Bootstrapping and Confidence Intervals

October 5, 2023

Bootstrap Example

14 = chocolate 4 = vanilla

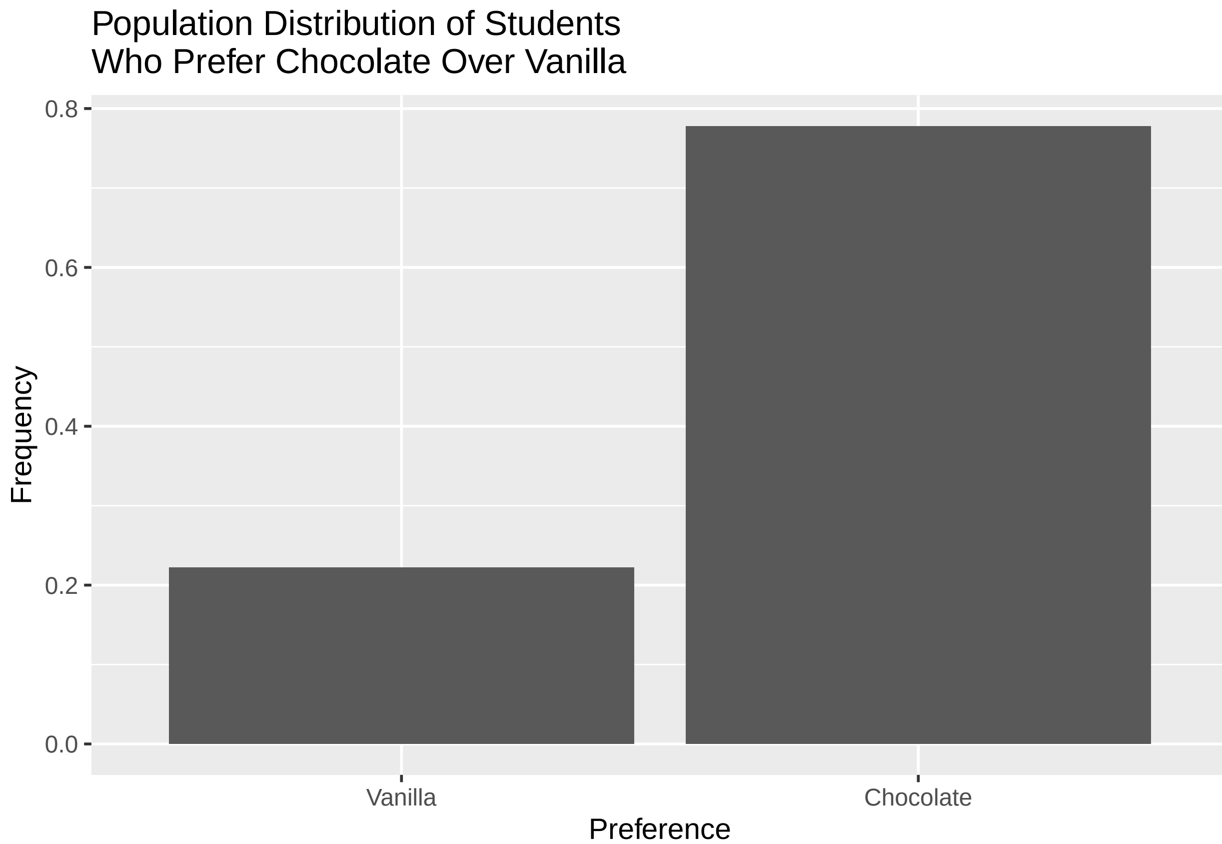
What is the true proportion of CSUEB masters students who prefer chocolate over vanilla?

Let $X_1, X_2, \cdots, X_n \overset{iid}{\sim} Bern(p = 14/18)$.

