DATA606 - Foundation for Inference

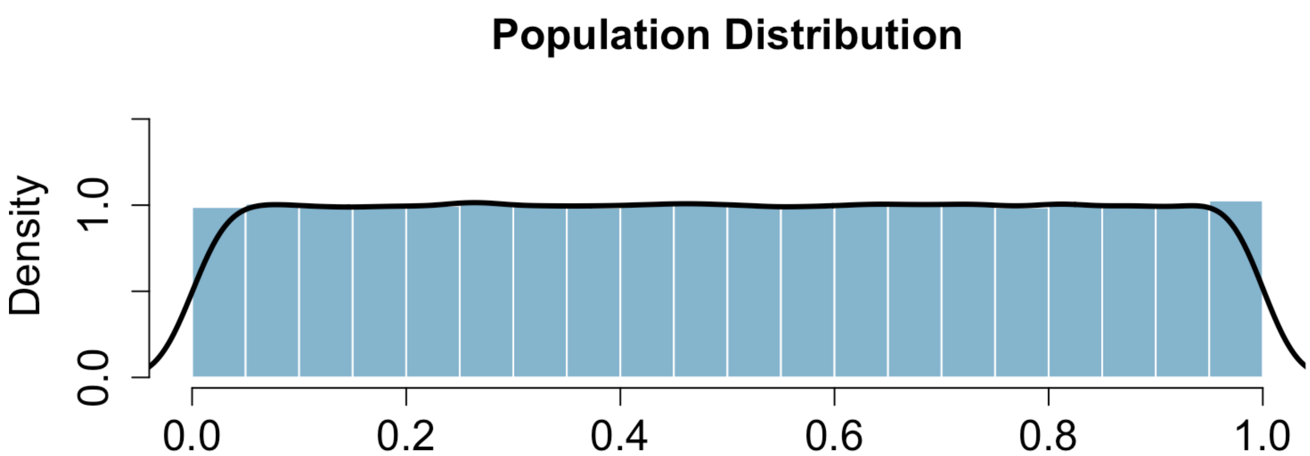
Jason Bryer, Ph.D.

March 2, 2017

Population Distribution (Uniform)

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

```
## [1] 0.5008915
```

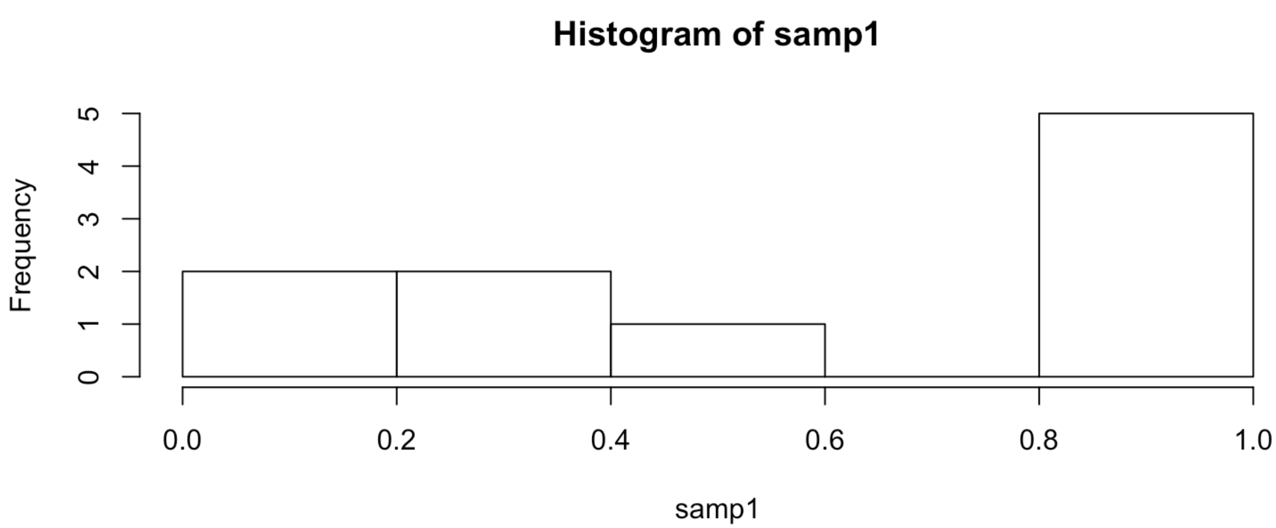


Random Sample (n=10)

