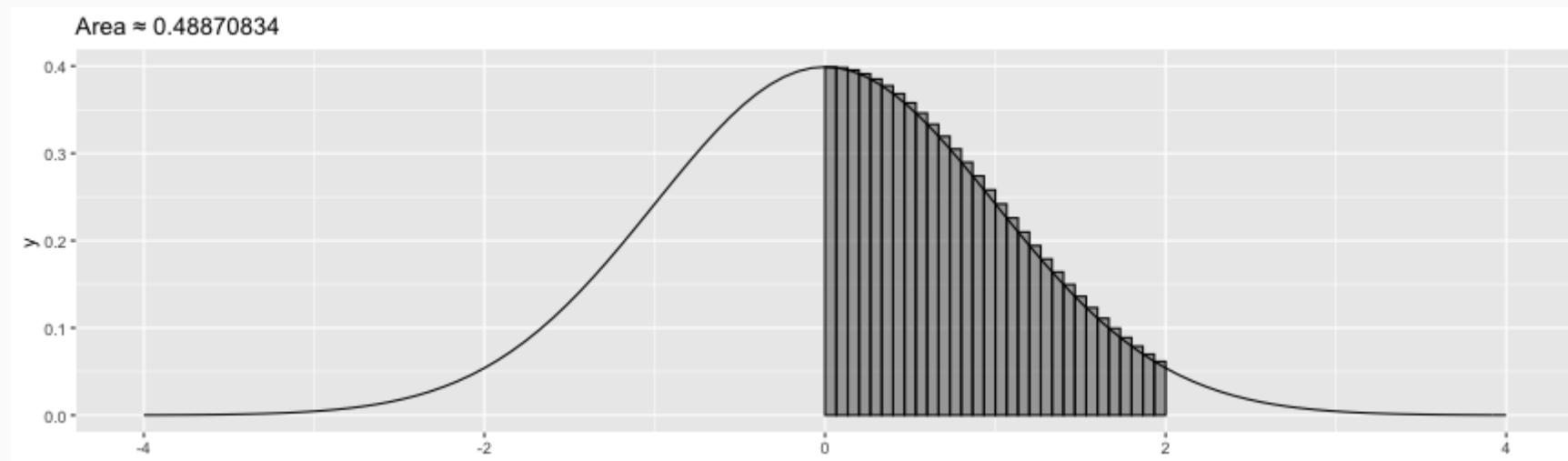
One strategy to find the area between two values is to draw a series of rectangles. Given  $n$  rectangles, we know that the width of each is  $\frac{2-0}{n}$  and the height is  $f(x)$ . Here is an example with 3 rectangles.



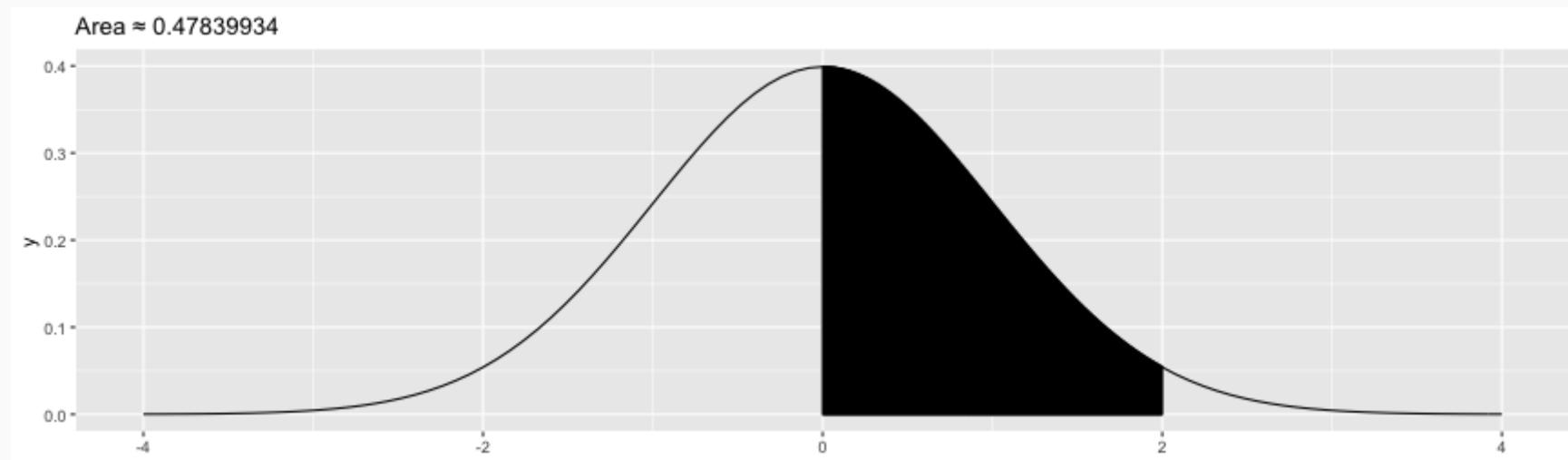
# Reimann Sums (10 rectangles)



