

# Inference for One Mean (using SE)

Stat 145

## SE for a mean

The standard error for a sample mean is

$$SE = \frac{\sigma}{\sqrt{n}}$$

and  $\bar{X} \approx \text{Norm}(\mu, \frac{\sigma}{\sqrt{n}})$  as long as the sample size is large enough.

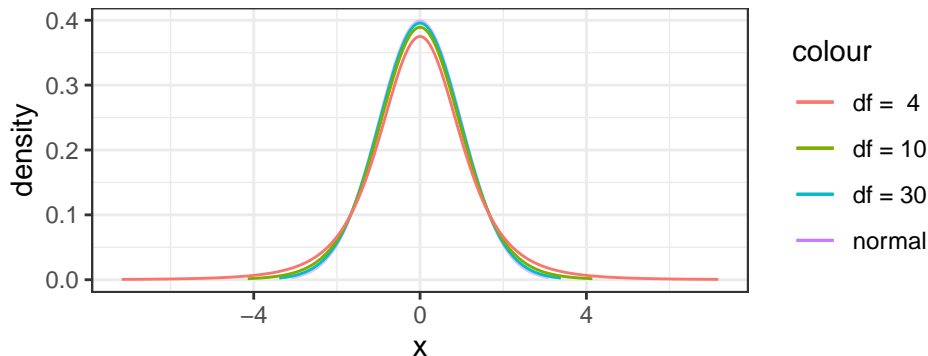
We can use these facts to compute confidence intervals and p-values for a mean  $\mu$ . But there is one problem: We don't know  $\sigma$ , so we have to estimate it by taking the standard deviation of our sample ( $s$ ). This adds some additional uncertainty (not only do we not know the population mean  $\mu$ , we also don't know the population standard deviation  $\sigma$ ). We will use the estimate

$$SE \approx \frac{s}{\sqrt{n}}$$

But when do this the distribution is no longer normal. It is slightly "shorter and more spread out", so we will replace the standard normal distribution with a new distribution, called a **t distribution**.

Actually, there are many t-distributions, and the one we use will depend on the sample size. If we have  $n$  observations in our data set, we will use the **t distribution with  $n - 1$  degrees of freedom**. The larger the degrees of freedom, the more the t distribution will resemble a normal distribution.

```
gf_dist("norm", color = ~ "normal", alpha = 0.5) %>%  
  gf_dist("t", df = 30, color = ~ "df = 30") %>%  
  gf_dist("t", df = 10, color = ~ "df = 10") %>%  
  gf_dist("t", df = 4, color = ~ "df = 4")
```



With the t-distribution in place, we can easily compute confidence intervals and p-values.

## Rule of Thumb

The t-distribution approximation is better when

- the sample size is larger
- the *population* is more like a normal distribution

Our rule of thumb breaks down into advice regarding small, medium, and large sample sizes:

- small (under 10): the t-approximation is fine if the population is normal or very nearly normal. This is hard to assess from the data (since we have so little), so there must be some other reason to believe that the population is approximately normal.
- medium (10–30): the t-approximation is fine as long as the population is unimodal and not heavily skewed and the sample has no clear outliers.
- large (30 or more): the t-approximation is fine for most unimodal populations. The main thing to watch for here is an extreme outlier in our sample since such an outlier can have a big impact on both the sample mean and the sample standard deviation.

Again, there are no “bright lines” here. As the sample size gets larger, it is less and less important how much the population looks like a normal distribution.

## Confidence intervals

$$\text{estimate} \pm \text{critical value} \cdot SE$$

becomes

$$\bar{x} \pm t_* SE$$

### Example: Newts

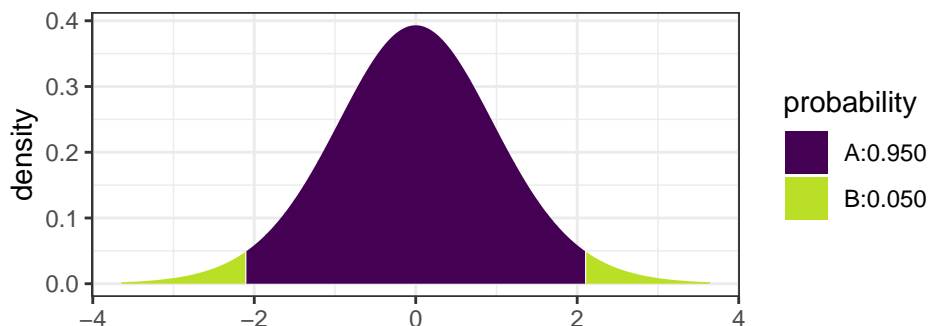
Biologists studied the rate at which new cells closed a razor cut made in the skin of an anesthetized newt. They quantified the rate in micrometers per hour ( $\mu\text{m}/\text{h}$ ). For a sample of 18 newts, they found  $\bar{x} = 25.67$  and  $s = 8.324$ . Use this information to compute a 95% confidence interval.