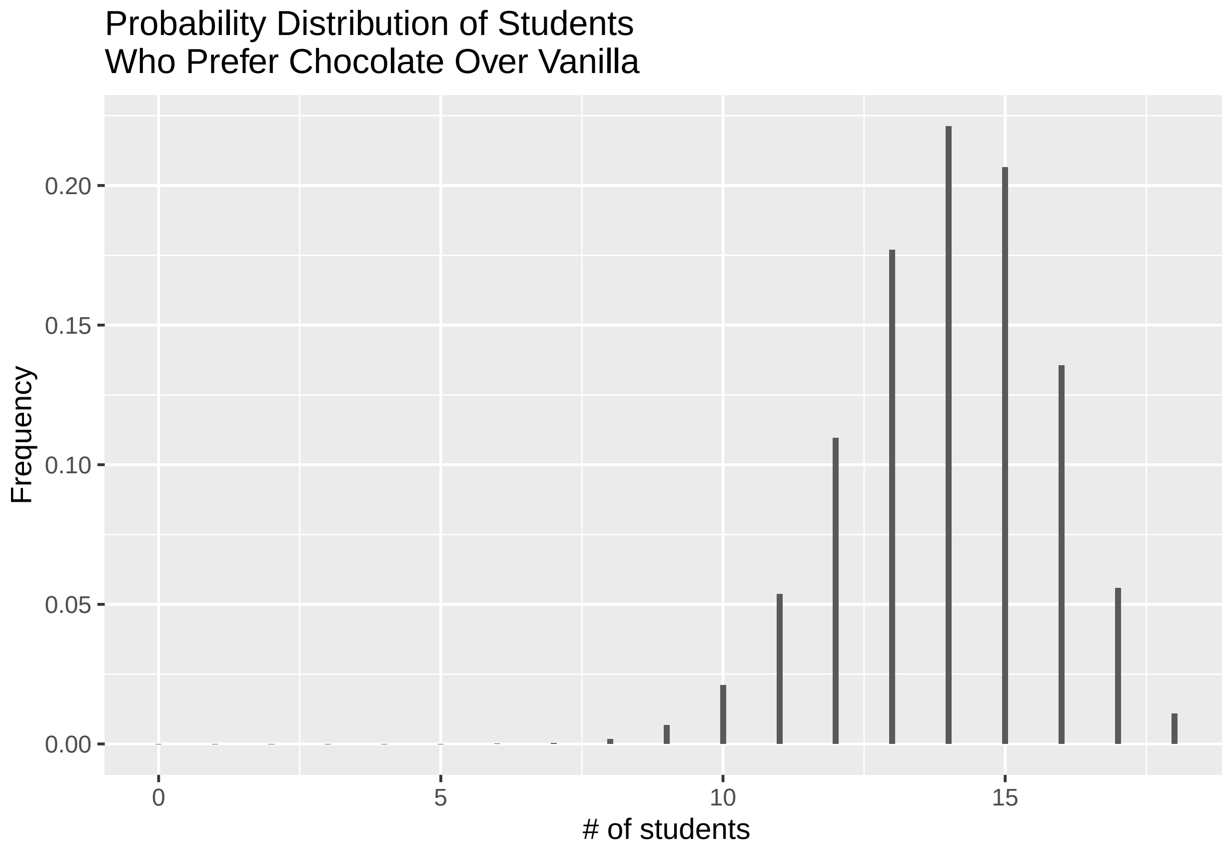
Then $E[X_i] = p$ and $Var[X_i] = p(1-p)$.

Population distribution of X

```
library(ggplot2)
library(dplyr)

n <- 1
phat <- 14/18
x <- c(0,1)
y <- dbinom(x, size = n, prob = phat)
pop <- data.frame(x = as.factor(x), y = y)

pop %>%
  ggplot(aes(x = x, y = y)) +
  geom_bar(stat = "identity") +
  scale_x_discrete(labels = c("Vanilla", "Chocolate")) +
  labs(x = "Preference",
       y = "Frequency",
       title = "Population Distribution of Students \nWho Prefer Chocolate Over Vanilla")
```