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

```
## [1] 0.5745289
```

```
hist(samp1)
```



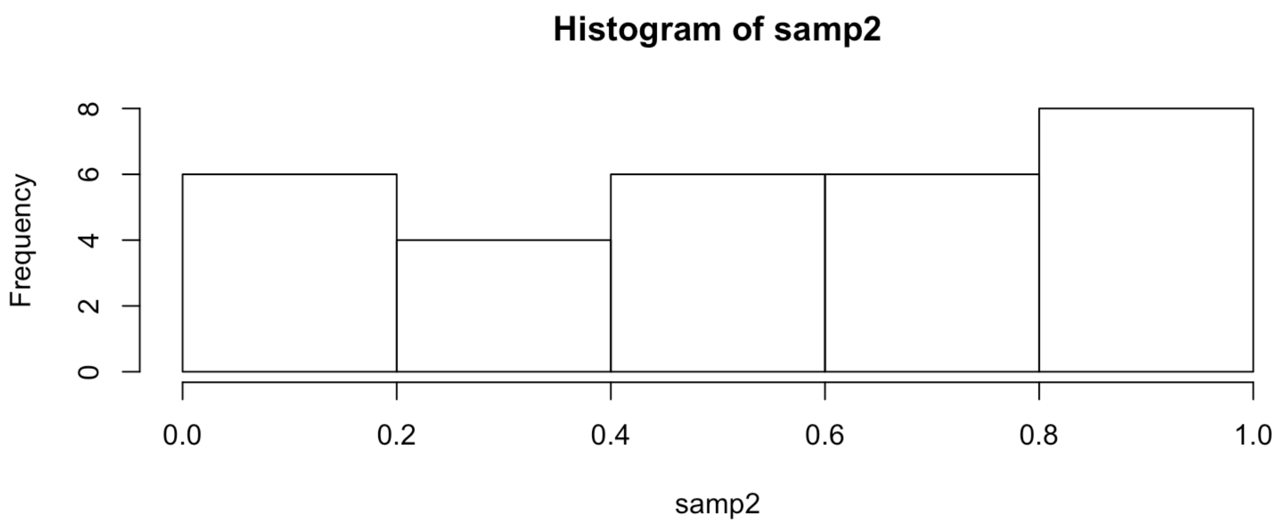
3/17

Random Sample (n=30)

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

```
## [1] 0.5466776
```

```
hist(samp2)
```