Here’s how we get R to do the work for us.<sup>1</sup>

```
t.star <- qt(.975, df = 17); t.star
```

```
## [1] 2.109816
```

```
t.star <- xct(.95, df = 17); t.star # c for "center"
```



```
## [1] -2.109816 2.109816
```

<sup>1</sup>`xct()` and `xcnorm()` from the `mosaic` package are handy, but the usual functions in R only work with probabilities *below*. It’s your choice whether you want to use `xct()` and `xcnorm()`, but know that `qt()` and `qnorm()` are the standard ways of doing this.

```
SE = 8.324 / sqrt(18); SE
```

```
## [1] 1.961986
```

```
ME = t.star * SE; ME
```

```
## [1] -4.139428  4.139428
```

```
# confidence interval  
25.67 + ME
```

```
## [1] 21.53057 29.80943
```

That's too many digits to report. Since the margin of error is 4.1, it makes sense to report this as (21.5, 29.8) or  $25.7 \pm 4.1$ . Digits beyond that are not known with any certainty anyway. (As a general rule, standard errors and generally only reported to 1 or 2 significant digits. But be sure to do your rounding at the very end of the computation, not along the way.)

With a sample of size 18, the confidence interval should be reasonably accurate as long as the population distribution is not bimodal or heavily skewed. A plot of the data could be used to make sure there is no evidence of these sorts of problems.

## P-values

For p-values we use the test statistic

$$t = \frac{\bar{x} - \mu_0}{SE}$$

and compare the result to the t-distribution using `pt()` (or of `xpt()`). As always, make sure you subtract from 1 if you need an upper tail and double if you are doing a 2-sided test.

### Example: Sweetness in Colas

Cola manufacturers are concerned with changes in flavor while their beverages are stored. In one test, 10 samples were tasted by trained taste testers before and after storage. The average sweetness lost was 1.02 units with a standard deviation of 1.20 units. Is this enough evidence to conclude that sweetness is being lost during storage?

Notice that this is a paired design: We have two measurements (sweetness before and after storage) and we are considering the difference in these two values. The summarized data only mentions the differences, but the raw data would include both measurements.

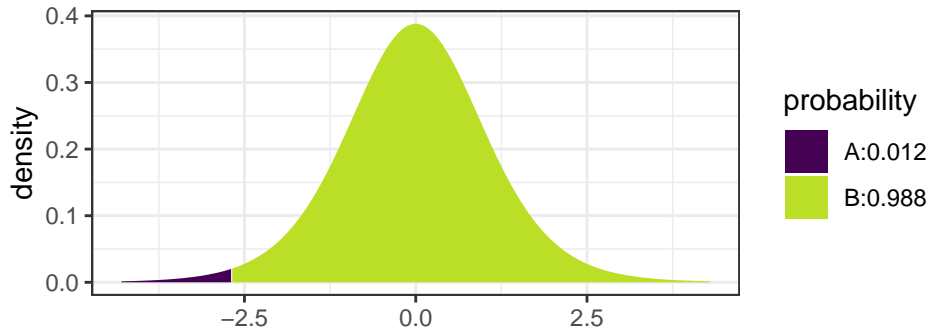
```
SE <- 1.20 / sqrt(10); SE
```

```
## [1] 0.3794733
```

```
t <- (-1.02 - 0) / SE; t
```

```
## [1] -2.687936
```

```
# p-value  
xpt(t, df = 9)
```



```
## [1] 0.01244025
```

This data does provide evidence of loss of sweetness since the p-value is quite small. Note: with such a small sample size, we must have *a priori* reasons to believe that the population is approximately normally distributed to believe this p-value is accurate.

