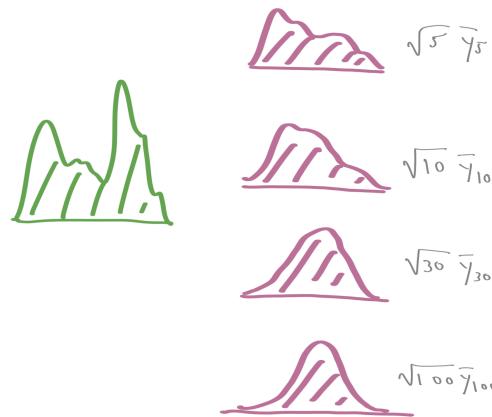


Probability Review



Statistical Experimental Design

Kris Sankaran | UW Madison | 14 September 2021

1. Estimators

2. Central Limit

3. Exercise

4. Common Distributions

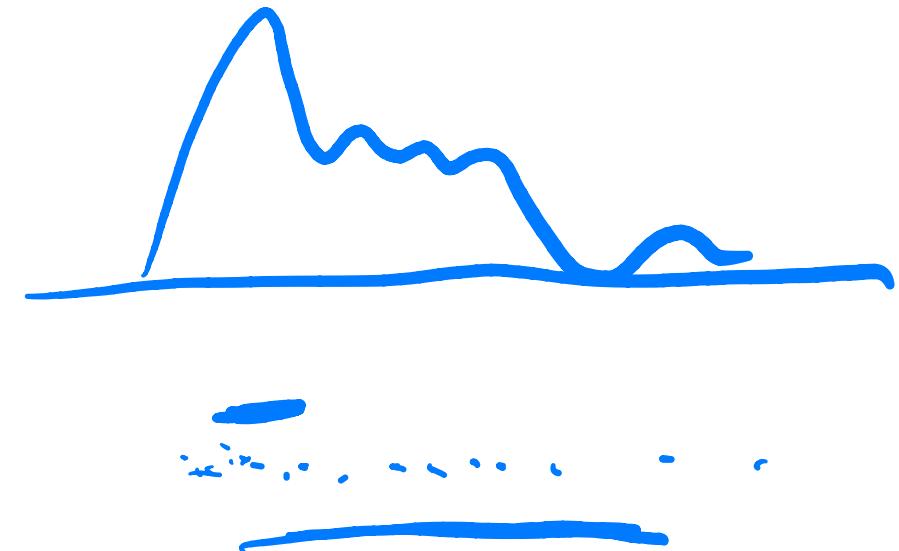
5. Exercise

Announcements

- For today's exercises, we have groups assigned on canvas
 - Sorted according to whether project group is found
 - Arrangement around room
- If you do not find a group, we will randomly assign you (there is no penalty)
- Let us know if you have a midterm conflict by Sunday 9/19
- There is a week 2 feedback form on Canvas

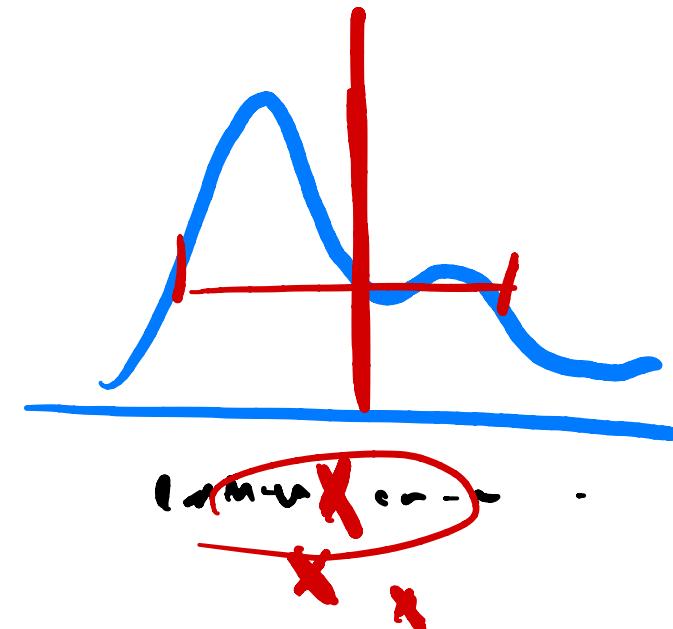
Sample → Population Properties

- The distribution \mathbf{P} summarizes our model of the world
- We will be happy if we can make precise statements about it
 - Where is the center?
 - How spread out is it?
 - What is its shape?
 - How many peaks does it have?
 - ...



Statistical Estimators

- We only have access to a sample x_1, \dots, x_n from \mathbf{P} .
 - Assume they are all independent replicates
- We can define functions of the sample in order to estimate properties of \mathbf{P}
 - $\bar{x} \approx \mu(\mathbf{P})$
 - $\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 \approx \sigma^2(\mathbf{P})$



How to evaluate estimators?

- Suppose we have 100 samples from a random normal distribution with unknown mean μ
- Which is a better estimator of the mean? Why?

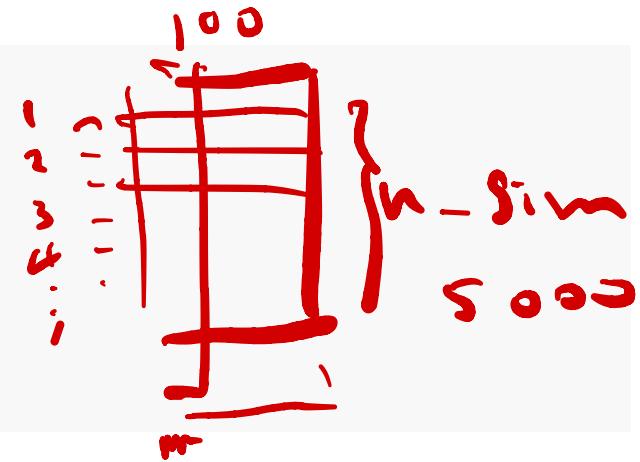
$$\text{Option 1: } \bar{x} = \frac{1}{10} \sum_{i=1}^{10} x_i$$

$$\text{Option 2: } \bar{x} = \frac{1}{100} \sum_{i=1}^{100} x_i$$

Low Bias and Variance

The code below generates 5000 datasets with population means 2.5, then computes the two estimators.

```
n_sim <- 5e3  
datasets <- matrix(rnorm(n_sim * 100, 2.5), n_sim, 100)  
means <- data.frame(  
  id = 1:n_sim,  
  partial = rowMeans(datasets[, 1:10]),  
  full = rowMeans(datasets))
```



Low Bias and Variance

The code below generates 5000 datasets with population means 2.5, then computes the two estimators.