# Reimann Sums (30 rectangles)



# Reimann Sums (300 rectangles)



$n \rightarrow \infty$

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)
```

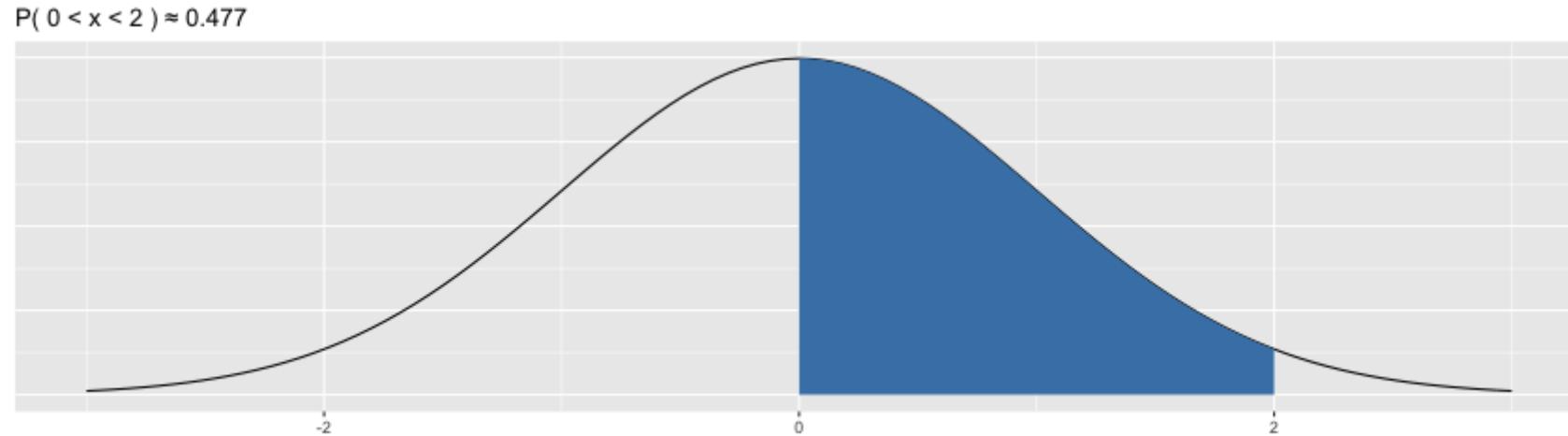
```
## 0.4772499 with absolute error < 5.3e-15
```

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



# Normal Distribution

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

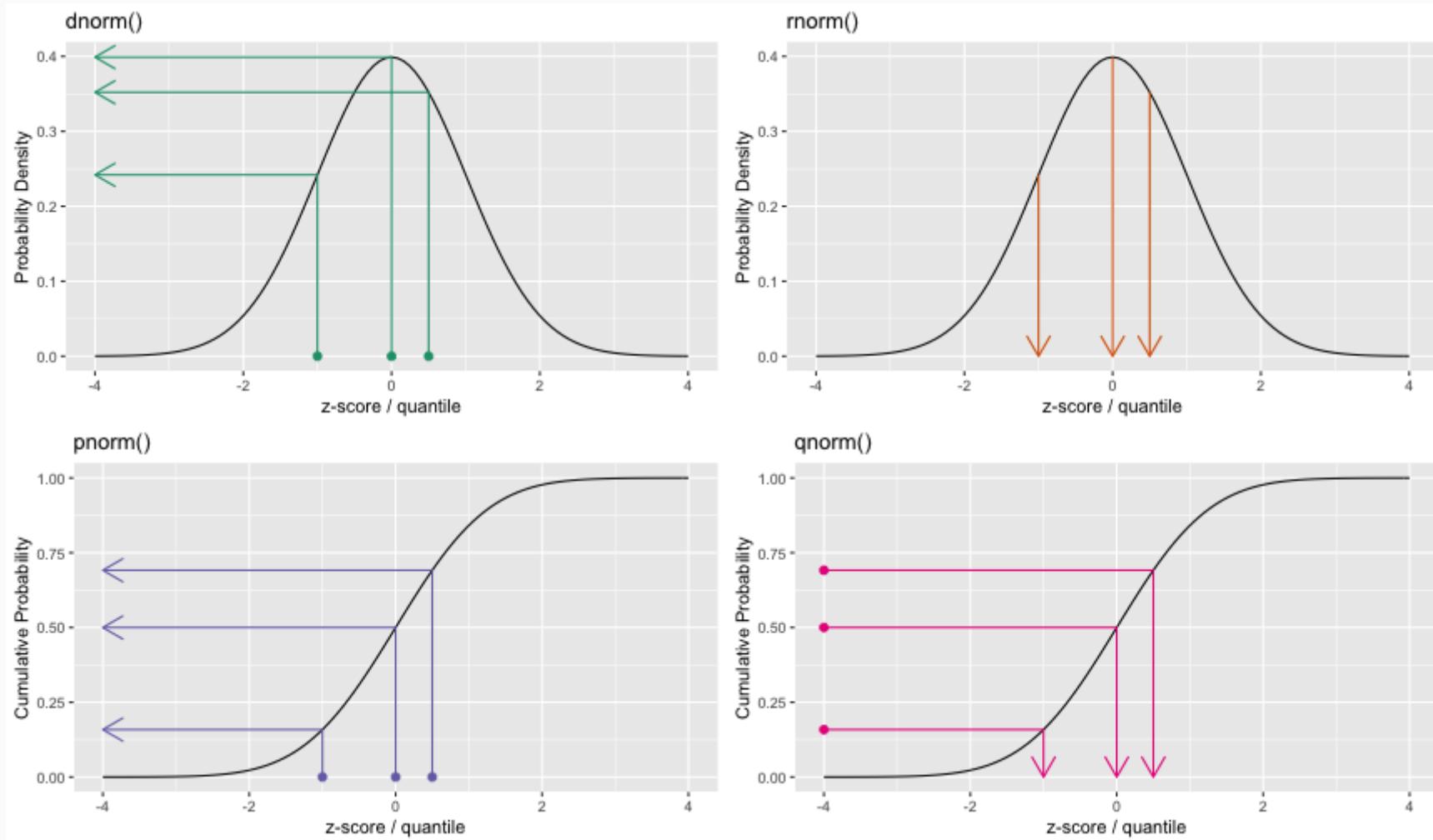


