UNIT 10 · BLOCK 3

1 One-Sample t-Test

Comparing one group's mean to a single known number — without knowing σ.

Why this matters

Last class, the *z*-test let us ask "is this group different from a benchmark?" — but it had a catch: it needed the **population** standard deviation (σ), a number we almost never actually have. The *t*-test fixes that. It uses the standard deviation we *can* compute, the one from our own sample. That one small swap is what makes this the test you'll reach for again and again — in your senior thesis, in published papers, on the exam. Master the one-sample *t* and the independent-samples and paired *t*-tests coming up are just variations on the same theme.

By the end you can…

- Recognize **when** a one-sample *t* applies: a sample mean vs. a known/hypothesized μ_0, with σ unknown.
- Compute $\boldsymbol{t = (\bar{x} - \mu_0) / (s/\sqrt{n})}$ and explain every piece — including the estimated standard error.
- Describe the ***t*-distribution**, why its tails are fatter than z's, and what $\boldsymbol{df = n - 1}$ does to it.
- Reach a decision by comparing *t* to a **critical value** or reading a ***p*-value** — and report an **effect size** (Cohen's *d*).

1. The big idea

A **one-sample *t*-test** compares **one group's mean** to a **single known or hypothesized number** — we call that number **μ_0** ("mu-naught," the value the null hypothesis claims the population mean equals). Same question as the *z*-test — *is our group different from the standard?* — but now we don't need to know the population spread. We use the **sample** standard deviation (**s**) instead, which is the one number our data actually hands us.

When should this be your test? Three things have to line up: you have **one sample** of quantitative scores; there's a **fixed comparison value** (μ_0) sitting out there — a spec, a published norm, a recommended amount; and you **do not know σ**, the population standard deviation. If a friend describes a study where one group's average is checked against a single agreed-upon number and nobody handed them the population SD, you're hearing a one-sample *t*.

Symbol	Plain English	Where it comes from
$\bar{x}$ (x-bar)	Sample mean	We **compute** it from our data
μ_0 (mu-naught)	The benchmark / hypothesized population mean	A known number we test against
σ (sigma)	Population standard deviation	We almost **never** know this
s	Sample standard deviation	We **compute** it from our data
df	Degrees of freedom	$df = n - 1$

CHECK **A coffee shop weighs 30 espresso shots and asks whether their average weight differs from the 18 g they advertise. They don't know the population SD of all shots ever pulled. Which test fits?**

One-sample *t*-test. One sample of numbers, one fixed benchmark (μ_0 = 18 g), and σ is unknown — so we estimate the spread with the sample's own s. That's exactly the one-sample *t* setup.

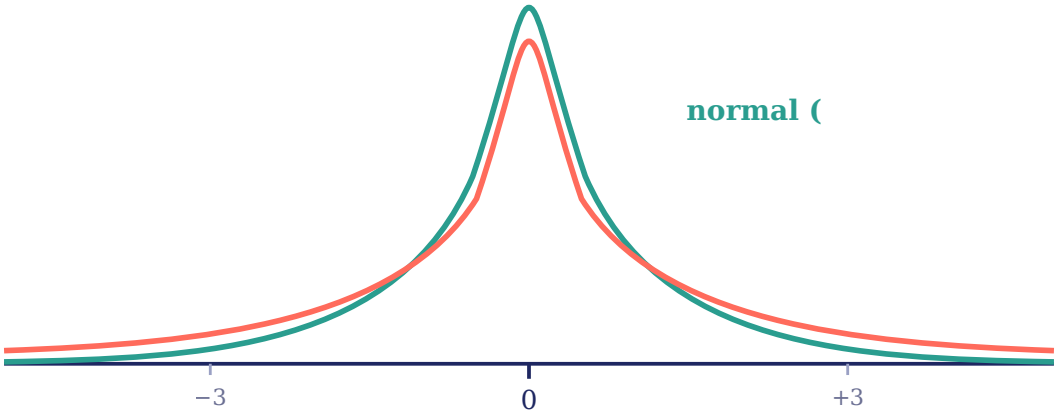
2. The catch with the *z*-test

The *z*-test formula needed σ, the spread of the *whole population*. But think about it — if we already knew that much about the population, why would we be testing a sample at all? In real research, **σ is a mystery.** So the *z*-test, as nice as it is, mostly can't be used. The fix is simple: swap in **s**, the standard deviation of our sample — the exact number Sheets hands us from our data. Now everything is computable.

But that swap isn't free. The sample SD (s) is itself only an *estimate* of σ, and a small sample can estimate it badly. Using an estimate where we used to have a known value adds a little wobble to the whole calculation. We can't just pretend that extra uncertainty isn't there — so statistics builds it in, with a slightly different reference curve. That curve is the ***t*-distribution**.