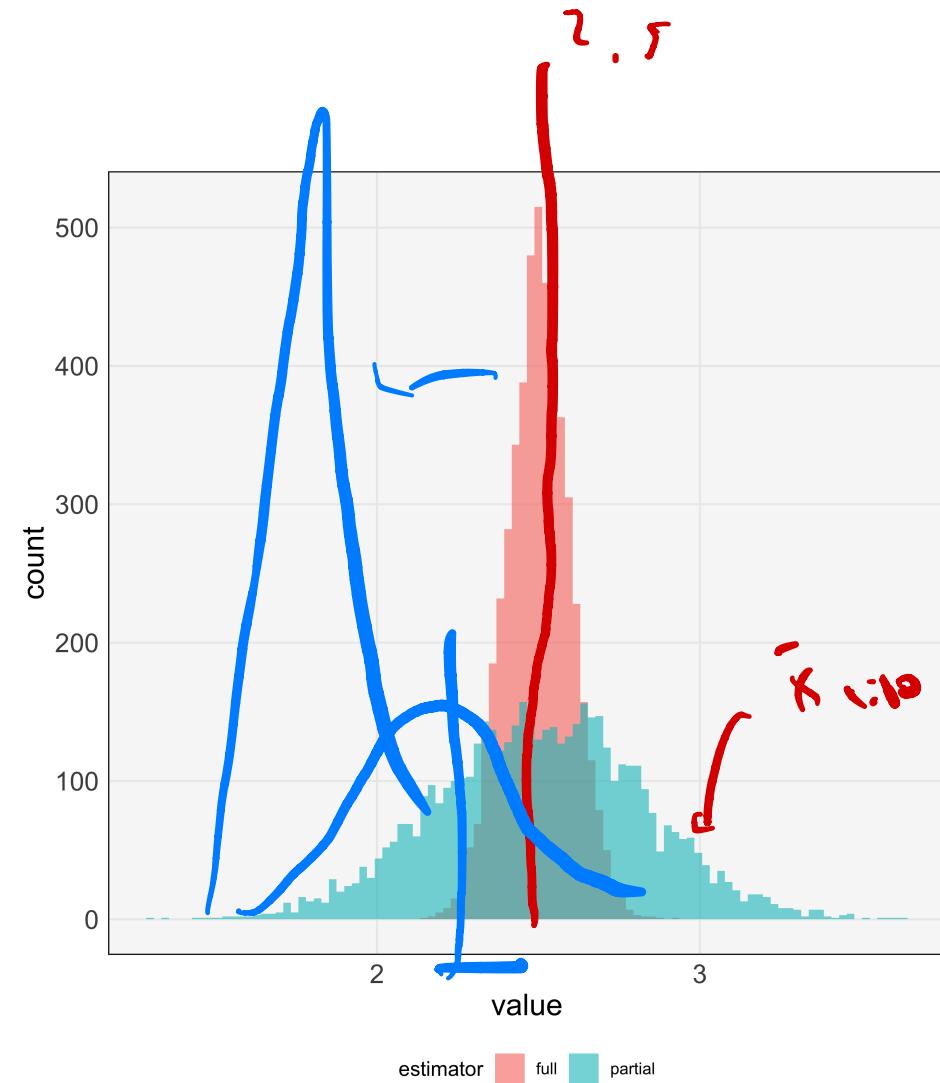
```
head(means)
```

```
##   id partial      full
## 1  1 2.615488 2.688658
## 2  2 2.026779 2.503644
## 3  3 1.561872 2.319888
## 4  4 3.065502 2.612235
## 5  5 2.665711 2.546388
## 6  6 1.921372 2.593992
```

Low Bias and Variance

- Unbiased: The statistic is centered around the truth
- Low Variance: The spread of the statistic is low
- Using all the data gives an estimate with lower variance than using only a fraction

$$T^3(X) = \bar{X} - 1$$



Central Limit Theorem

Theorem Statement

$$\mu = 0$$

$$\sigma^2 = 1$$

If y_i are drawn i.i.d. from some distribution with mean μ and variance σ^2 , then

$$y_1 \rightarrow \sqrt{n} \frac{y_1 - \mu}{\sigma} \stackrel{\text{N}(0,1)}{\sim} \mathcal{N}(0,1).$$

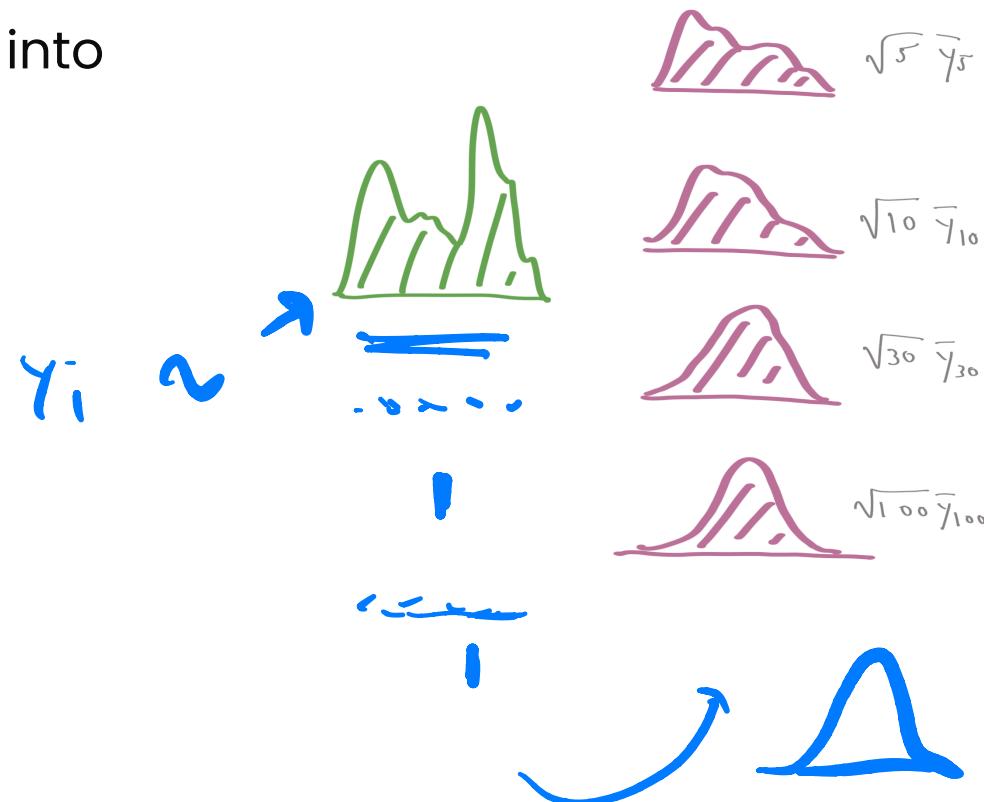
$$y_1, y_2 \xrightarrow{\sqrt{2}} \left(\frac{y_1 + y_2}{2} \right)$$

$$\vdots \qquad \vdots$$

$$y_1, y_2, \dots, y_{100} \xrightarrow{\sqrt{100}} \left(\frac{y_1 + \dots + y_{100}}{100} \right) \stackrel{\text{N}(0,1)}{\sim} \mathcal{N}(0,1)$$