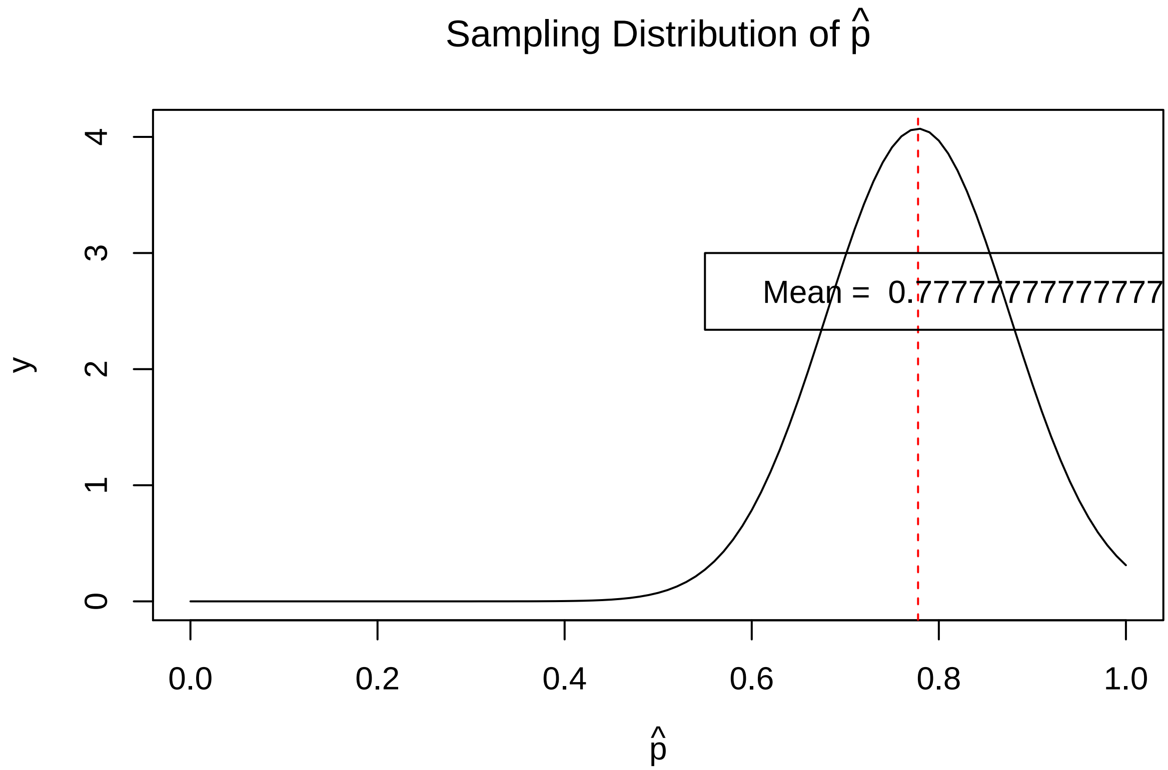



```r
n <- 18
p <- phat # pretend this is the truth
x <- 0:n
y <- dbinom(x, size = n, prob = p)
pop <- data.frame(x = x, y = y)

pop %>%
  ggplot(aes(x = x, y = y)) +
  geom_bar(stat = "identity",
           width = 0.1) +
  labs(x = "# of students",
       y = "Frequency",
       title = "Probability Distribution of Students \nWho Prefer Chocolate Over Vanilla")
```



Sampling Distribution of $\hat{p}$

```
mu_phat <- p
sd_phat <- sqrt(p*(1 - p)/n)

x <- seq(0, 1, 0.01)
y <- dnorm(x, mean = mu_phat, sd = sd_phat)

plot(x, y, type = "l",
     xlab = expression(hat(p)),
     main = expression(paste("Sampling Distribution of ", hat(p)) ))
abline(v = p, col = "red", lty = 2)
legend(0.55, 3, paste("Mean = ", mu_phat))
```