```
pnorm(2) - pnorm(0)
```

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



# R's built in functions for working with distributions



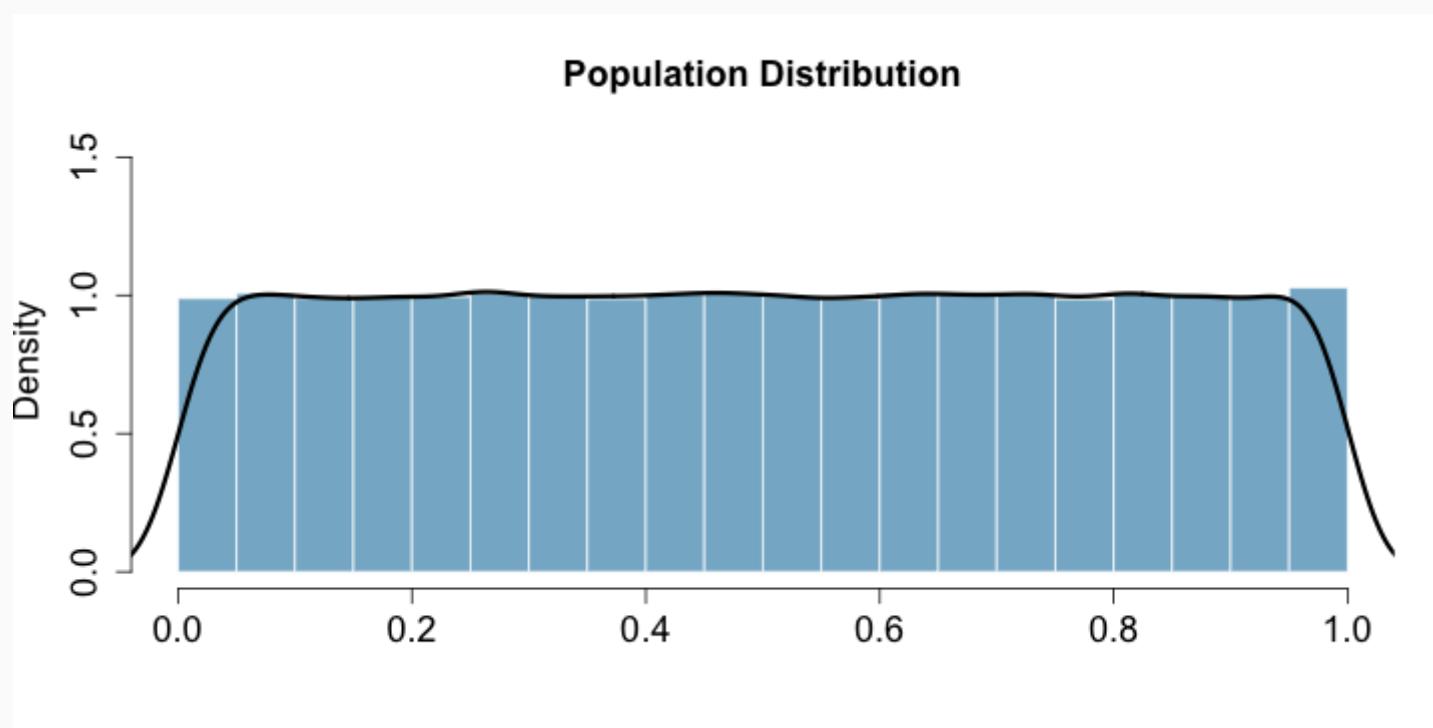
See <https://github.com/jbryer/DATA606Fall2021/blob/master/R/distributions.R>

# Foundation for Inference

# Population Distribution (Uniform)

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