Theorem Importance

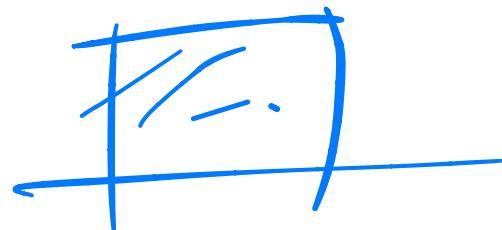
This theorem reduces calculations across arbitrary distributions into calculations with normal distributions.



Exercise Warm-Up

What will be the shape of the histogram in the block below? Why?

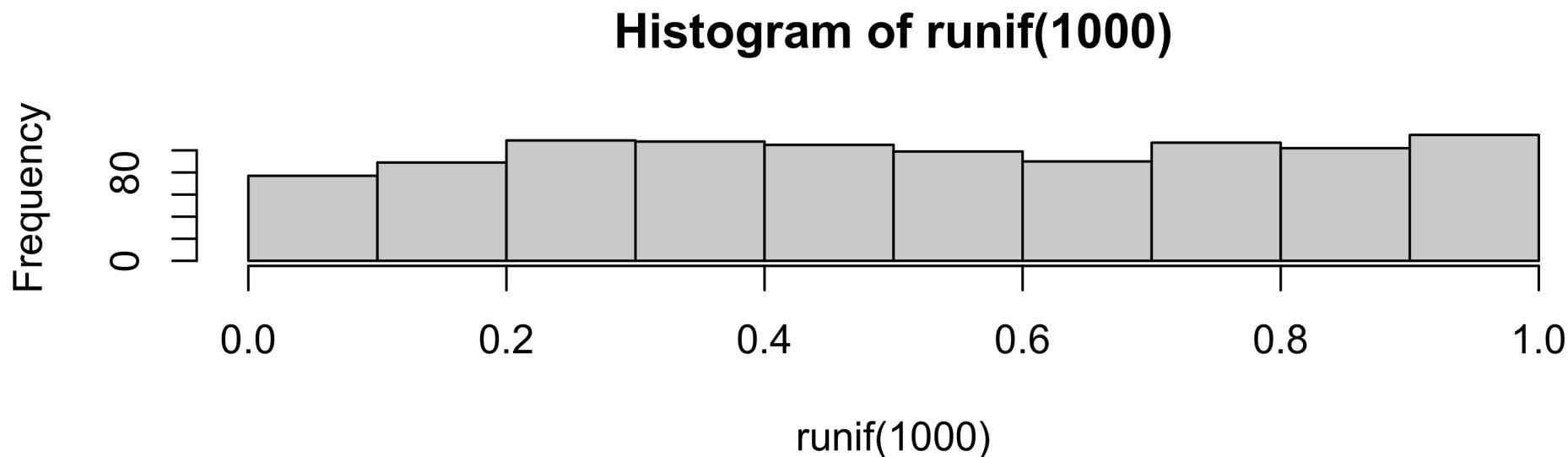
```
#hist(runif(1000))
```



Exercise Warm-Up

What will be the shape of the histogram in the block below? Why?

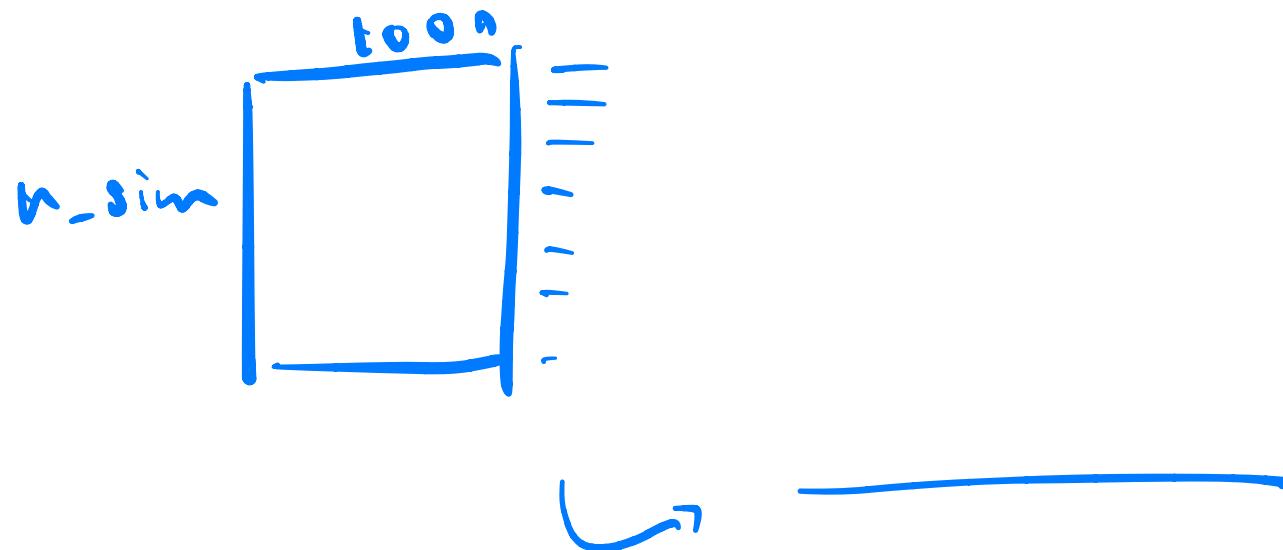
```
hist(runif(1000))
```



Exercise Warm-Up

What will be the shape of the histogram in the block below? Why?

```
datasets <- matrix(runif(n_sim * 1000), nrow = n_sim, ncol = 1000)  
#hist(rowMeans(datasets), breaks = 100)
```