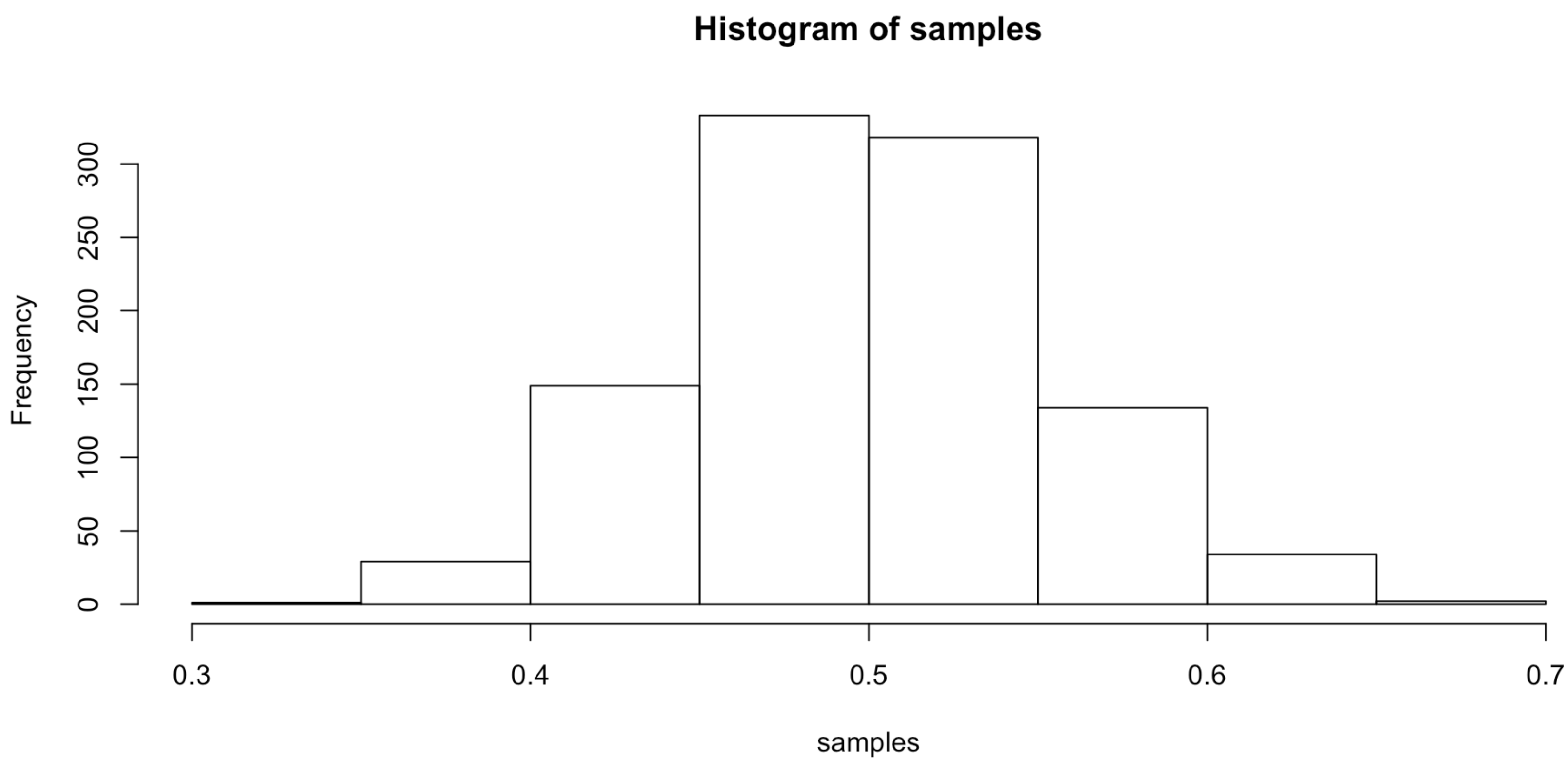
Lots of Random Samples

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

```
## [1] 0.5294721 0.4424369 0.5102434 0.4409382 0.5492505 0.5829651 0.5322821
## [8] 0.5063398
```

Sampling Distribution

```
hist(samples)
```



6/17

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

```
shiny_demo('CLT_mean')
```

Standard Error and Confidence Interval

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

```
## [1] 0.5747245
```

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

```
## [1] 0.04941707
```

The confidence interval is then $\mu \pm 2 \times SE$

```
(samp2.ci <- c(mean(samp2) - 2 * samp2.se, mean(samp2) + 2 * samp2.se))
```

```
## [1] 0.4758903 0.6735586
```

