

One and two sample t-tests

1 Confidence interval cheat sheet

Let's say you are taking a large sample of n observations from a population with mean μ and standard deviation σ .

Your sample mean, \bar{x} is going to be your estimate of the population mean. s is the standard deviation of your sample.

The standard error of the mean SEM , is related to s as follows:

$$SEM = \frac{s}{\sqrt{n}} \quad (1)$$

You're asked to give a confidence interval for the mean of your sample. Say you want a 95% confidence interval. This means that $\alpha = 0.05$ and $\alpha/2 = 0.025$. Since the sample is large, you can just use z-scores (which implies that a normal distribution of means is assumed) as opposed to t-scores (which are a bit larger than z-scores for small samples).

The confidence interval is given by:

$$[\bar{x} - z_{\alpha/2} * SEM, \bar{x} + z_{\alpha/2} * SEM] \quad (2)$$

If on the other hand your sample was fairly small (e.g. $n = 10$ or $n = 20$), you would want to get the t value instead of the z score:

$$[\bar{x} - t_{\alpha/2,df} * SEM, \bar{x} + t_{\alpha/2,df} * SEM] \quad (3)$$

Remember that $df = n - 1$.

This is how it would work in practice. Let's simulate sampling 87 normally distributed variables with mean 10 and standard deviation 2.

```
> sample87 = rnorm(87,10,2)
> sample87
 [1] 10.702794  7.803232 11.567596 12.197608 12.145459 12.110301  9.795155  8.587220
 [9]  8.490917  8.008483 11.452050 10.217464  8.278613  8.654241 11.953143  8.113491
[17]  8.723327 10.605430  9.139390 10.921014 10.426906  7.760826 11.515485  8.649291
[25]  8.446337 10.526330  9.023458  9.901821  7.889055  7.976825  8.926175 12.974170
[33] 10.451839  6.921623  9.222301 11.136571  7.434310  9.912896 14.221466 11.428891
[41]  8.137289  9.167073  5.068394 10.147317 10.527114  9.170393 10.390491  6.154173
[49]  6.974125 11.029793  9.988622  7.975544 10.547826  6.106702  7.677219  7.722798
[57]  6.001083  8.697960 10.176290 10.951141  6.619053  8.250223 12.113802 11.133227
[65] 10.094023 10.622176  9.460355  9.029852  8.543819 10.391742  9.414384  8.193490
[73]  8.921530 11.007032  5.903301  4.779386 10.207699  9.771383 13.044497  8.482020
[81]  7.699894 12.289890  5.353009  6.722348  6.334610 12.364507  6.601207
```

We calculate the mean and standard deviation, as well as the standard error of the mean.

```
> xbar = mean(sample87)
> s = sd(sample87)
> xbar
 [1] 9.743203
> s
 [1] 1.790519
```

```
> SEM = s/sqrt(87)
> SEM
[1] 0.1919638
```

Finally we construct the 95% confidence interval:

```
> xbar - qnorm(0.975)*SEM
[1] 9.36696
> xbar + qnorm(0.975)*SEM
[1] 10.11944
```

So the mean of the distribution from which we have drawn is actually contained within our 95% confidence interval. Good for us.

2 One sample t-test - test for the mean of a single sample

When we are constructing the confidence interval, we are saying e.g. that with 95% certainty, the mean should be within that interval. This allows us to test whether the sample could have been drawn from a distribution with a certain mean. The t-test will return the confidence interval at the desired level, the number of degrees of freedom ($n - 1$ as always), the value of t , and the probability p that the population mean could have been μ .