$$\frac{1}{3} - \frac{1}{4}$$

Exercise Warm-Up

~~100~~ 0.5

$$\frac{\sqrt{1000}(\bar{T} - 0.5)}{\sqrt{1/12}} \approx N(0, 1)$$

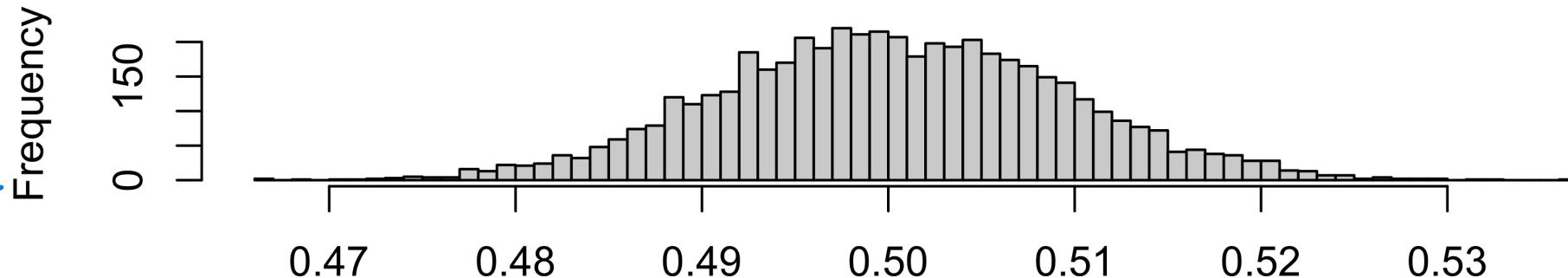
What will be the shape of the histogram in the block below? Why?

```
datasets <- matrix(runif(n_sim * 1000), nrow = n_sim, ncol = 1000)
hist(rowMeans(datasets), breaks = 100)
```



Histogram of rowMeans(datasets)

$$\sqrt{1000} \cdot \frac{(\bar{Y} - \mu)}{\sigma} \approx N(0,1)$$



SD નો. ૧૩

$$\sqrt{12,000} (\bar{y} - 0.5) \approx N(0, 1) \text{ Means(datasets)}$$

$$\bar{Y} \approx N(0.5, \frac{1}{12,32})$$

Exercise

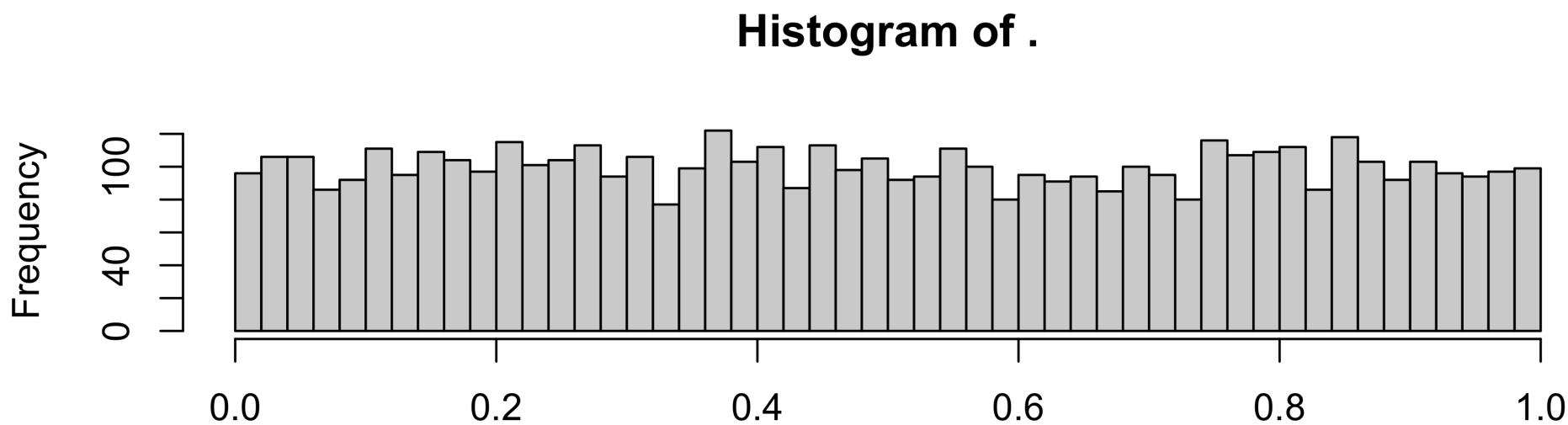
What will be the differences between the three histograms below? Why?

```
plot_hist <- function(sample_size) {  
  matrix(runif(n_sim * sample_size), n_sim, sample_size) %>%  
    rowMeans() %>%  
    hist(breaks = 50)  
}  
  
#plot_hist(1)  
#plot_hist(2)  
#plot_hist(1000)
```

Exercise

What will be the differences between the three histograms below? Why?

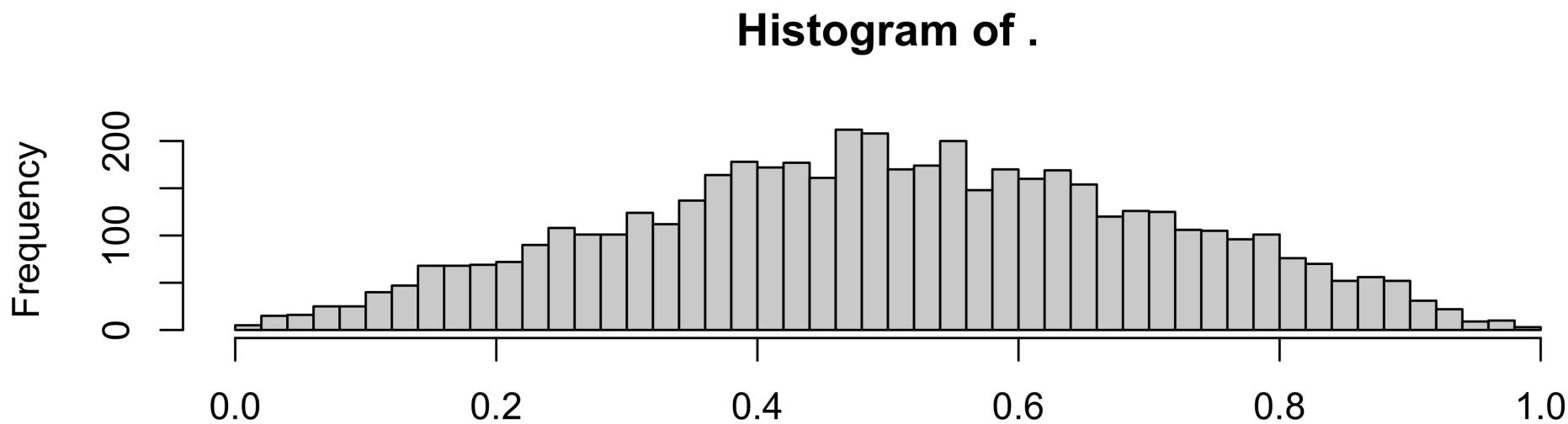
```
plot_hist(1)
```



Exercise

What will be the differences between the three histograms below? Why?

