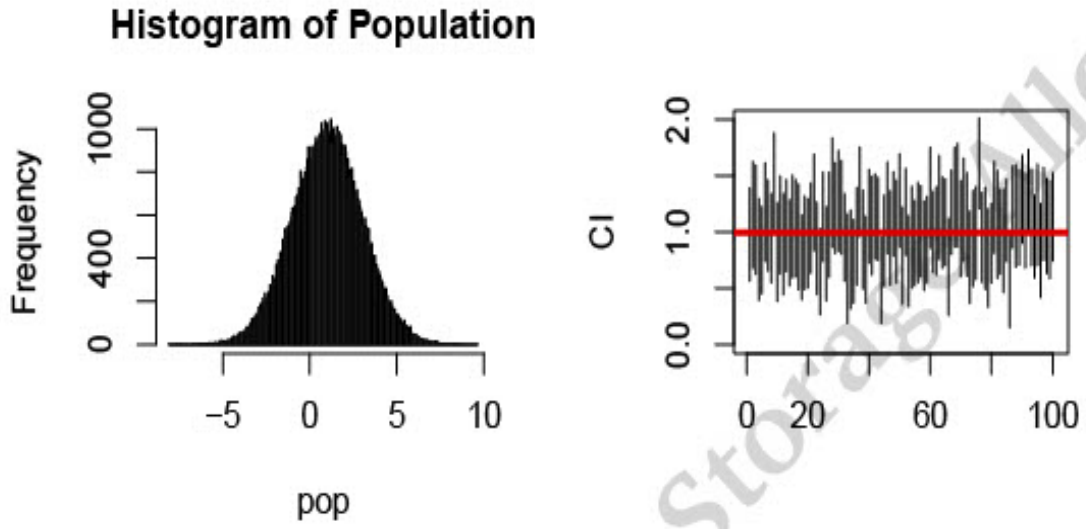
```
plot(c(1, 1), CI[1, ], ylim = c(0, 2), xlim = c(0, 101), ylab = "CI", xlab = '',
     type = "l")
for (i in 2:n.trial) {
  lines(c(i, i), CI[i, ])  # Draw CIs (vertically).
}
abline(h = pop.mean, col = "red", lwd = 3)  # The population mean (horizontally).
```



5.3 Two-Sample, Two-Sided Confidence Interval

The following is data from a Statistics class, when students were asked their gender, and what percentage of time they attend class. We will assume percentage is normally distributed, although it is not.

```
dat <- read.table('attend_dat.txt', header = T)
attendance <- dat[, 1]
gender <- dat[, 2]
pa.boy <- attendance[gender == 0]  # Percent of time attending class for boys.
pa.girl <- attendance[gender == 1]  # Percent of time attending class for girls.

n.boys <- length(pa.boy)  # Number of boys. Same as sum(y == 0).
n.girls <- length(pa.girl)  # Number of girls. Same as sum(y == 1).

# The sample mean of these attendance rates is higher for boys than girls:
mean(pa.boy)

[1] 87.57

mean(pa.girl)

[1] 86.4
```

Suppose you wonder if the two true/population means (of attendance rate) are **different**, then, you need to build a 2-sample, 2-sided CI. We will first start by computing 1-sample, 2-sided CIs for each mean:

```
t.test(pa.boy)$conf.int[1:2]

[1] 79.95 95.19

t.test(pa.girl)$conf.int[1:2]

[1] 81.93 90.87
```

Given the huge overlap between these two confidence intervals, (and given that the two groups - boys and girls - are independent), we can conclude that the data does not provide sufficient evidence to conclude that the attendance rates of boys and girls are different.

Comparing two CIs is not the most elegant way of answering the question. If the comparison of two means (or proportions) is all we care about, then we should compute the CI for the **difference** between the population means (or proportions), i.e., a **2-sample** CI for the difference between means.

```
t.test(pa.boy, pa.girl, alternative = "two.sided")  # Default conf.level = 0.95.


Welch Two Sample t-test

data:  pa.boy and pa.girl
t = 0.27, df = 51, p-value = 0.8
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -7.558  9.891
sample estimates:
mean of x mean of y
    87.57     86.40
```

There are two interpretations:

1. We can be 95% confident that the difference between the true/population means is between -7.559 and 9.891.
2. There is a 95% probability that a 95% CI for the difference between the true means, computed from a random sample, will include the difference between the true/pop means.

Corollary:

The fact that the 2-sided CI, (-7.558, 9.891), includes zero implies that we CANNOT tell if there is a difference between the two proportions. We just cannot say anything. Note, it would be WRONG to conclude that there is NO difference between the true/population means.