```
## [1] 0.5009061
```

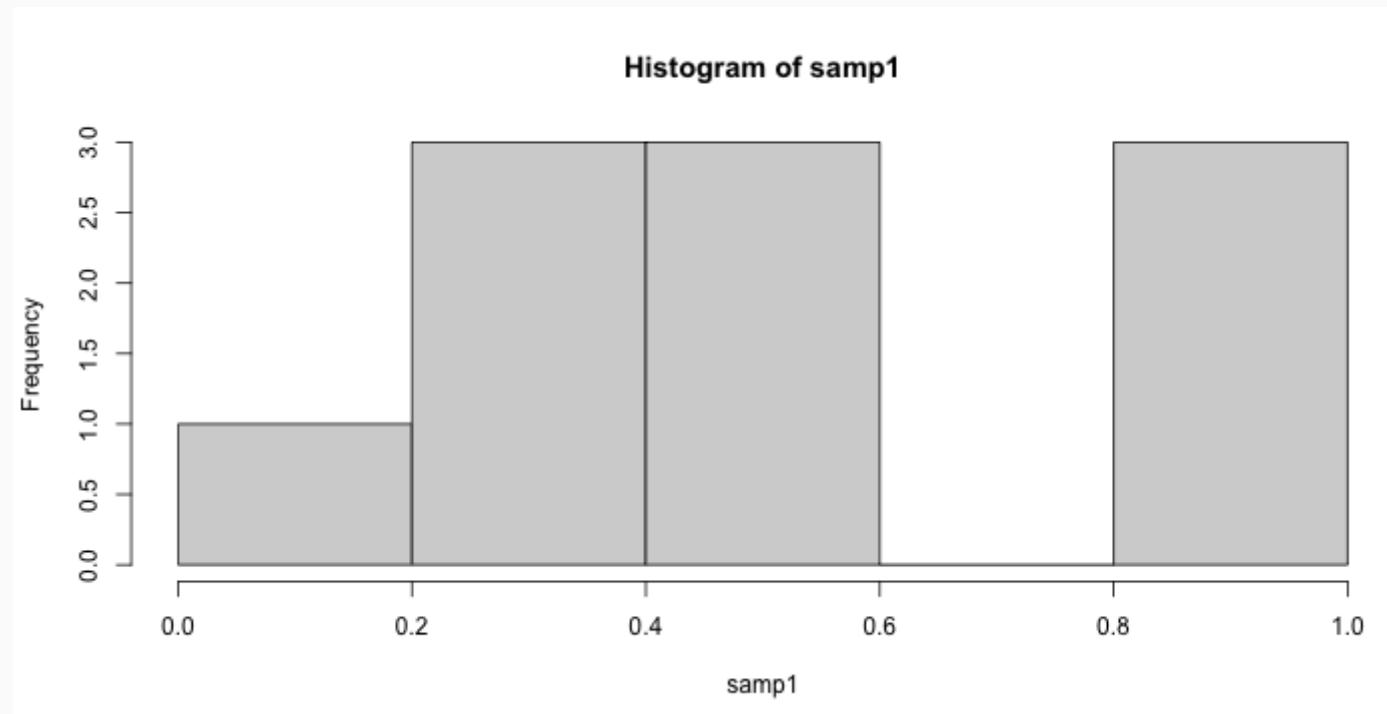


# Random Sample (n=10)

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