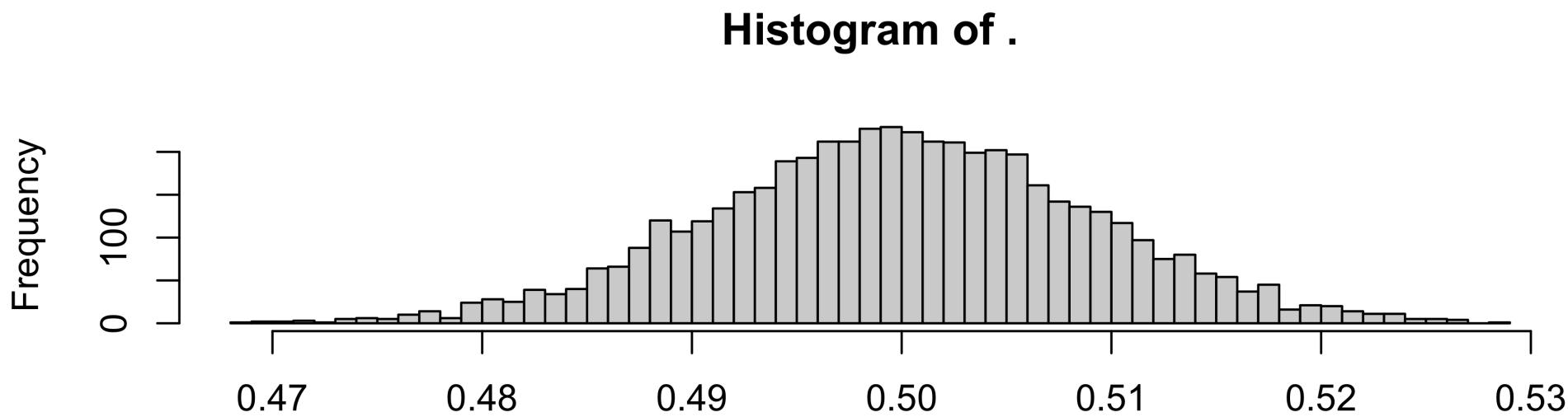
```
plot_hist(2)
```



Exercise

What will be the differences between the three histograms below? Why?

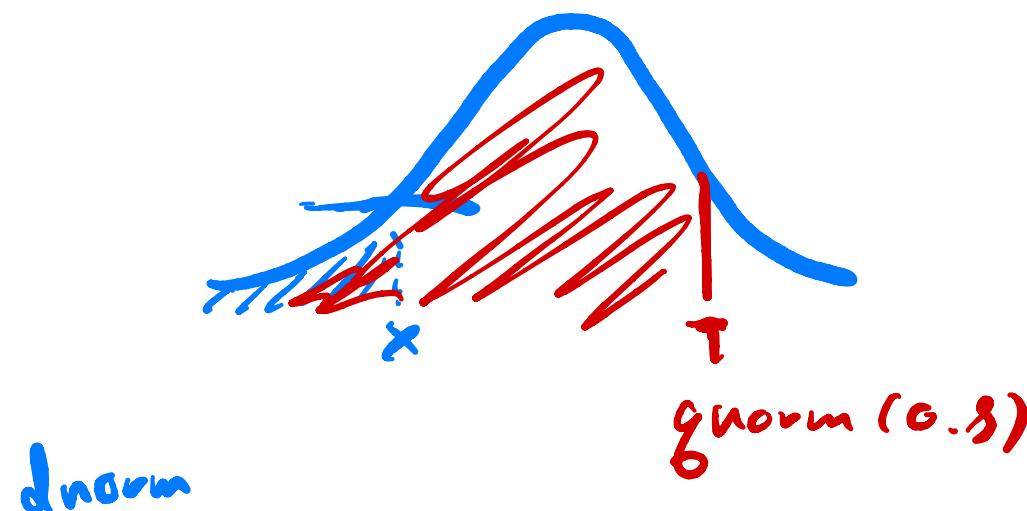
```
plot_hist(1000)
```



Useful Distributions

R Syntax

- $r\{\text{name of density}\}(n)$ will sample n points
- $d\{\text{name of density}\}(x)$ will compute the probability density at x
- $p\{\text{name of density}\}(x)$ will integrate the density up to x
- $q\{\text{name of density}\}(p)$ will find the x value of the density at the p quantile



t Distribution

- We can use the `dt` function to compute the density of the *t* distribution.
- Evaluate over a grid of `x` values to make a plot

```
x <- seq(-3, 3, length.out = 100)
data.frame(x, density = dt(x, df=2)) %>%
  ggplot() +
  geom_line(aes(x, density))
```