5.4 Two-Sample, One-Sided Confidence Interval

Suppose you are NOT interested in whether there is a **difference** between the attendance rates of boys and girls. Instead you are interested in a "weaker" question, namely, is the attendance rate for boys **higher** than that of girls? Denote μ_1 = true/pop mean attendance rate for boys. μ_2 = true/pop mean attendance rate for girls. Then you must build the lower confidence bound for $\mu_1 - \mu_2$. (Or equivalently an upper confidence bound for $\mu_2 - \mu_1$).

```
t.test(pa.boy, pa.girl, alternative = "greater")  # "greater" = LOWER conf. bound.
                                                  # "less" = UPPER conf. bound.
```

Interpretations:

1. We are 95% confident that $\mu_1 - \mu_2$ is larger than -6.11.
2. There is a 95% probability that a random 95% lower confidence bound for the difference will be lower than the true difference.

Corollary: This "interval" still includes zero. So, there is no evidence for μ_1 being greater than μ_2.

Recall that you can compute a lower confidence bound for each of μ_1 and μ_2 separately:

```
t.test(pa.boy, alternative = "greater")$conf.int[1:2]

[1] 81.24   Inf

t.test(pa.girl, alternative = "greater")$conf.int[1:2]

[1] 82.67   Inf
```

Example

We will compare the grades on a statistics midterm of those who pick up their tests within the first one or two weeks after the test to those who do not pick it up in that period of time. We use this as a proxy for attendance. The following analysis is conducted to see if there is a statistically significant difference between the means of the two groups.

```
attend <- c(9.0, 14.0, 15.0, 12.5, 13.5, 14.5, 12.5, 8.5, 17.5, 9.5, 12.0, 11.0,
            14.0, 14.5, 14.0, 21.5, 12.5, 10.5, 17.5, 5.0, 10.5, 17.5, 16.5, 19.0,
            18.0, 15.5, 13.5, 21.5, 10.5, 17.0, 18.5, 12.0, 15.0, 17.5, 11.5,
            15.5, 17.0, 17.0, 20.0, 15.5, 12.0, 13.0, 23.0, 11.5, 14.0, 13.0, 22.5,
            8.5, 11.0, 9.5, 11.5, 17.0, 11.5, 17.5, 7.5, 8.0, 14.5, 9.5, 19.0,
            16.5, 18.5, 10.5, 16.5, 14.5, 13.5, 14.5, 12.0, 17.0, 13.0, 11.0, 12.5,
            9.0, 19.0, 15.0, 16.0, 11.0, 7.0, 22.0, 13.0, 7.5, 14.5, 13.0, 18.5,
            13.0, 18.5, 10.0, 20.5, 10.5, 17.5, 13.0, 19.5, 10.0, 13.0, 19.5, 10.5,
            14.5, 11.0, 14.5, 7.0, 7.0, 9.0, 16.0, 13.0, 19.5, 15.0, 17.0, 18.0,
            10.5, 15.0, 8.5, 10.0, 14.0, 16.0, 12.5, 13.5, 17.0)
non.attend <- c(3.0, 12.5, 8.5, 18.5, 5.5, 18.5, 7.5, 13.5, 6.5, 17.0, 11.5, 13.0,
                13.0)
```

To see if the data provide evidence for the claim that $\mu_1 =$ mean of attend is higher than $\mu_2 =$ mean of non.attend, the appropriate CI is a lower confidence bound for $\mu_1 - \mu_2$, which is equivalent to testing

$$H_0 : \mu_1 - \mu_2 \leq 0$$
$$H_1 : \mu_1 - \mu_2 > 0$$

```
t.test(attend, non.attend, alternative = "greater", conf.level = 0.95)


Welch Two Sample t-test

data:  attend and non.attend
```

```
t = 1.8, df = 14, p-value = 0.05
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 0.04549     Inf
sample estimates:
mean of x mean of y
    13.98     11.42
```

This means that we can be 95% confident that the true (i.e. population) mean grade of the attending students is higher than that of the non-attending students by at least 0.045. Because 0 is not included in the CI, the "corollary" conclusion is that the mean grade of attending students is higher than that of the non-attending students. One often says that the difference is "statistically significant." (The same conclusion follows from the p-value; it's smaller than $\alpha = 0.05$, and so we can reject $H_0 : \mu_1 \leq \mu_2$ in favor of $H_1 : \mu_1 > \mu_2$.)