3. The *t*-distribution (the bell with fatter tails)

There's one price for estimating σ with s: a little extra uncertainty. The ***t*-distribution** builds that caution right in. It looks like the normal bell — same center at zero, same general hump — but with slightly **fatter tails** and a marginally shorter peak. Fatter tails just mean "extreme values are a bit more likely than the normal curve would say, so be a touch more cautious before calling something rare."



z) *t* (fatter tails)
The *t*-distribution (coral) sits a hair lower in the middle and carries more weight in the tails than the normal curve (teal). That extra tail weight is the "honest caution" you pay for estimating σ with s. As the sample grows, the coral curve creeps up toward the teal one until they overlap.

Because the tails are fatter, the cutoff for "rare enough to be surprising" sits a little farther from zero than it does for *z*. With *z* you needed to clear about ±1.96 for a two-tailed test at the .05 level; with *t* you have to clear a slightly *larger* number — and exactly how much larger depends on your sample size, through the degrees of freedom.

4. Degrees of freedom — why *df* = *n* − 1

How fat the tails are depends on the **degrees of freedom (*df* = *n* − 1)**. Small sample fatter tails more caution. As your sample grows, the tails thin out and ***t* looks just like *z***. With our 200 students, *t* and *z* are nearly identical — but *t* is always the honest choice when you used s.

Why ***n* − 1** and not *n*? Computing s uses the sample mean (x̄) you already calculated from the same data. Once the mean is fixed, the deviations from it are not all free to vary: if you know the mean and you know all but one of the scores, that last score is *determined* — it has to be whatever value makes the mean come out right. So only ***n* −**

1 of the deviations are genuinely "free." That lost piece of freedom is exactly what $df = n - 1$ is counting, and it's why s divides by $n - 1$ back in Unit 4.

What degrees of freedom do

Think of *df* as a dial for caution. Tiny *df* (say 4) noticeably fatter tails the critical *t* is large. Big *df* (say 199) the tails have thinned almost to the normal curve the critical *t* is barely above 1.96. More data, less penalty. The *t*-distribution isn't one curve — it's a *family*, one curve per *df*.

CHECK **You run a one-sample *t*-test on a sample of *n* = 25 people. What are the degrees of freedom?**

24. *df* = *n* − 1 = 25 − 1 = 24. You lose one degree of freedom because the sample mean was already used up in computing s. With *df* = 24 the critical *t* for a two-tailed test at α = .05 is about 2.06 — a bit bigger than z's 1.96.

5. The formula

```
t = (x̄ − μ₀) / (s / √n)
```

In words: take how far your sample mean ($\bar{x}$) sits from the benchmark (μ_0), then divide by the **estimated standard error**, `s / √n`. It's the z-formula with **s** in place of σ — that's the whole change. Let's name the two parts, because every *t*-test you'll ever run has this same shape:

- **Top — the signal:** `x̄ − μ₀` is the raw gap between your sample's mean and the benchmark, in the original units (hours, grams, points). Bigger gap bigger *t*.

- **Bottom — the noise:** `s / √n` is the **standard error of the mean** — how much sample means typically bounce around just by chance. Notice the **√n**: more data shrinks the standard error, so the same gap produces a larger *t* with a bigger sample.

So *t* is a **signal-to-noise ratio**: how big is the difference, measured against how much wobble we'd expect by luck alone? A *t* near 0 means the mean is right about where μ_0 predicts; a *t* far from 0 (in either direction) means the gap is large relative to the noise.

> **Standard error ≠ standard deviation**
>
> **s** describes how spread out the individual scores are. **s/√n** (the standard error) describes how spread out the *sample means* would be if you repeated the study many times. The standard error is always the smaller of the two, and it shrinks as *n* grows — that's the engine behind every significance test.

6. Reaching a decision — critical value or *p*-value

Once you have your *t*, there are two equivalent ways to decide whether it's surprising enough to reject the null hypothesis (H_0: the population mean equals μ_0). They always agree — pick whichever your software or table makes easy.

(a) Compare *t* to a critical value. Look up the **critical *t*** for your *df* and significance level (α, usually .05, two-tailed). That cutoff marks where the fat tails begin. If your calculated |t| is *bigger* than the critical value, your result lands out in the tail — too rare to chalk up to chance — so you **reject H_0**. If it's smaller, you **fail to reject H_0**.

(b) Read the *p*-value. The ***p*-value** is the probability of getting a *t* at least this far from zero *if H_0 were true*. Small *p* (conventionally < .05) = "this would be a weird fluke if nothing were going on" reject H_0. Larger *p* fail to reject. The two-tailed *p* counts both tails, since the mean could be off in either direction.