Let's try this with the age guessing data. Remember the woman who everyone guessed was younger than 33 from her photo? Well, what is the likelihood that if you quizzed a large number of people, their average guess would be 33, but just the groups in SI 544 happened by chance to have all guessed as they did (that is below the actual age)? We simply run the t.test on the data. But first let's load it in:

```
> ages = read.table(
"http://www-personal.umich.edu/~ladamic/si544f06/data/ageguessing.dat",head=T)
> ages
```

	A	B	C	D	E	F	G	H	I	Truth
P1	35	33	32	31.0	36	32	32	33	30	27
P2	60	57	64	56.0	63	62	58	62	60	54
P3	40	38	44	43.0	45	44	41	40	37	51
P4	45	31	39	37.0	41	39	42	42	40	55
P5	60	57	62	63.0	67	54	59	60	52	69
P6	25	27	29	30.0	25	25	23	26	22	24
P7	22	22	31	36.0	28	32	28	35	26	37
P8	65	59	66	52.5	58	60	63	70	65	62
P9	26	26	27	30.0	27	28	27	21	31	34
P10	23	28	26	22.0	27	27	25	23	34	33
P11	40	39	46	42.0	53	48	48	48	55	47
P12	42	40	52	47.0	46	46	45	48	46	40

```
> justguesses = ages[1:12,1:9]
```

```
> justguesses["P10",]
  A B C D E F G H I
P10 23 28 26 22 27 27 25 23 34
> ages["P10","Truth"]
[1] 33
```

```
> t.test( justguesses["P10",],mu=ages["P10","Truth"])
```

One Sample t-test

```

data: justguesses["P10", ]
t = -5.7076, df = 8, p-value = 0.0004504
alternative hypothesis: true mean is not equal to 33
95 percent confidence interval:
 23.32782 28.89440
sample estimates:
mean of x
 26.11111

```



So we had our 9 group guesses, ranging from 22 to 34. Then we had the self-reported age of 33. We passed the guesses to `t.test` as the sample, and we asked whether $\mu = 33$ could have been the actual mean guess of the population. The answer is a resounding no. R tells us that it's doing a one sample t-test, which is good, because we only gave it one sample. It tells us that the t-statistic is 5.7 SEMs *below* the hypothetical mean, that's a ways below. In fact, the likelihood of drawing this sample from a population whose true mean is $\mu = 33$ is $p = 10^{-3}$. So we can more than comfortably reject the hypothesis that the mean is 33. The t test also gives us a little bit of other useful info. It gives us the 95% confidence interval, as we've learned before, and \bar{x} .

Shall we try another one?



```
> t.test( justguesses["P8", ], mu=ages["P8", "Truth"])
```

One Sample t-test

```

data: justguesses["P8", ]
t = 0.0319, df = 8, p-value = 0.9753
alternative hypothesis: true mean is not equal to 62
95 percent confidence interval:
 58.04095 66.07017
sample estimates:
mean of x
 62.05556

```

The mean guessed age could in fact correspond to the self-reported age. Maybe this is a bit of a difference between men and women on the personals site (but of course we can't make such claims without doing lots more stats, which we're not going to do for the moment). Instead...

3 t-test for comparing the means of two samples

Let's switch to a different data set. One where we have the 2006 and 2007 graduates graduates and the number of courses they took. The third column **numcourses** has the number of courses they took (including things like SI 690). The first column has their specialization.

```
> # you're reading in the file coursesbyspecialization.txt
> speccourses = read.table(file=file.choose(),head=T)
> summary(speccourses)
      specialization      graddate      numcourses
arm          :33      Fall_2006 :17      Min.       :11.00
dual-arm-lis: 6      Summer_2006: 5      1st Qu.    :16.00
hci          :39      Winter_2006:67      Median     :17.00
iemp        :21      Winter_2007:91      Mean       :17.95
info        :26                          3rd Qu.    :19.00
lis         :55                          Max.       :40.00
```

The **summary()** function has given us the numerical summary of the number of classes taken, and the number of students in each specialization. Suppose we could only interview a few students (rather than having this nice more or less complete data set). Let's sample 10 students and test whether the mean of the population could be 17.

```
> sisample = sample(speccourses$numcourses,10,replace=F)

> sisample
[1] 15 20 18 20 17 17 15 15 18 16

> t.test(sisample,mu=18)
```

One Sample t-test

```
data:  sisample
t = -1.4886, df = 9, p-value = 0.1708
alternative hypothesis: true mean is not equal to 18
95 percent confidence interval:
 15.73227 18.46773
sample estimates:
mean of x
 17.1
```

Our sample had a mean of 17.1, but we still can't reject the hypothesis that the mean of the whole population could be 18 (the mean of the entire population of students is actually 17.95), but of course your sample results will vary.