The result is statistically significant, but is it physically significant? That's a different question! In other words, how much higher is the mean of the attendees, and do we care? To answer that, look at the sample means of the two groups (last line of the output). The attending students' grade is $\frac{13.98-11.42}{11.42} \cdot 100 \approx 22\%$ higher than that of the non-attending students. That's big enough to be considered physically significant.

It's important to note that **statistical significance** and **physical significance** are two different concepts. The difference between the two means may be statistically significant, but it may be so small that no one really cares about it, i.e., it may be physically non-significant.

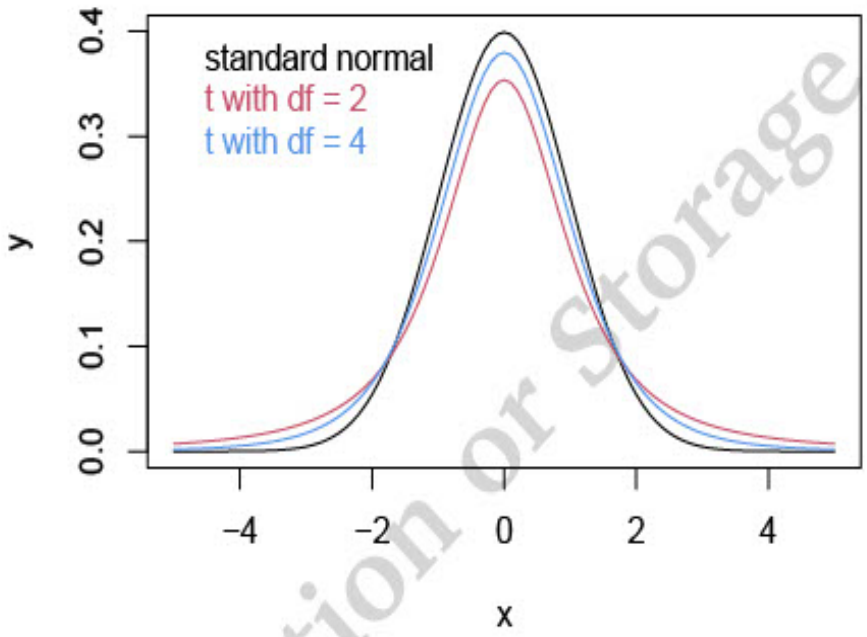
5.5 The t-distribution

All confidence intervals require knowing areas under distributions in order to get the correct z^* and t^* in the CI formulas. Table 1 in the book gives areas under the standard normal to the **left** of any number. Table 6 in the book gives areas under the t-distribution to the **right** of any number. Note that z^* and t^* themselves are NOT given in these tables. z and t (not starred) are what we compute in the p-value approach.

In R, the analogs of pnorm(), qnorm(), and dnorm(), for the t-distribution are pt(), qt(), and dt(). For example,

```
pnorm(1.645, 0, 1, lower.tail = T)  # Area left of 1.645 under standard normal.

[1] 0.95

pt(1.645, df = 5, lower.tail = F)  # Area right of 1.645 under t with df=5.

[1] 0.08044

# The quantiles of the normal and t distributions are obtained in the following
# way:
qnorm(0.05, 0,1, lower.tail = T)  # z which has 0.05 area to its left.

[1] -1.645

qt(0.05, df = 5, lower.tail = T)  # t which has 0.05 area to its left.

[1] -2.015

# Finally, you can see what the two distributions look like:
x <- seq(-5, 5, .1)  # x going from -5 to +5 in 0.1 steps.
```

```r
y_1 <- dnorm(x, 0, 1)  # Standard normal density.
y_2 <- dt(x, 2)  # t density with df = 2.
y_3 <- dt(x, 5)  # t density with df = 5.
plot(x, y_1, type = "l", ylab = 'y')
lines(x, y_2, col = 2)
lines(x, y_3, col = 4)
legend('topleft', c('standard normal', 't with df = 2', 't with df = 4'),
       text.col = c(1, 2, 4), bty = 'n')
```



5.6 Confidence Interval When σ is Unknown (Small Sample)

We know that the sampling distribution of the sample mean is the normal distribution with parameters μ_x and $\sigma_x/\sqrt{n}$. And so, $\frac{\bar{x}-\mu_x}{\sigma_x/\sqrt{n}}$ has a standard Normal distribution. However, if σ_x is unknown, then it has to be approximated with the sample standard deviation s; which is fine, if the sample size is large. However, for small samples, the approximation is poor, and so the sampling distribution of $\frac{\bar{x}-\mu_x}{s/\sqrt{n}}$ does not follow the standard normal, but rather the t-distribution with $n-1$ degrees of freedom where n is the sample size. To find the confidence interval, all we need to know is how to compute areas under the t-distribution between two numbers, just like we what did with the normal distribution. In R, we can replace **qnorm(.05 / 2)** with **qt(0.05 / 2, sample.size - 1)**. The results will be very similar if the sample size is large since the t-distribution converges to normal as the sample size $n \to \infty$. But for small samples (e.g., 20), the confidence interval calculated using a t-distribution will cover the population mean the correct number of times (if the population is normal), while the normal confidence interval will not. For small samples taken from non-normal populations, we do not have any formulas. We should use the bootstrap method instead; see below.