Say it correctly

When *p* is not small enough, we **"fail to reject"** H_0 — we never "~~accept~~" it or "prove there's no difference." Absence of evidence isn't evidence of absence; maybe the effect is real but our sample was too small to catch it. This wording isn't fussiness — it's the honest description of what the test can and can't show.

CHECK **Your one-sample *t*-test returns *p* = .21. What's the right conclusion?**

Fail to reject H_0. *p* = .21 is well above .05, so the result is consistent with chance and we don't have enough evidence to reject the null. We never "accept" H_0 or claim we've proven no difference — we just haven't found grounds to reject it.

Worked example — compute a one-sample *t*

Question: A wellness coach claims her clients average **7 hours** of sleep. You sample ***n* = 25** of them and find a mean of $\bar{x}$ **= 6.2** hours with **s = 1.0**. Is their true average different from 7? Test at α = .05, two-tailed. Predict each step before revealing it.

Worked example — one-sample *t*, step by step

$\bar{x} = 6.2$, $\mu_0 = 7$, $s = 1.0$, $n = 25$. We'll build *t*, find *df*, and compare to the critical value.

STEP 1 — THE STANDARD ERROR (THE "NOISE")

SE = s / √*n* = 1.0 / √25 = 1.0 / 5 = `0.20`. This is how much sample means of 25 people typically wobble by chance.

STEP 2 — THE GAP (THE "SIGNAL")

$\bar{x} - \mu_0 = 6.2 - 7 =$ `−0.8` hours. The sample sleeps eight-tenths of an hour *below* the claimed 7.

STEP 3 — DIVIDE SIGNAL BY NOISE

t = ($\bar{x} - \mu_0$) / SE = −0.8 / 0.20 = `−4.0`. The sample mean sits 4 standard errors below the benchmark.

STEP 4 — DEGREES OF FREEDOM

df = *n* − 1 = 25 − 1 = `24`. This tells us which t-curve to compare against.

STEP 5 — COMPARE TO THE CRITICAL VALUE

The critical *t* for *df* = 24, two-tailed α = .05, is about `2.06`. Our |t| = 4.0 is far bigger than 2.06.

STEP 6 — DECIDE

|t| = 4.0 > 2.06, so we `reject H₀` (*p* < .001). The clients' true mean sleep **is** significantly different from — specifically below — 7 hours.

Notice the rhythm: **standard error gap *t* *df* compare decide**. Every one-sample *t*-test runs through those same six beats. The negative

sign on *t* just means the sample fell *below* μ_0; the *size* (4.0) is what made it significant.

A second worked example — on OUR data

Question: Do SJNY students sleep **less** than the recommended **8 hours**? We'll test `sleep_hours` (column *F*) against the benchmark μ_0 = **8**. The numbers from our 200 students:

- sleep mean $\bar{x}$ = **6.82**
- s = **1.02**
- *n* = **200**, so *df* = **199**

Plug in:

```
t = (6.82 − 8) / (1.02 / √200) = −1.18 / 0.0721 = −16.48
```

The verdict

With ***df* = 199**, the critical *t* (two-tailed, α = .05) is only about **1.97**, and |t| = 16.48 blows past it — that gives ***p* < .001**. The mean sits a huge distance below 8. **Yes — SJNY students sleep significantly less than 8 hours.**

Try it yourself

Here's the live one-sample *t*-test. Paste your own numbers and a benchmark (μ), or hit **Load class data** to run the sleep-vs-8-hours example above and watch the *t* and *p* drop out.

7. Assumptions — when the *t*-test is trustworthy

The one-sample *t* leans on a couple of conditions. They don't have to hold perfectly — the test is fairly forgiving — but knowing them tells you when to be careful.

- **Independent observations.** Each score comes from a separate, unrelated case. Measuring the same person twice, or sampling clusters of friends who influence each other, breaks this — and it's the assumption that matters most. No formula can rescue a non-independent sample.
- **Roughly normal (or a big enough *n*).** The scores should come from a population that's approximately normal — symmetric, single hump, no wild outliers. The good news: thanks to the **Central Limit Theorem**, with a decent sample (often $n \geq 30$) the sampling distribution of the mean is close to normal *even when the raw scores aren't*, so the *t*-test still behaves. With small samples and obvious skew or outliers, be cautious.
- **Quantitative, interval/ratio data.** A mean has to make sense — so the outcome is a number you can average (hours, grams, test points), not a category.