```
## [1] 0.5042593
```

```
hist(samp1)
```

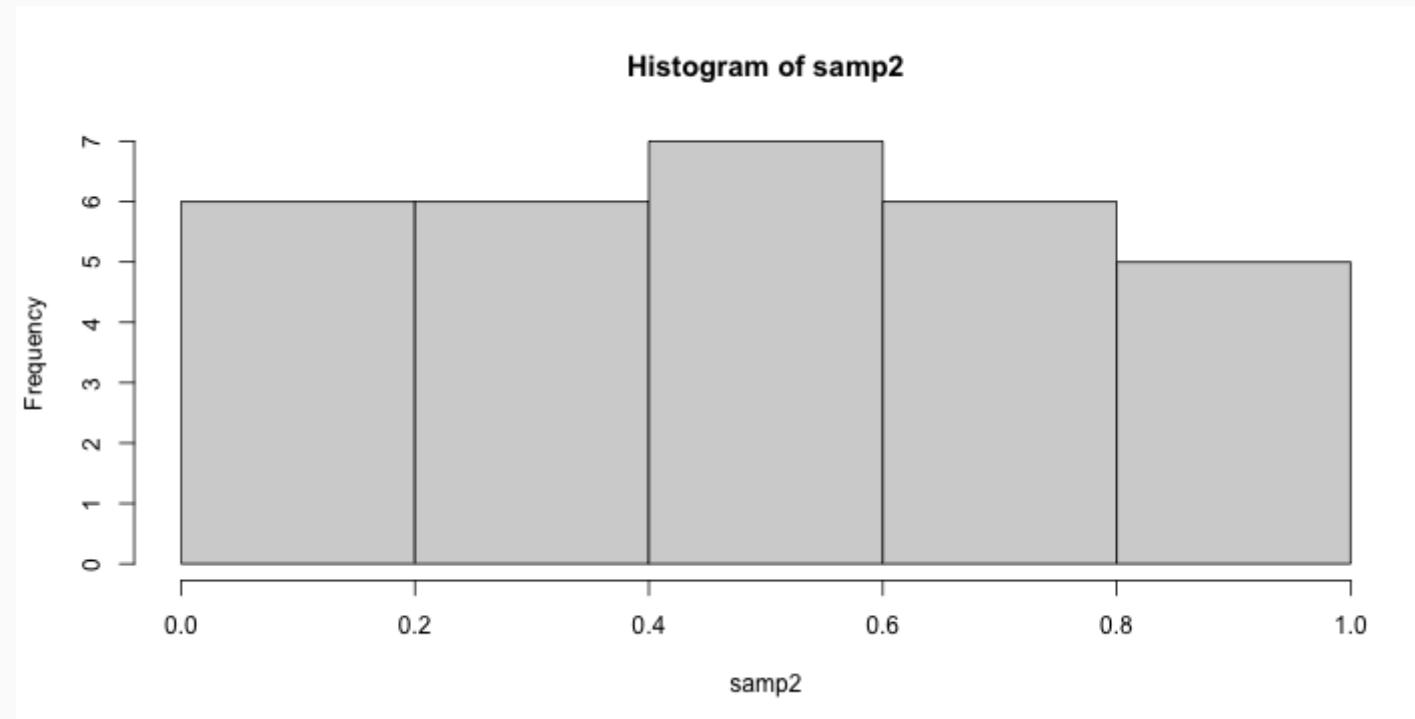


# Random Sample (n=30)

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