5.7 Bootstrap: CI without formulas

We have confirmed in 4.2 that the CI computed with the formula $\bar{x} \pm z^* \cdot \frac{\sigma}{\sqrt{n}}$ has the correct coverage property (about 95% of such CIs cover the true mean). But that conclusion is based on several assumptions:

1. We know what z^* to use in the formula.
2. We can approximate σ with the sample standard deviation.

But for some statistics (e.g., sample mean) we don't even have a formula for a CI. One solution to that problem is Bootstrapping.

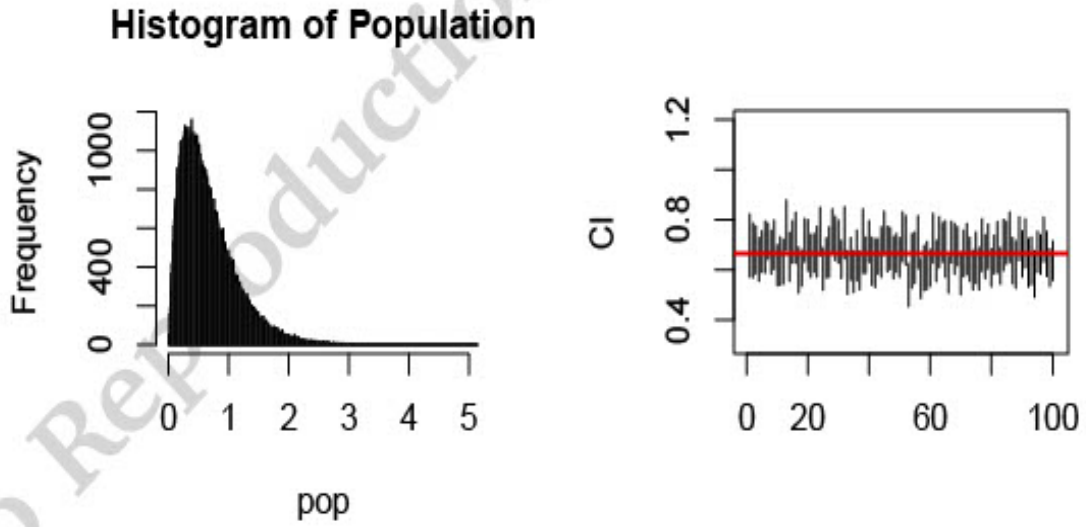
Example: Producing the Correct CI for Mean

Instead, of using the formula $\frac{sd(sample)}{\sqrt{n}}$ for the standard deviation of the sampling distribution of the sample mean, we can actually build (though approximately) the sampling distribution itself. This is done by taking multiple samples - called bootstrap samples - from the single observed sample! The theory behind bootstrap argues that the std dev of this "sampling distribution" is a pretty good estimate of the standard dev of the sampling distribution of the sample mean. Armed with this approximation to the sampling distribution, we can take its appropriate quantiles to give us CI; after all, $\bar{x} \pm 1.96 \cdot \frac{\sigma}{\sqrt{n}}$ mark quantiles of the true sampling distribution). So, that's the idea: to build a histogram of the sample statistic of interest by treating the sample as if it were the population.