### Your turn

Compute the critical value assuming you want to construct a confidence interval for a mean in each of these situations:

1.  $n = 15$ , confidence level = 95%
2.  $n = 25$ , confidence level = 90%
3.  $n = 10$ , confidence level = 98%

In each example below, compute the p-value or confidence interval using the standard error and t-distribution

4. Use the `RestaurantTips` data set from the `Lock5withR` package to give a 90% confidence interval for the mean tip.

```
library(Lock5withR)
df_stats(~ Tip, data = RestaurantTips)
```

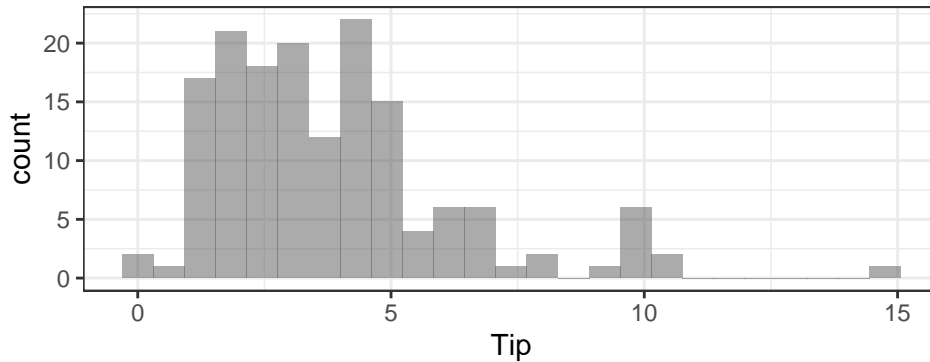
```
##   response min  Q1 median Q3 max   mean      sd  n missing
## 1      Tip 0.25 2.1   3.35  5  15 3.849299 2.420857 157      0
```

5. The percent tipped is perhaps more interesting than the absolute amount. Compute a 95% confidence interval for the mean tip percent.
6. Is there evidence that mean tip percent is less than 18% (a common tip guideline)? First answer this question by looking at your confidence interval, then compute a p-value.
7. Compute a 98% confidence interval for mean resting pulse using the data in `BodyTemp50`.
8. Why are confidence intervals usually more interesting than p-values in the context of a single mean?

```
t.test()
```

When we have the raw data, we can get R to do all the work for us using `t.test()`:

```
gf_histogram(~Tip, data = RestaurantTips)
```



```
mean(~Tip, data = RestaurantTips)
```

```
## [1] 3.849299
```

```
t.test(~Tip, data = RestaurantTips)
```

```
##  
## One Sample t-test  
##  
## data: Tip  
## t = 19.923, df = 156, p-value < 2.2e-16  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## 3.467663 4.230936  
## sample estimates:  
## mean of x  
## 3.849299
```

**What is the alternative?**

Note that the default hypothesis test is testing  $H_0 : \mu = 0$  and is two-sided. To change the hypothesis, use the `mu` argument.

```
t.test(~Tip, data = RestaurantTips, mu = 5)
```

```
##  
## One Sample t-test  
##  
## data: Tip  
## t = -5.9558, df = 156, p-value = 1.656e-08  
## alternative hypothesis: true mean is not equal to 5  
## 95 percent confidence interval:  
## 3.467663 4.230936  
## sample estimates:  
## mean of x  
## 3.849299
```

For a one-sided test, you can set `alternative = "less"` or `alternative = "greater"`.

```
t.test(~Tip, data = RestaurantTips, mu = 5, alternative = "less")
```

```
##  
## One Sample t-test  
##  
## data: Tip  
## t = -5.9558, df = 156, p-value = 8.278e-09  
## alternative hypothesis: true mean is less than 5  
## 95 percent confidence interval:  
##      -Inf 4.168992  
## sample estimates:  
## mean of x  
## 3.849299
```

### Changing the confidence level

We can change the confidence level for our interval using `conf.level`:

```
t.test(~Tip, data = RestaurantTips, conf.level = 0.98)
```

```
##  
## One Sample t-test  
##  
## data: Tip  
## t = 19.923, df = 156, p-value < 2.2e-16  
## alternative hypothesis: true mean is not equal to 0  
## 98 percent confidence interval:  
## 3.395172 4.303426  
## sample estimates:  
## mean of x  
## 3.849299
```

### `t.test()` is your friend

In practice, we will usually use `t.test()` when we have the data available to us, but you should know how to calculate all of the numbers that appear in the output of `t.test()`.