```
## [1] 0.4856277
```

```
hist(samp2)
```



# Lots of Random Samples

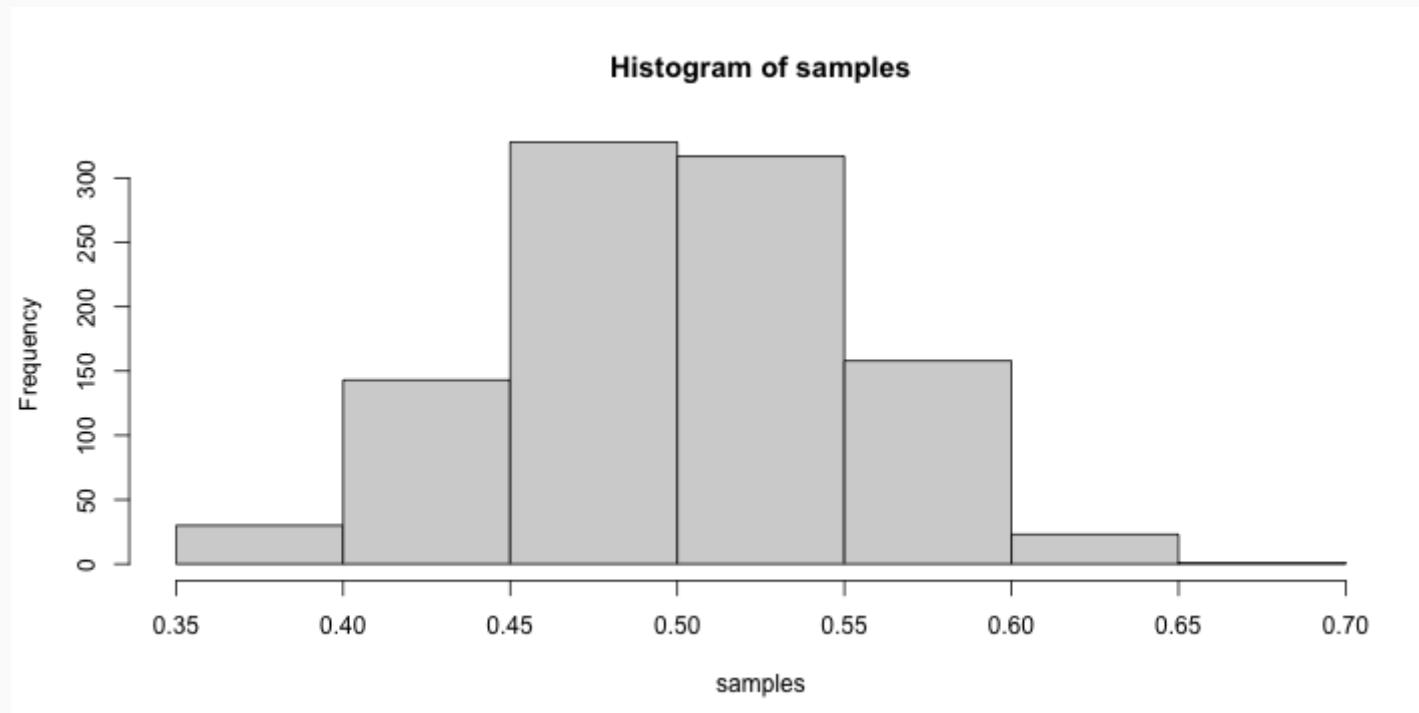
```
M <- 1000
samples <- numeric(length=M)
for(i in seq_len(M)) {
  samples[i] <- mean(sample(pop, size=30))
}
head(samples, n=8)
```

```
## [1] 0.3625186 0.4650553 0.4908226 0.5003795 0.5566081 0.4773403 0.4808297 0.4353974
```



# Sampling Distribution

```
hist(samples)
```



# Central Limit Theorem (CLT)

Let  $X_1, X_2, \dots, X_n$  be independent, identically distributed random variables with mean  $\mu$  and variance  $\sigma^2$ , both finite. Then for any constant  $z$ ,

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

where  $\Phi$  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:

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

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



# CLT Shiny App

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



# Standard Error

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