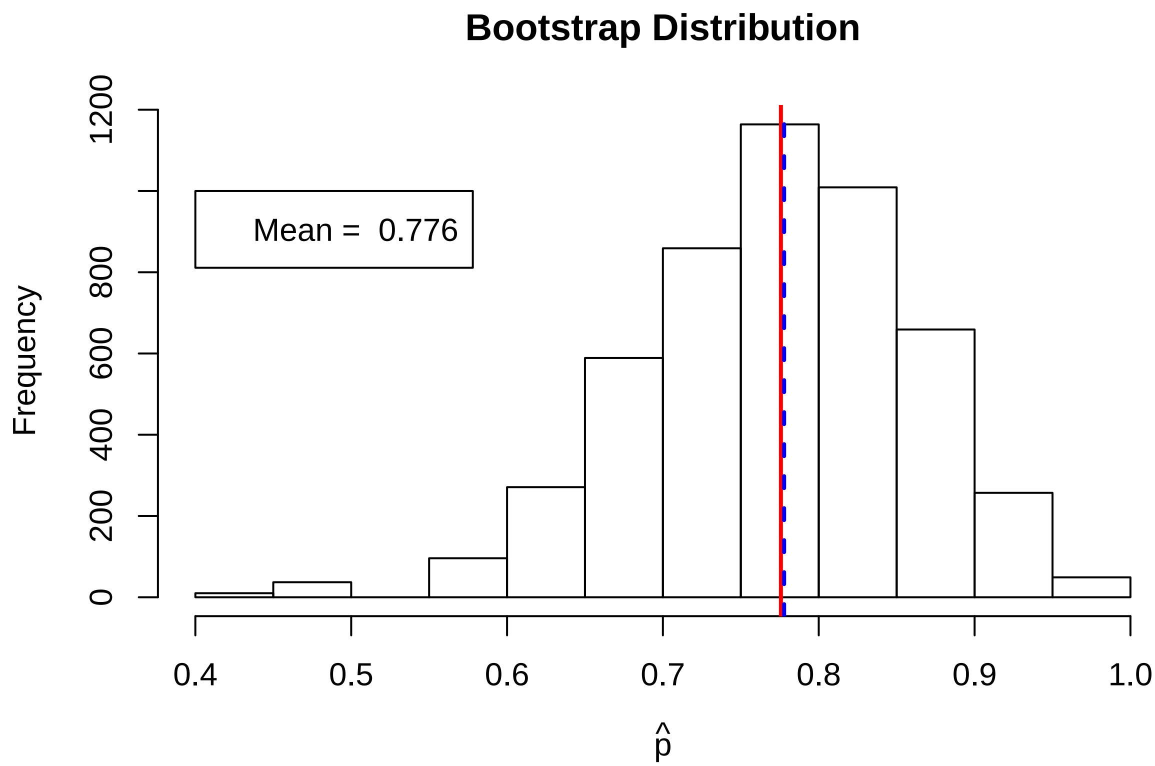


Bootstrap Distribution

```
B <- 5000
n_boot <- n

orig_samp <- c(rep(0,4 ), rep(1,14 ))
table(orig_samp)
```

```
## orig_samp
##  0  1
##  4 14
```

```
phats <- rep(NA, B)

for(i in 1:B){
  boot_samp <- sample(orig_samp, size = n_boot, replace = TRUE)
  phats[i] <- mean(boot_samp)
}

hist(phats, breaks = 10,
     xlab = expression(hat(p)),
     main = "Bootstrap Distribution")
abline(v = mean(phats), col = "red", lwd = 2)
abline(v = p, col = "blue", lwd = 2, lty = 2)
legend(0.4, 1000, paste("Mean = ", round(mean(phats), 3)))
```




```r
sqrt( (p*(1-p))/n)
```

```
## [1] 0.09799079
```

```r
sd(phats)
```

```
## [1] 0.09690646
```

Discussion Questions (in groups):

1. Why is knowing $\hat{p}$ not enough?

- Need to know variability

1. Why do we want to know the distribution of $\hat{p}$?

- Inference

Bootstrap Percentile Confidence Intervals

To compute a 95% bootstrap confidence interval for some parameter, we compute the 2.5th and 97.5th percentiles of the bootstrap distribution.

We can use the `quantile()` function in R.

```r
boot_ci <- quantile(phats, c(0.025, 0.975))
boot_ci
```

```
##      2.5%     97.5%
## 0.5555556 0.9444444
```

Bootstrap CI: We are 95% confident that the true proportion of CSUEB master's students who prefer chocolate over vanilla is between 0.56 and 0.94.