t Distribution

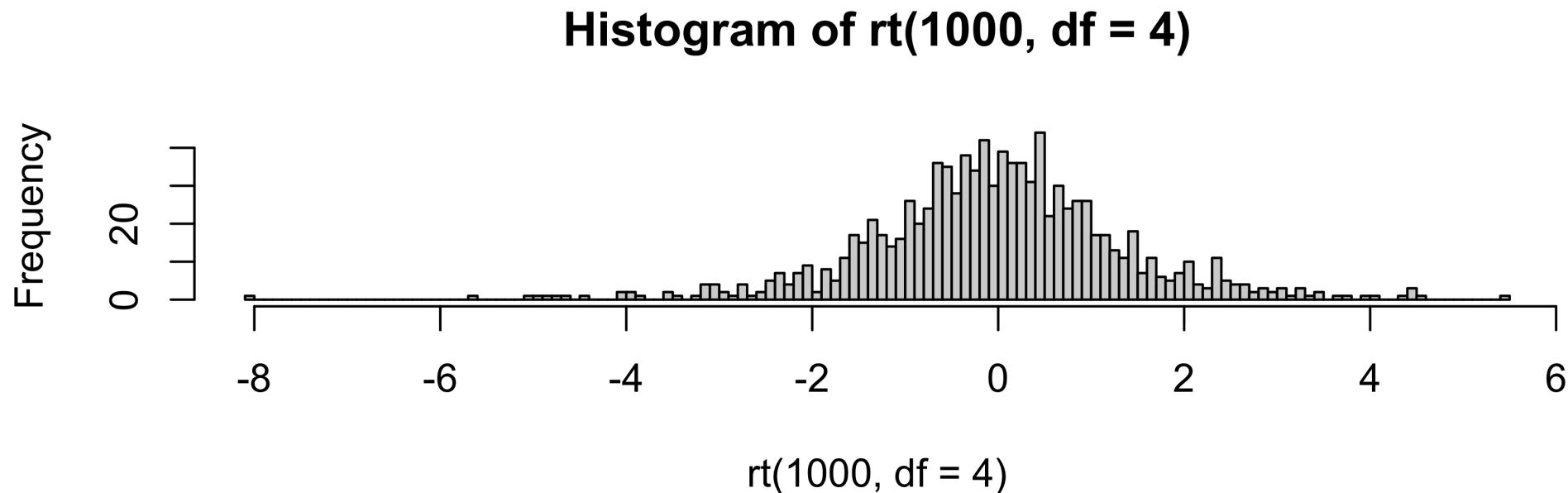
It has a hyperparameter, called the "degrees-of-freedom" (df). Smaller df means heavier tails.

```
densities <- map_dfr(  
  seq(1, 10, .5),  
  ~ data.frame(x, density = dt(x, .), df = .)  
)  
  
ggplot(densities) +  
  geom_line(aes(x, density, col = df, group = df)) +  
  theme(legend.position = "right")
```

t Distribution

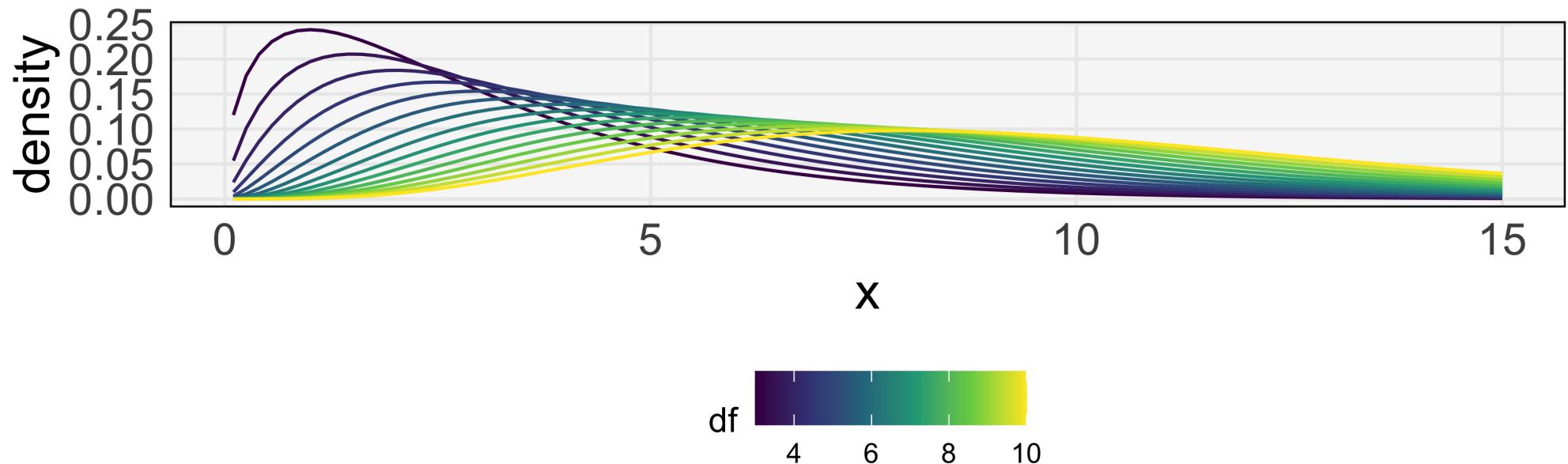
Here are examples generating samples and computing quantiles.

```
hist(rt(1000, df = 4), breaks = 100)
```



chi-square Distribution

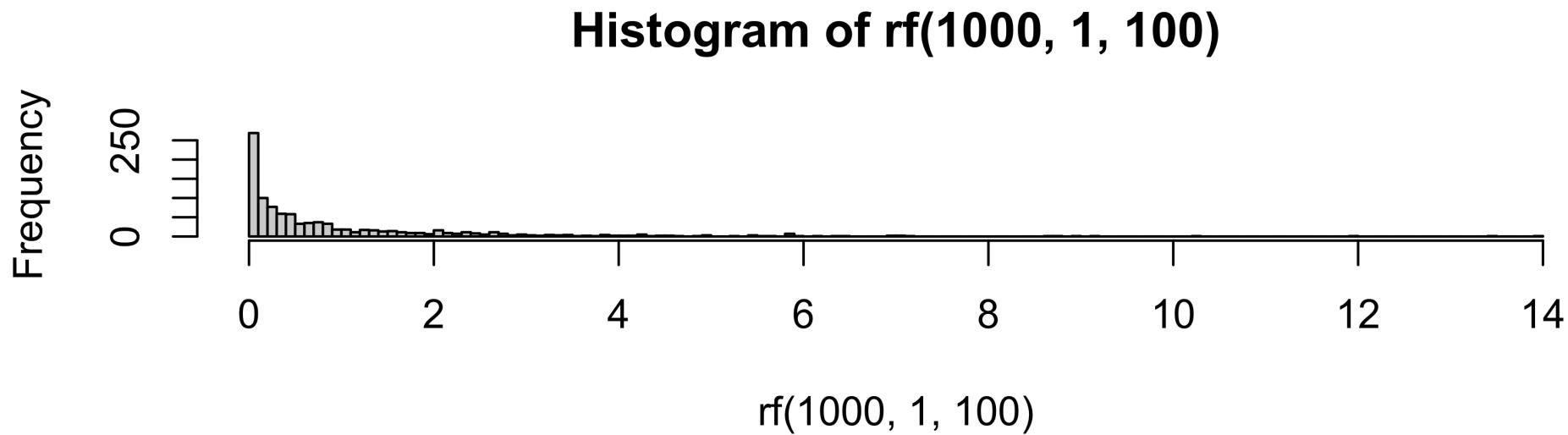
The chi-square distribution is nonnegative with one parameter and can be referenced using [\(prefix\)chisq](#).