Confidence Intervals

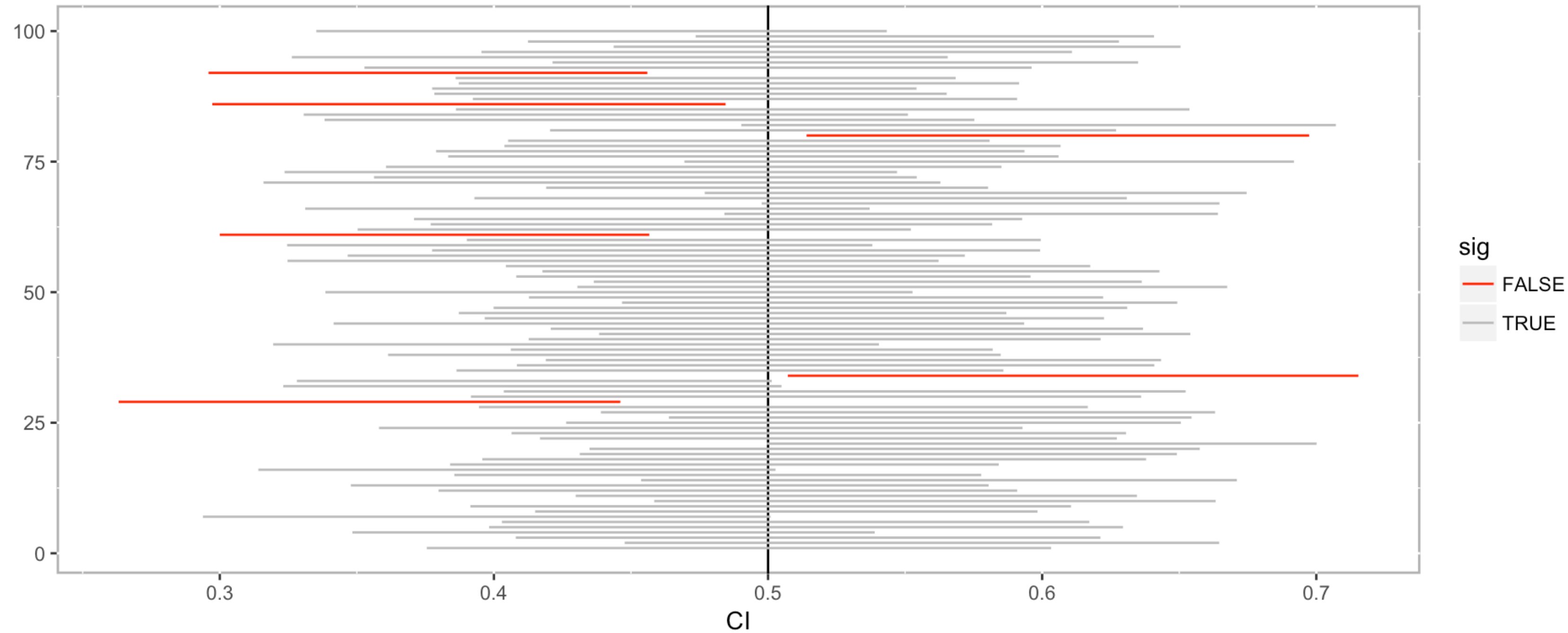
We are 95% confident that the true population mean is between 0.4758903, 0.6735586.

That is, if we were to take 100 random samples, we would expect at least 95% of those samples to have a mean within 0.4758903, 0.6735586.

```
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) - 2 * se,
                mean(samp) + 2 * 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'))
```



12/17

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 value is within the confidence interval, then we to reject the hypothesis.

```
(samp2.ci <- c(mean(samp2) - 2 * sd(samp2) / sqrt(length(samp2)),
               mean(samp2) + 2 * sd(samp2) / sqrt(length(samp2))))
```

```
## [1] 0.4758903 0.6735586
```

Since 0.5 fall within 0.4758903, 0.6735586, we to reject the null hypothesis.