Now, when it comes to testing the coverage properties of a CI for some parameter, recall that we take multiple samples from a population already. So, in the bootstrap approach, we will have to take multiple (bootstrap) samples from each of the samples taken from the population. For technical reasons that we won't go into, the bootstrap samples must be taken with replacement.

```
rm(list = ls(all = TRUE))
set.seed(1)
N <- 100000
pop <- rgamma(N, 2, 3)  # Draw from gamma instead of normal.
pop.mean <- mean(pop)
pop.sd <- sd(pop)
pop.median <- median(pop)
c(pop.mean, pop.sd, pop.median)

[1] 0.6659 0.4705 0.5590

hist(pop, breaks = 400, main = 'Histogram of Population')

n.trial <- 100
sample.size <- 90
CI <- matrix(nrow = n.trial, ncol = 2)
for (i in 1:n.trial) {
  sample.trial <- sample(pop, sample.size)  # Take a sample.
  # Now, the bootstrap block (which you have to type in):
  # For each sample, take a bootstrap sample (with replacement), and compute
  # the sampling distribution of the sample means. The appropriate quantiles
  # of this sampling distribution give the confidence interval.
  n.boot <- 100  # Number of bootstrap samples, from each sample.
  boot.stat <- numeric(n.boot)
    for (j in 1:n.boot) {
```

```
      boot.sample <- sample(sample.trial, sample.size, replace = T)
      # With replacement.
      boot.stat[j] <- mean(boot.sample)  # Store the means.
      }  # End of loop over bootstrap.

  CI[i, ] <- quantile(boot.stat, c(0.05 / 2, (1 - 0.05 / 2)))
  # CI[i,] <- c(mean(sample.trial) + qnorm(.05/2) * pop.sd / sqrt(sample.size),
  #             mean(sample.trial) - qnorm(.05/2)*pop.sd / sqrt(sample.size))
  # For small sample, replace qnorm(.05/2) with qt(0.05/2, sample.size - 1).
  # CI[i, ] <- sort(boot.stat)[(n.boot + 1) * c(0.05/2, 0.95/2)]  # See Geyer.
}  # End of loop over samples.

count <- 0
for (i in 1:n.trial) {
  if (CI[i, 1] <= pop.mean && CI[i, 2] >= pop.mean)
    count <- count + 1
    }
count

[1] 95

plot(c(1, 1), CI[1, ], ylim = c(0.3, 1.2), xlim = c(0, 101), ylab="CI", xlab = '',
     type = "l")
for (i in 2:n.trial) {
  lines(c(i, i), CI[i, ])  # Draw CIs (vertically).
}
abline(h = pop.mean, col = "red", lwd = 2)  # Draw the population mean
                                            # (horizontally).
```

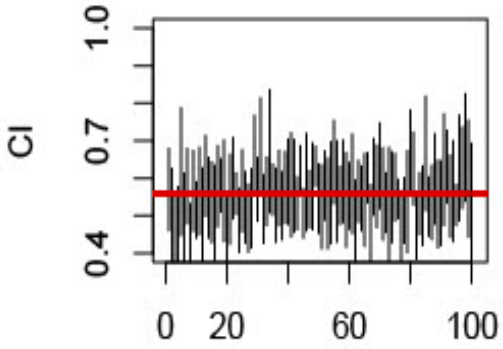


It may seem like the bootstrap method makes no assumptions, and that it will work all the time. However, it turns out that it does have some problems. Some of the problems are addressed by Schenker (1985). For example, he shows that the particular version we use above (called percentile bootstrap) gives CIs which cover the population parameter less frequently than they should, especially for small samples. For example, with a sample size of 20, a 90% CI will cover the pop mean around

78% of the time.

5.7.1 Confidence Interval for Sample Median

```
n.trial <- 100
sample.size <- 90
CI <- matrix(0, n.trial, 2)
for (i in 1:n.trial) {
  sample.trial <- sample(pop,sample.size,replace=F)
  n.boot <- 100
  boot.stat <- numeric(n.boot)
  for (j in 1:n.boot) {
    boot.sample <- sample(sample.trial, sample.size, replace = T)
    boot.stat[j] <- median(boot.sample)  # Median
    }
  CI[i, ] <- quantile(boot.stat, c(0.05 / 2, (1 - 0.05 / 2)))
}

count <- 0
for (i in 1:n.trial) {
  if (CI[i, 1] <= pop.median && CI[i, 2] >= pop.median)
    count <- count + 1
}
count

[1] 96

plot(c(1, 1), CI[1, ], ylim = c(0.4, 1), xlim = c(0, 101), xlab = '', ylab = "CI",
     type = "l")
for (i in 2:n.trial) {
  lines(c(i, i), CI[i, ])
  abline(h = pop.median, col = "red", lwd = 2)
}
```



Note that the number of times that the confidence interval covers the true median is close to 95. In other words, the way we are computing a confidence interval for a population median gives us confidence intervals that cover the population median the expected number of times. In practice, when you have a **single** sample, and no population, you can use this bootstrap method to build a confidence interval for the population median.

A quick partial fix to the problem of under-coverage is proposed by Charles Geyer:

http://www.stat.umn.edu/geyer/old/5601/examp/percent.html

and it involves revising the CI line just a bit. The commented line in in the above code will let you test this idea.

Reference

Schenker, Nathaniel (1985): Qualms About Bootstrap Confidence Intervals Journal of the American Statistical Association, Vol. 80, No. 390 (Jun., 1985), pp. 360-361.