F Distribution

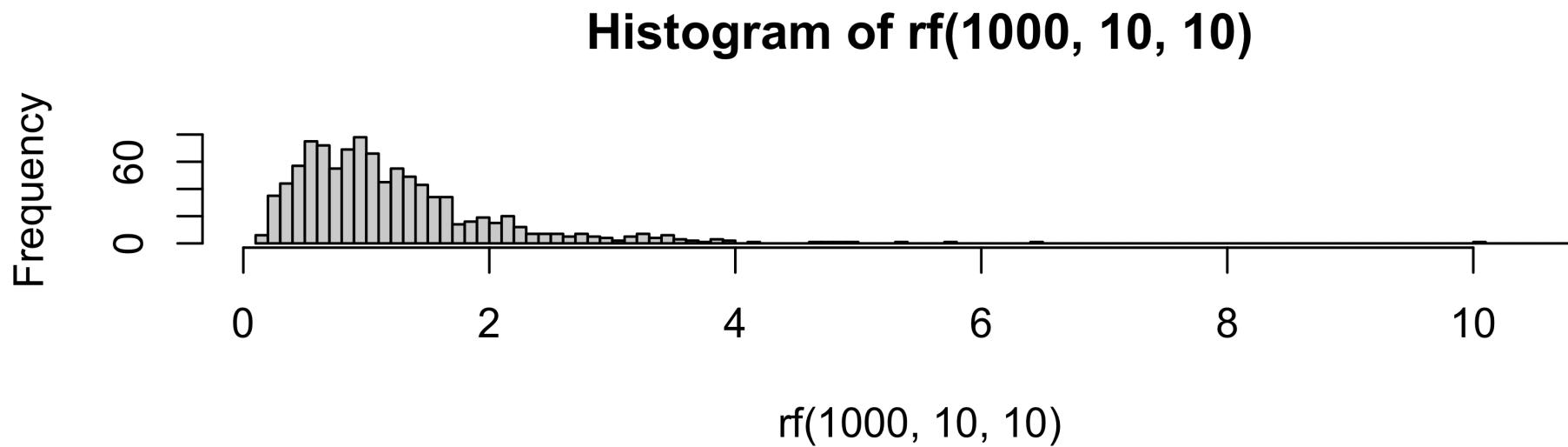
The F distribution is also nonnegative, but has two parameters.

```
hist(rf(1000, 1, 100), breaks = 100)
```



F Distribution

```
hist(rf(1000, 10, 10), breaks = 100)
```



Where these distributions arise

Main Idea

- We will often want the distribution of a particular statistic
- We may know the distribution of individual terms within the statistic
- Learning how one distribution arises as a function of another is key

A hand-drawn diagram in red ink. On the left, there is a tall vertical rectangle representing a population distribution, with a horizontal wavy arrow underneath it pointing to the right. To its right is a smaller, shorter rectangle representing a sample distribution, containing several diagonal lines. Below these two rectangles is a horizontal wavy arrow pointing to the right, followed by a horizontal bar with a tilde over it, representing the sample mean (\bar{x}). To the right of this is a bell-shaped curve representing a normal distribution.

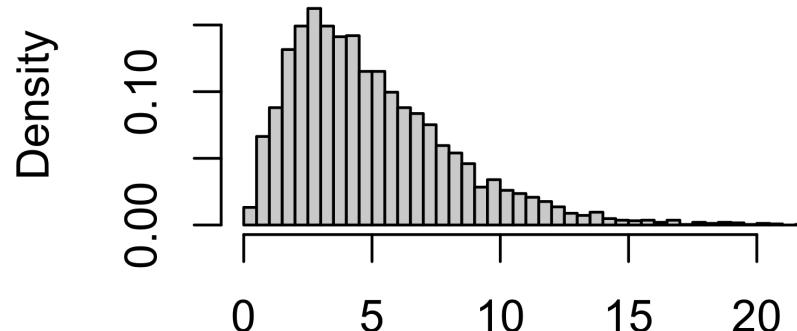
$$\frac{x_1 + \dots + x_{100}}{x_{101} + \dots + x_{200}}$$

Chi-square Distribution

This distribution arises as the sum-of-squares of standard normals. If $z_k \sim \mathcal{N}(0, 1)$, then $\sum_{k=1}^K z_k^2 \sim \chi_K^2$.

```
rchisq(n_sim, 5) %>%  
  hist(breaks = 50, freq=F, ylim = c(0, .18))
```

Histogram of .

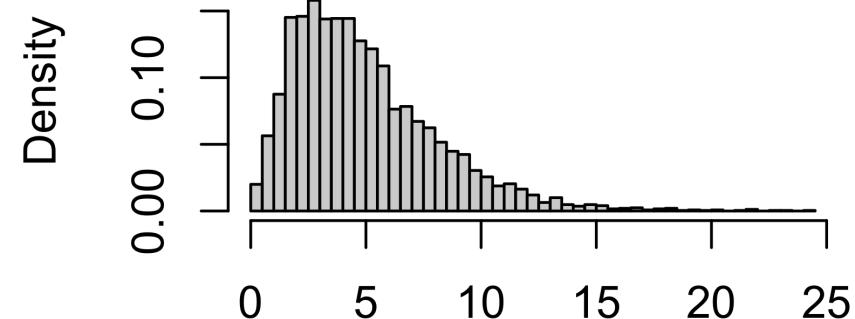


Chi-square Distribution

This distribution arises as the sum-of-squares of standard normals. If $z_k \sim \mathcal{N}(0, 1)$, then $\sum_{k=1}^K z_k^2 \sim \chi_K^2$.

```
matrix(rnorm(n_sim * 5)^2, n_sim, 5) %>%
  rowSums() %>%
  hist(breaks = 50, freq=F, ylim = c(0, .18))
```

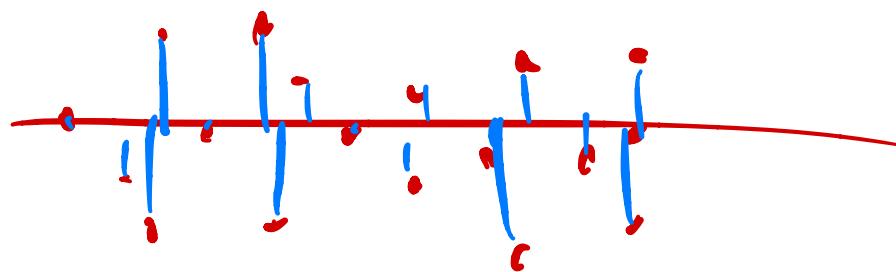
Histogram of .