Hypothesis Testing (using -values)

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

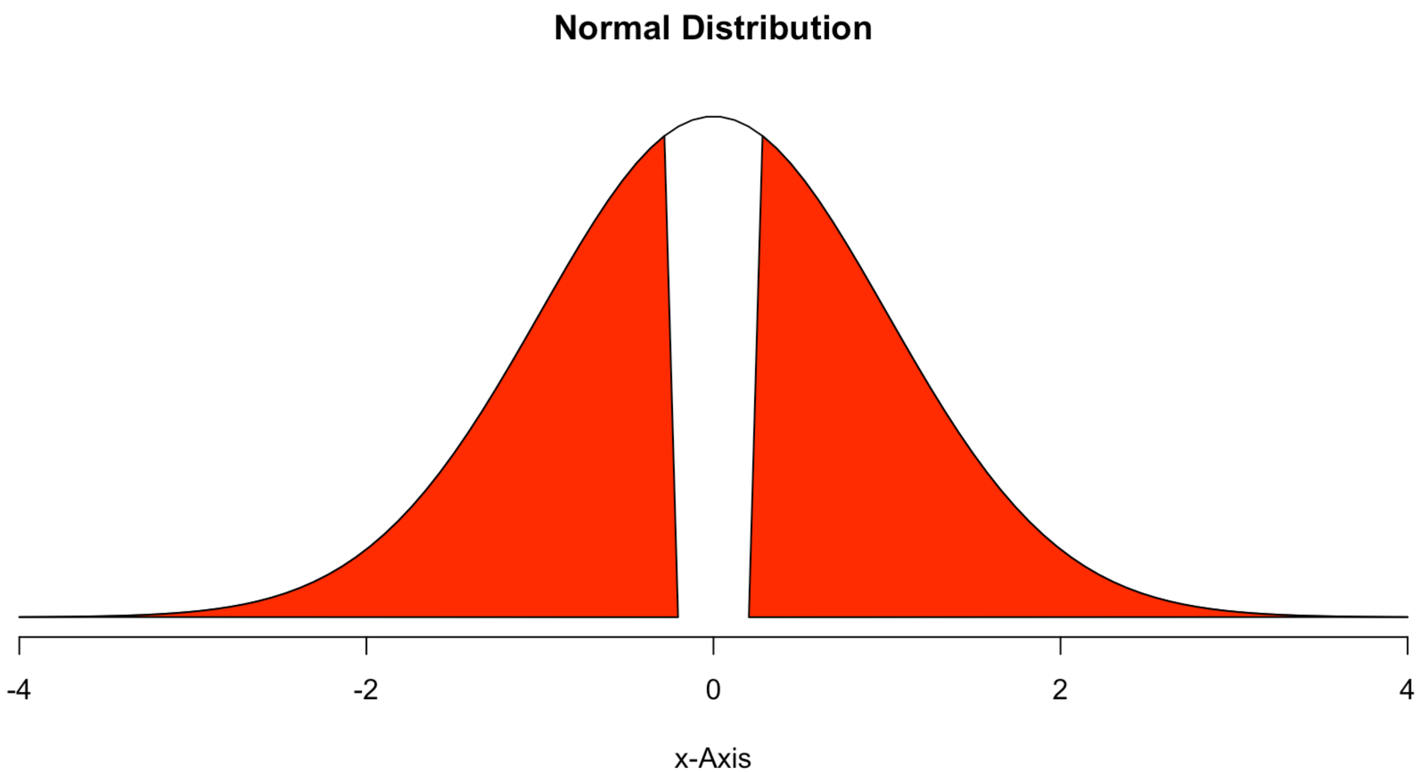
$$Z = \frac{\bar{x} - null}{SE} = \frac{0.49 - 0.50}{0.049} = -.204081633$$

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

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

Hypothesis Testing (using -values)

```
normalPlot(bounds=c(-.204, .204), tails=TRUE)
```



Type I and II Errors

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

Visualizing Type I and Type II errors: http://shiny.albany.edu/stat/betaprob/