> **The fragile one**
>
> Mild non-normality is usually fine, especially with larger *n*. **Non-independence is not.** If your observations are linked — repeated measures, matched pairs, clustered sampling — a one-sample *t* can be badly misleading, and you want a different design (often the paired *t*-test in Unit 12).

8. Effect size — Cohen's *d*

A tiny *p*-value tells you the difference is **real**, not how **big** it is. With 200 students, even a trivial gap turns up "significant" because the huge *n* shrinks the standard error to almost nothing. So we also report

Cohen's *d* — the gap measured in standard deviations, which doesn't care how big your sample is:

```
d = (x̄ − μ₀) / s = (6.82 − 8) / 1.02 = −1.16
```

> **Read it plainly**
>
> SJNY students sleep about **1.16 standard deviations** below the 8-hour mark. By the usual rule of thumb (0.2 small, 0.5 medium, 0.8 large), **1.16 is a large effect.** The sleep gap isn't just real — it's big. Notice the difference from *t*: Cohen's *d* divides by **s** (raw spread), while *t* divides by **s/√n** (standard error). That's why a giant sample can make *t* huge while *d* stays modest.

9. Confidence intervals — the estimate, not just the verdict

A hypothesis test hands you a verdict: different from 8 hours, or not. A **confidence interval (CI)** answers the better question hiding underneath: *so what IS the true mean, roughly?* Instead of one number, you report a **range** that very likely contains the population mean. The recipe recycles pieces you already have:

```
95% CI = x̄ ± t_critical × SE

sleep:  6.82 ± 1.97 × 0.0721    [6.68, 6.96]
```

Read it aloud: "*we're 95% confident the true average sleep of SJNY students lands between 6.68 and 6.96 hours.*" The multiplier is the same critical *t* from §6 (1.97 for *df* = 199), and the SE is the same noise term from the test — nothing new was computed, the ingredients were just reassembled from "verdict" into "estimate."

- **Width = precision.** A narrow CI (ours spans barely a quarter-hour) means the estimate is tight. Bigger *n* smaller SE narrower interval. Small samples produce embarrassingly wide ones.

- **The CI and the test always agree.** If the benchmark (8 hours) falls *outside* the 95% CI, the two-tailed test at α = .05 rejects — guaranteed. 8 is nowhere near [6.68, 6.96], which is why |t| = 16.48 was enormous. One picture, same verdict, plus the estimate for free.
- **What 95% means (and doesn't).** The procedure is right 95% of the time: across endless repeated samples, 95% of the intervals built this way capture the true mean. It is *not* "a 95% chance the true mean is in this one interval" — the true mean isn't random; our interval is. Subtle, and a favorite exam trap.

See it happen

The **Confidence Interval simulation** draws sample after sample and stacks up the intervals — watch roughly 1 in 20 of the 95% CIs *miss* the true mean entirely. Two minutes in that sim makes the "95% of intervals capture it" idea stick better than any paragraph.

CHECK

A 95% CI for mean stress is [4.2, 5.8]. A researcher wants to test whether the true mean differs from the benchmark 6 (two-tailed, α = .05). What will the test say?

Reject. The 95% CI and the two-tailed test at α = .05 are two views of the same math: any benchmark *outside* the interval gets rejected, any benchmark *inside* it does not. Since 6 > 5.8, the test rejects — no extra computation needed.

Report it in APA style

When this result goes into a paper or lab report, it gets one standardized APA (7th ed.) sentence: the test, the descriptives, the

statistic with its *df*, the *p*, and the effect size. The template for a one-sample *t*:

```
t(df) = value, p = value, d = value
```

House rules: statistical letters (*t*, *p*, *M*, *SD*, *d*) are **italicized**; statistics get **two decimals**; *p*-values drop the **leading zero** (.032, not 0.032) and are never written "= .000" — below .001, write ***p* < .001**. Our sleep result, written out:

> **The write-up**
>
> "SJNY students slept significantly less (*M* = 6.82 hours, *SD* = 1.02) than the recommended 8 hours, *t*(199) = −16.48, *p* < .001, *d* = −1.16, 95% CI [6.68, 6.96]."

The CI rides along in square brackets — APA style encourages it, because it upgrades the sentence from verdict to estimate: not just "less than 8," but "between 6.68 and 6.96."

Notice the anatomy: the **direction** in plain words ("slept significantly less"), the **descriptives** that anchor the claim (*M*, *SD*), then the full statistical sentence with the effect size riding along. Every inferential test from here to the end of the course reports in exactly this rhythm.