### Too much information?

If all you want is the confidence interval or the p-value, we can grab just that part using `confint()` or `pval()`:

```
t.test(~Tip, data = RestaurantTips, mu = 5) %>% confint()
```

```
## mean of x lower upper level  
## 1 3.849299 3.467663 4.230936 0.95
```

```
t.test(~Tip, data = RestaurantTips, mu = 5) %>% pval()
```

```
## p.value  
## 1.65553e-08
```

10. Use `t.test()` to check your answers above.

## Alternative methods: Bootstrap and randomization

In each of the situations above, we can also compute the confidence interval or p-value using bootstrap or randomization methods.

11. For each confidence interval above, compute a bootstrap confidence interval and compare it to the interval you computed using SE and the t-distribution.

### Randomization distribution = shifted bootstrap distribution

To compute a p-value using randomization, we need a way to simulate data where the null hypothesis is true. For the one-mean situation, we can do this by shifting the bootstrap distribution so that it is centered in the correct place. This will give us a distribution with the correct shape and center.

Here is an example:

```
Tips_boot <- do(5000) * mean(~Tip, data = resample(RestaurantTips))
mean(~mean, data = Tips_boot)
```

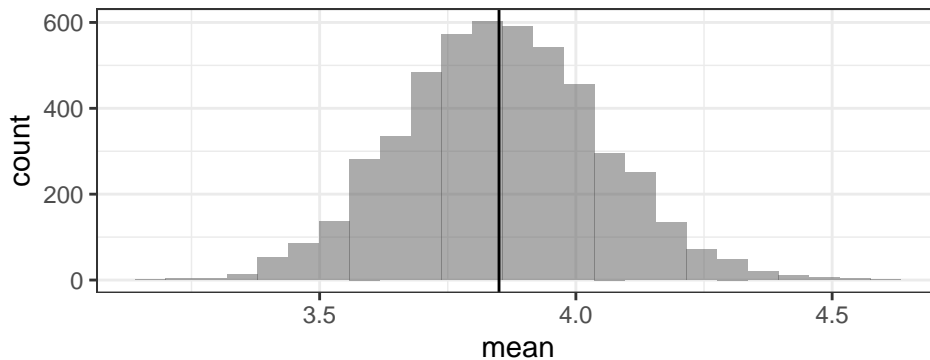
```
## [1] 3.853067
```

```
mean(~Tip, data = RestaurantTips)
```

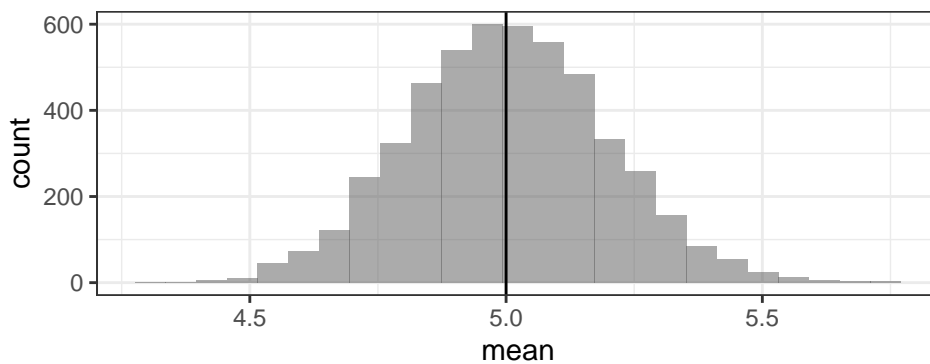
```
## [1] 3.849299
```

To test  $H_0 : \mu = 5$ , we can shift the distribution by  $5 - 3.85 = 1.15$ .

```
gf_histogram(~mean, data = Tips_boot) %>%
  gf_vline(xintercept = ~ 3.85)
```



```
Tips_rand <- Tips_boot %>% mutate(mean = mean - 3.85 + 5) # shift the mean
gf_histogram(~mean, data = Tips_rand) %>%
  gf_vline(xintercept = ~ 5)
```



12. For each hypothesis test above, redo it using a randomization distribution.