Let's compare this to our confidence interval based on our original sample of data.

```
ci_low <- phat - qnorm(.975) * sqrt(phat*(1-phat)/n)
ci_low
```

```
## [1] 0.5857194
```

```
ci_high <- phat + qnorm(.975) * sqrt(phat*(1-phat)/n)
ci_high
```

```
## [1] 0.9698362
```

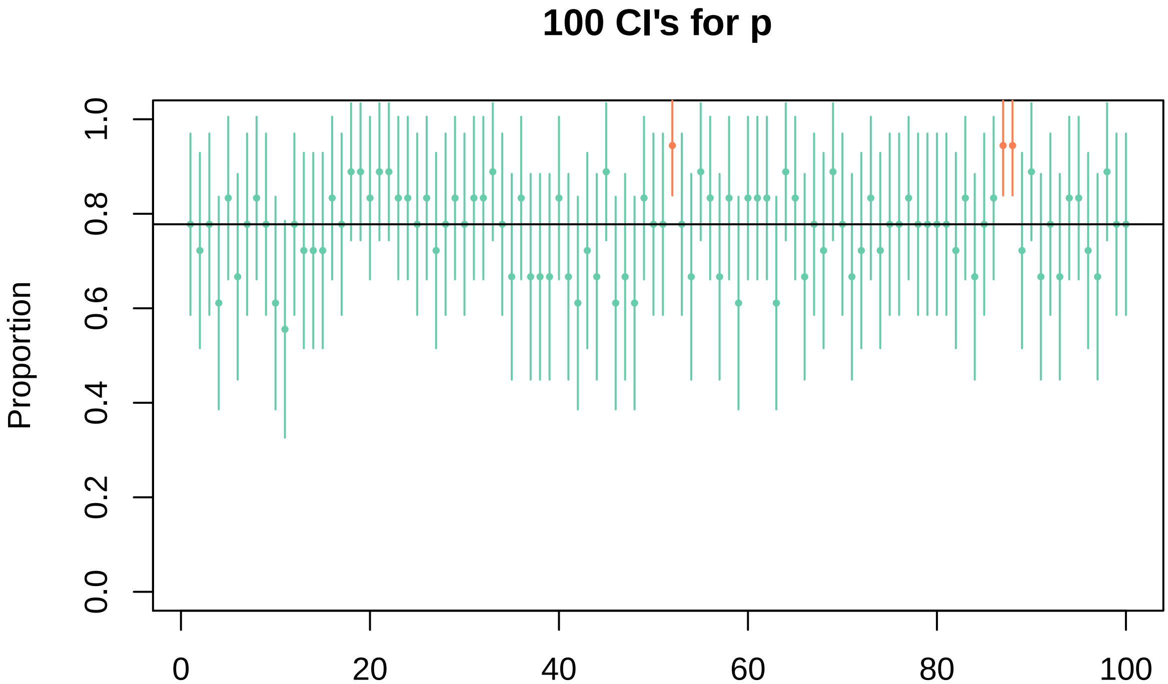
CI from original sample: We are 95% confident that the true proportion of CSUEB master's students who prefer chocolate over vanilla is between 0.59 and 0.97.

Confidence Intervals (via Simulation)

```
n_samp <- 100
phats <- rbinom(n_samp, n, p) / n

ci_low  <- phats - qnorm(0.975) * sqrt(phats*(1 - phats) / n)
ci_high <- phats + qnorm(0.975) * sqrt(phats*(1 - phats) / n)

color <- rep(NA, n_samp)

for(i in 1:n_samp){
  if(p > ci_low[i] & p < ci_high[i]){
    color[i] <- "aquamarine3"
  }
  else color[i] <- "coral"
}
table(color)
```

```
## color
## aquamarine3       coral
##          97           3
```

```
x <- 1:n_samp
plot(x, phats, ylim = c(0,1),
     pch = 16, cex = 0.5, col = color,
     main = "100 CI's for p",
     xlab = "",
     ylab = "Proportion")
segments(x, ci_low, x, ci_high,
         col = color)
abline(h = p, col = "black")
```


100 CI's for p
Proportion
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100