More interestingly, if we have two samples, we may want to figure out if they are drawn from distributions with the same mean. For example, looking at the class enrollment data, we may want to figure out if students in one specialization take more classes than students in another.

```
> tapply(speccourses$numcourses,speccourses$specialization,mean)
      arm dual-arm-lis      hci      iemp      info      lis
17.18182 17.50000 18.20513 19.23810 18.07692 17.72727
```

tapply is a super handy function. It says to apply the function “mean” to the number of courses, but group the data by specialization first. You’ll remember that we also used **tapply()** to get averages by state for the libraries data set.

Notice that IEMP students take 19.2 courses on average but ARM students only 17.2 (2 courses fewer). We want to know whether this difference is significant. One clue is the standard deviation of the samples. For IEMP, it is over 5 (5.58), so there’s higher variance than for the other specializations. Therefore, IEMP students take more courses on average, but they are also more variable, and there are actually not that many of them (21). t-test to the rescue!

```
> tapply(speccourses$numcourses, speccourses$specialization, sd)
      arm dual-arm-lis      hci      iemp      info      lis
1.530003  1.643168  3.357470  5.584843  2.528606  3.369001

> IEMP = speccourses[speccourses$specialization=="iemp", "numcourses"]
> IEMP
[1] 20 16 17 15 17 15 17 20 19 22 40 19 17 27 14 16 17 22 17 18 19

> ARM = speccourses[speccourses$specialization=="arm", "numcourses"]
> ARM
[1] 20 17 19 16 19 18 19 16 17 18 17 17 17 17 22 17 16 15 18 15 16 18 16 16 18 16 16
[28] 17 17 15 16 17 19

> t.test(ARM, IEMP)
```

Welch Two Sample t-test

```
data:  ARM and IEMP
t = -1.6483, df = 21.925, p-value = 0.1135
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -4.6438995  0.5313454
sample estimates:
mean of x mean of y
 17.18182  19.23810
```

So the t-test says that we can’t reject the null hypothesis that IEMP and ARM students basically take the same average number of classes. Isn’t it nice to save the reputation of an entire specialization on the same page that you try and tarnish it. This is why it’s cool to use statistics and even cooler to properly report the results.

Another important thing to learn is that if we have many groups, just by chance, we are likely to find a pair that has a statistically significant difference. To control for this we need to use a **pairwise t-test**. This is different from the paired t-test we’ll get to in a moment. All it does is it inflates the p-values to correct for “accidental” findings of significance:

```
> pairwise.t.test(speccourses$numcourses, speccourses$specialization)
```

Pairwise comparisons using t tests with pooled SD

```
data:  speccourses$numcourses and speccourses$specialization
```

```

      arm dual-arm-lis hci iemp info
dual-arm-lis 1.00 -      - - -
hci          1.00 1.00  - - -
iemp         0.41 1.00  1.00 - -
info         1.00 1.00  1.00 1.00 -
```

```
lis          1.00 1.00          1.00 1.00 1.00
```

```
P value adjustment method: holm
```

Each entry is the adjusted p-value for the pairwise t-test. Since almost all of them are actually adjusted up to the max of 1, no pairwise t-test is significant.