Chi-square Distribution

A related (but nontrivial) fact is that if $y_i \sim \mathcal{N}(\mu, \sigma^2)$,

$$\frac{1}{\sigma^2} \sum_{i=1}^n (y_i - \bar{y})^2 \sim \chi_{n-1}^2$$



Chi-square Distribution

```
hist(rchisq(n_sim, 9), breaks = 50, col = rgb(0, 0, 1, .6))
datasets <- rerun(n_sim, rnorm(10, 2.5, 1))
ss <- map_dbl(datasets, ~ sum(. - mean(.)) ^ 2))
hist(ss, breaks = 50, col = rgb(0, 1, 0, 0.6), add = TRUE)
```

t Distribution

The t distribution can be formed as the ratio,

$$\frac{\mathcal{N}(0, 1)}{\sqrt{\frac{\chi_K^2}{K}}} \quad \text{w. k. d.f.}$$

This ratio often occurs when we standardize using an estimate of the standard deviation,

$$\frac{\sqrt{n}(\bar{y} - \mu)}{S} \quad \text{w. k. d.f.}$$

where $S = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (y_i - \bar{y})^2}$

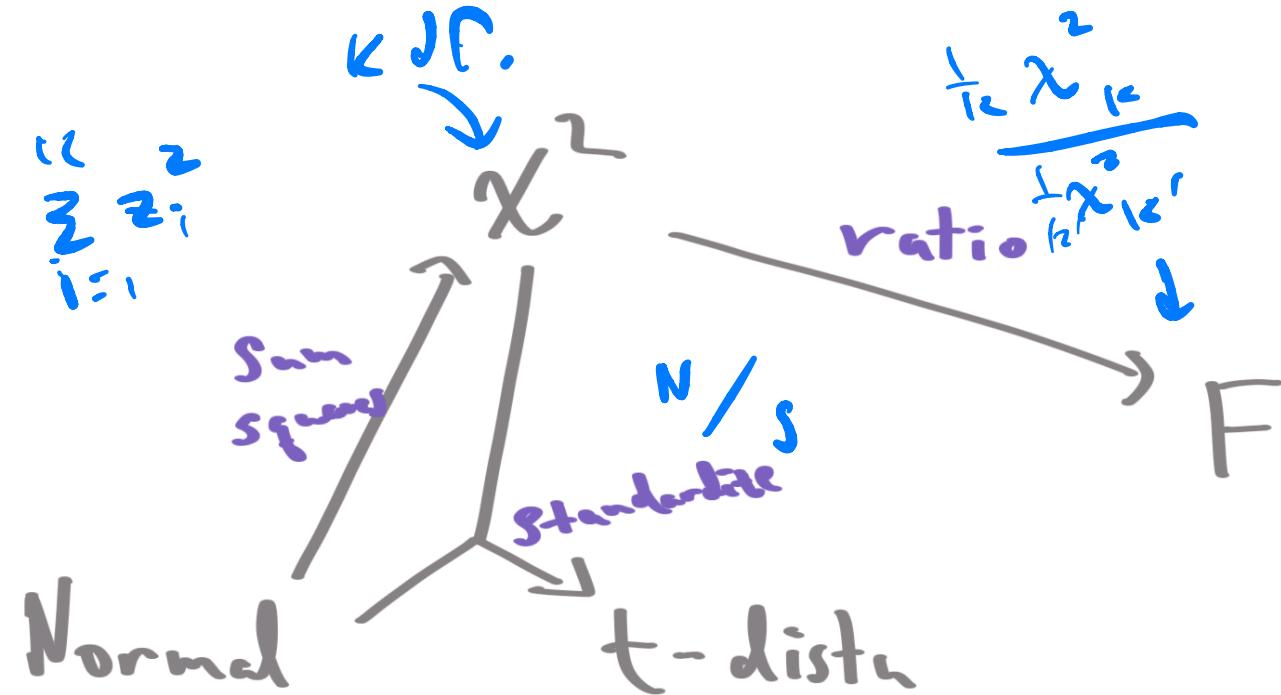
F Distribution

This distribution arises as the ratio,

$$F_{u,v} = \frac{\frac{1}{u}\chi_u^2}{\frac{1}{v}\chi_v^2} \xrightarrow{\text{?}} \frac{\sum (\gamma_i - \bar{\gamma})^2}{\sum (z_i - \bar{z})^2}$$

Since chi-squares come up whenever we compute sums-of-squares of normals, this statistic will arise whenever we want to compare two different sums-of-squares.

Summary



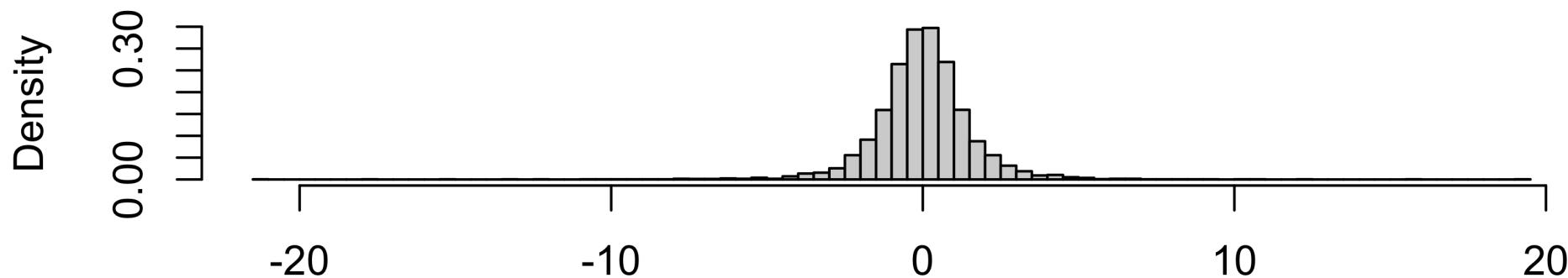
Exercise

Write a small simulation to generate t -distributed variables (with whatever d.f. you want), without using `rt`.

- Bonus: Make a histogram
- Bonus: Use only `rnorm()`.

Solution 1

```
df <- 3  
(rnorm(n_sim) / sqrt(rchisq(n_sim, df) / df)) %>%  
  hist(breaks = 100, freq = F, xlab = NULL, main = NULL)
```



Solution 2

```
normalized_stat <- function(n) {  
  y <- rnorm(n)  
  sqrt(n) * mean(y) / sd(y)  
}  
  
unlist(rerun(n_sim, normalized_stat(df + 1))) %>%  
  hist(breaks = 100, freq = F, xlab = NULL, main = NULL)
```