```
## [1] 0.5151478
```

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

```
## [1] 0.05238345
```



# 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

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

```
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] 0.4124763 0.6178194
```



# Confidence Intervals (cont.)

We are 95% confident that the true population mean is between 0.4124763, 0.6178194.

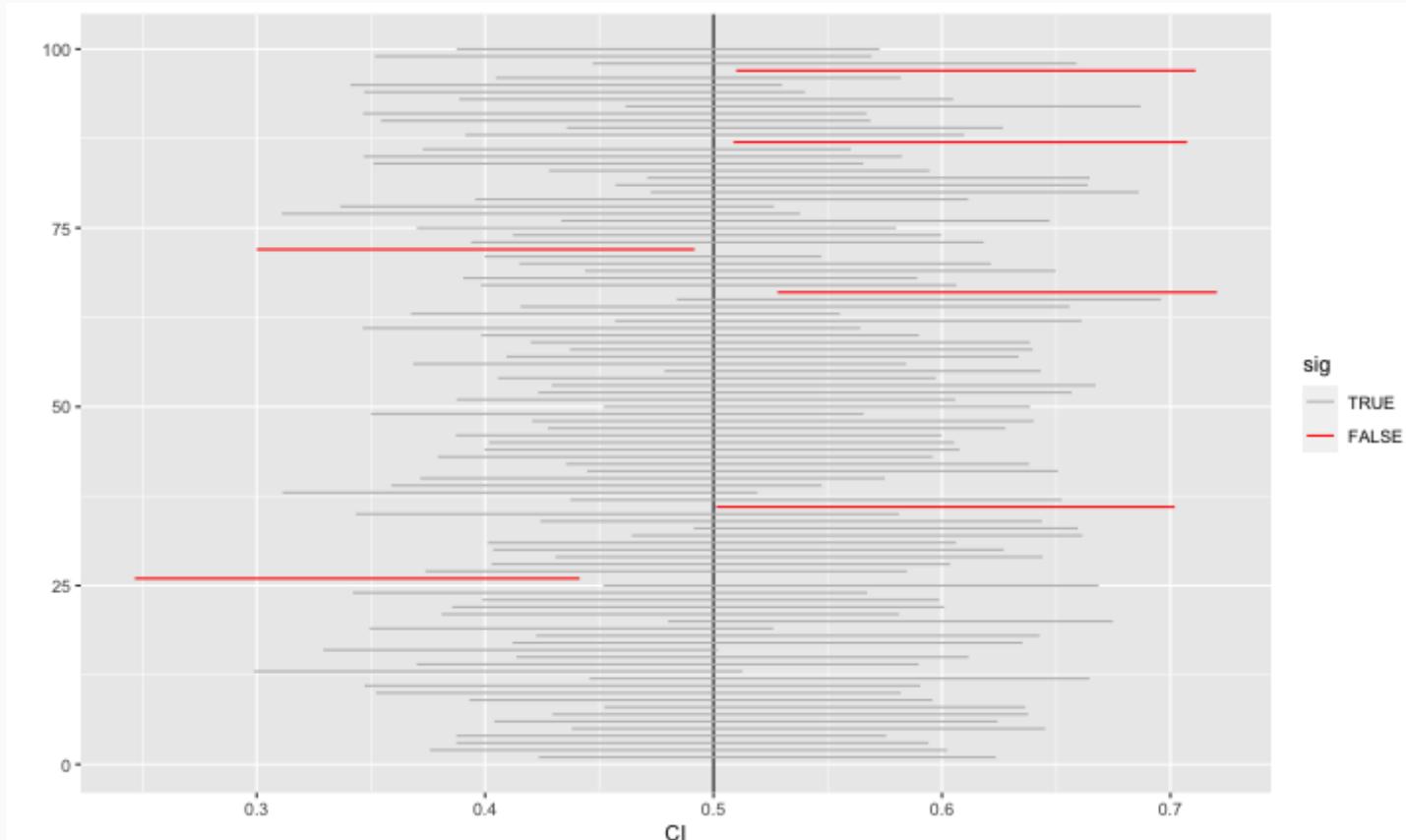
That is, if we were to take 100 random samples, we would expect at least 95% of those samples to have a mean within 0.4124763, 0.6178194.

```
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'))
```



# One Minute Paper

Complete the one minute paper:

<https://forms.gle/ENFqTnDB5fJDw3kx9>

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