To remember:

A pairwise t-test is used when you have 2 groups, and you're comparing them all pairwise. A paired t-test is used when you have matched pairs of observations across two samples

3.1 Application of t-tests in HCI research

Enough playing around though. Let's see how a t-test was used to do some HCI research on sensemaking by (then) PhD student Yan Qu and our very own Prof. George Furnas. Their 2005 Chi conference paper on "Sources of Structure in Sensemaking" can be downloaded from http://www.si.umich.edu/cosen/ITR_CAKP/CHI2005-sp395-qu-furnas.pdf#search=%22elderly%20drink%20furnas%22

To summarize the experiment, they had asked 30 grad students to gather information about a topic by browsing the web and write an outline for a talk they were pretending to be needing to give at a local library. The first topic was "tea" and the second was "everyday drinks for old people" (referred to here as "elderly drink"). The subjects were using a tool 'Cosen' for bookmarking useful web resources. This allowed the researchers to keep track of how many URLs the subjects were bookmarking and also how many folders they were organizing them into.

Let's load the data:

```
> sense = read.table("http://www-personal.umich.edu/~ladamic/courses/si544f06  
/data/sensemaking.txt",head=T)
```

```
> sense
```

	SubjectID	Group	Num_Folders	Num_Bookmarks
1	T1	T	5	17
2	T2	T	5	34
3	T3	T	0	6
4	T4	T	2	14
5	T5	T	4	17
6	T6	T	7	23
7	T7	T	6	30
8	T8	T	5	20
9	T9	T	11	38
10	T10	T	15	31
11	T11	T	3	13
12	T12	T	9	27
13	T13	T	8	18
14	T14	T	7	18
15	T15	T	6	45
16	E1	E	2	4
17	E2	E	4	12
18	E3	E	0	1
19	E4	E	4	14
20	E5	E	6	5
21	E6	E	0	3
22	E7	E	6	10
23	E8	E	2	7
24	E9	E	8	14
25	E10	E	7	29
26	E11	E	4	13
27	E12	E	0	9

28	E13	E	1	6
29	E14	E	4	6
30	E15	E	4	8

Summarize by task:

```
> attach(sense)
> tapply(Num_Folders,Group,mean)
      E      T
3.466667 6.200000
> tapply(Num_Bookmarks,Group,mean)
      E      T
9.4 23.4
```

We can immediately see that both the number of bookmarks and the number of folders is greater for the task of gathering information on a broad and popular topic. But we should do a proper t-test just to make sure:

```
> t.test(Num_Bookmarks ~ Group,data=sense)
```

Welch Two Sample t-test

```
data: Num_Bookmarks by Group
t = -4.3301, df = 23.817, p-value = 0.0002315
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -20.675635 -7.324365
sample estimates:
mean in group E mean in group T
      9.4          23.4
```

Yup, definitely different. Same for folders:

```
> t.test(Num_Folders ~ Group,data=sense)
```

Welch Two Sample t-test

```
data: Num_Folders by Group
t = -2.3582, df = 25.167, p-value = 0.02644
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -5.1197251 -0.3469416
sample estimates:
mean in group E mean in group T
 3.466667      6.200000
```

Notice that I've specified the argument for the t-test as a formula: **t.test((variable I'm taking the mean of) ~ (column that has two possible values, e.g. E and T))** So what is the t-test doing? It is testing whether the difference in the means of the two populations is significantly different from 0. It is computing

$$t = \frac{\bar{x}_2 - \bar{x}_1}{SEDM} \quad (4)$$

where the **standard error of the difference of means** is

$$SEDM = \sqrt{SEM_1^2 + SEM_2^2} \quad (5)$$

By default the R `t.test()` function does not assume that the variance of the two populations being sampled from is the same (this is the *Welch procedure*).