Stat in the Wild — one group vs. a fixed standard

The one-sample *t*-test shows up anywhere there's a **known benchmark** to measure against: a factory checks whether its bolts hit the **5 cm spec**; a school asks whether its students' average IQ differs from the national **100**; a renter asks whether this city's rent differs from the **national average**; a clinic checks whether patients' average wait time beats the **15-minute** promise.

The pattern: one group, one fixed number everyone agrees on, and the question "*are we different from it?*" Whenever you have a benchmark sitting there — a spec, a standard, a published average, a recommended amount — this is your test.

One number worth testing against

The one-sample *t* needs a benchmark — and some benchmarks carry weight beyond the arithmetic. Take animal agriculture. Producing a kilogram of beef emits roughly **60 kg of CO_2-equivalent**, while a kilogram of peas emits about **1 kg** (Poore & Nemecek, 2018). Imagine sampling the carbon footprint of *n* students' typical dinners and asking whether the average differs from a **climate-budget benchmark** — say, the per-meal footprint we'd need to hit a 1.5 °C target. That's a one-sample *t*: one sample of numbers, one fixed standard, σ unknown.

Our class dataset lets us ask quieter versions of the same question. We could test whether students' average `animal_compassion` score differs from the scale's neutral midpoint, or whether weekly meat servings differ from a dietary-guideline target. In every case the logic is identical to the sleep example — **$\bar{x}$ versus μ_0, divided by the standard error** — but the benchmark turns a dry calculation into a question with stakes. A "single known number" can be a spec, a norm, or a line we've collectively decided matters.

In Google Sheets

You don't run a *t*-test by hand — Sheets does it. Open the **SJNY Student Life** dataset. The slickest way is the built-in function, which compares a range against a benchmark and hands back the *p*-value directly:

```
=T.TEST(F2:F201, ARRAYFORMULA(F2:F201*0 + 8), 2, 1)    two-tailed p for sleep vs 8
```

The arguments are `(your data, the benchmark repeated, tails, type)`: **2** = two-tailed, **1** = one-sample. (The little `ARRAYFORMULA(…*0 + 8)` trick just builds a same-length column of 8's to compare against — a one-sample *t* is mathematically a paired test against a constant.) If you'd rather see the machinery, build *t* from its three ingredients and then convert to *p*:

```
=(AVERAGE(F2:F201) - 8) / (STDEV.S(F2:F201) / SQRT(COUNT(F2:F201)))    t = −16.48
=T.DIST.2T(ABS(-16.48), 199)                                           p < .001
```

Here `=AVERAGE` gives x̄, `=STDEV.S` gives the sample s (the ".S" version divides by $n - 1$, which is what you want), and `=COUNT` gives *n*. `=T.DIST.2T` turns your *t* and *df* into the **two-tailed** *p*-value — the chance of a *t* this far from zero in *either* direction. And to find the **critical *t*** — the cutoff you'd compare |t| against — use:

```
=T.INV.2T(0.05, 199)    critical t ≈ 1.97   (for df = 24 it returns ≈ 2.06)
```

`=T.INV.2T(α, df)` is the inverse of `T.DIST.2T`: feed it your significance level and *df*, and it returns the boundary where the rejection region begins. If your |t| exceeds it, reject H_0.

Watch out for

z needs σ; *t* needs only s — don't know the population SD? Use *t* (you almost always will). The formula is $\boldsymbol{t = (\bar{x} - \mu_0) / (s/\sqrt{n})}$, and the denominator is the **standard error**, not the plain SD. Remember $\boldsymbol{df = n - 1}$, not *n*: with $n = 200$, $df = 199$. When *p* isn't small enough you **"fail to reject"** H_0 — never "accept" it or claim you've proven no difference. A small *p* means "real," not "big," so always pair it with Cohen's *d*. The benchmark must be a **single known number** — a spec, a standard, a published value — not another group's mean (that's the independent-samples *t* in Unit 11). And the sign just shows direction: $t = -16.48$ means *below* μ_0; the size is what matters.

Key terms

One-sample *t*-test · **Benchmark (μ_0)** · **Population SD (σ)** · **Sample SD (s)** · ***t*-distribution** · **Degrees of freedom ($df = n - 1$)** · **Standard error ($s/\sqrt{n}$)** · **Critical value** · ***p*-value** · **Fail to reject** · **Cohen's *d*** · **Independence** · **Confidence interval (95% CI)**

PRACTICE — NEVER GRADED

Check yourself

Eleven quick true/false items with instant explanations.

(Different from your Canvas quiz — real practice.)

Sigma says: The whole leap from z to t is one swap — *we don't know σ, so we use s.* That's it. Everything else (fatter tails, degrees of freedom, the honest extra caution) follows from estimating the spread instead of knowing it. When you used your own data's SD, you used t. Always.