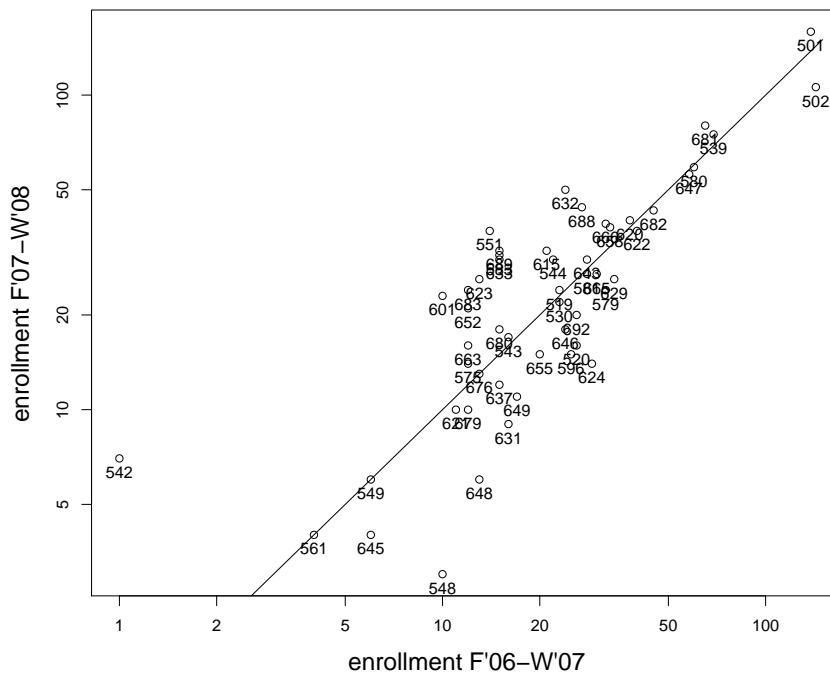
Otherwise, you can have R assume that the two populations have the same variance. `t.test(x,y,var.equal=T)`. This will then estimate a SEM by pooling all the data points into one group and taking the standard deviation. The t statistic in this case has $n_1 + n_2 - 2$ degrees of freedom.

Both options should give you similar results, but when in doubt, go with R's default and don't assume the variance in the two populations is the same.

4 The paired t-test

Paired tests are used when there are two measurements on the same experimental unit. A "before and after" or "the same subject under condition 1 and condition 2". In this case, we will consider enrollment in the same course in the 06-07 schoolyear vs. the 07-08 schoolyear, leaving out courses that were not offered in one of the years.

```
> # you'll open the file courseenroll.txt
> courses = read.table(file=file.choose(),head=T)
> bothyears = courses[((courses$F06W07>0)&(courses$F07W08>0)),]
> plot(bothyears$F06W07,bothyears$F07W08,log="xy",
+ cex.lab=1.5, xlab = "enrollment F'06-W'07",
+ ylab = "enrollment F'07-W'08")
> x = 1:150
> y = x
> lines(x,y)
> text(bothyears$F06W07,bothyears$F07W08,labels = bothyears$course,pos=1)
```



```
> attach(bothyears)
> t.test(F06W07,F07W08,paired=T)
```

Paired t-test

```
data: bothyears$F06W07 and bothyears$F07W08
t = -1.3138, df = 51, p-value = 0.1948
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -4.958984  1.035907
sample estimates:
mean of the differences
      -1.961538
```

```
> detach(bothyears)
```

We have told R that the observations are paired, we have the number of . From the t.test we can tell than the students took about 0.6 classes more in the first year compared to the second.

If we ignore that the data were paired, we get a wider confidence interval and a larger p-value. And this is undesirable.

```
> attach(bothyears)
> t.test(F06W07,F07W08,paired=F) #WRONG!
```

Welch Two Sample t-test

```
data: bothyears$F06W07 and bothyears$F07W08
t = -0.3674, df = 102, p-value = 0.7141
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -12.552246  8.629169
sample estimates:
mean of x mean of y
 27.84615 29.80769
```

```
detach(bothyears)
```

The paired t-test gave a narrower confidence interval and a lower p-value. Even though neither result is significant, we can see that with a paired test we are much more certain that the difference in the mean enrollment is in fact small. With 95% confidence, we know that there are between 1 fewer and 5 more students per course on average this year compared to last.