

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
  Group X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12
1     A 35 60 42 38 64 32 25 55 32  23  40  48
2     B 33 62 44 40 54 23 28 64 26  30  47  48
3     C 37 64 36 39 57 31 27 59 23  25  42  42
4     D 37 68 42 40 65 28 34 60 24  27  45  47
5     E 33 65 42 48 60 28 28 56 26  30  51  40
6     F 37 60 35 42 57 25 27 60 28  29  50  45
7     G 35 63 45 43 65 31 35 60 36  28  55  42
8     H 31 62 40 37 57 25 30 60 25  30  45  45
9     I 32 63 36 46 61 37 29 59 26  25  49  49
10 truth 27 54 51 55 69 24 37 62 34  33  47  40
> guesses = ages[1:9,]
> truth = ages[10,]
> guesses$X10
[1] 23 30 25 27 30 29 28 30 25
> truth$X10
[1] 33

> t.test(guesses$X10,mu=truth$X10)
```

One Sample t-test

```
data:  guesses$X10
t = -6.4018, df = 8, p-value = 0.0002087
alternative hypothesis: true mean is not equal to 33
```

```
95 percent confidence interval:
 25.44328 29.44561
sample estimates:
mean of x
 27.44444
```



So we had our 9 group guesses, ranging from 23 to 30. 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 6.4 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 $p = 10^{-4}$. 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 (I know, don't you hate that I had you do those by hand for the last assignment?), and \bar{x} .

Shall we try another one?



```
> t.test(guesses$X8,mu=truth$X8)

One Sample t-test

data:  guesses$X8
t = -3.2208, df = 8, p-value = 0.01222
alternative hypothesis: true mean is not equal to 62
95 percent confidence interval:
 57.23340 61.21105
sample estimates:
mean of x
 59.22222
```

Even this fellow was guessed to be younger than his self-reported age. But we can only reject the hypothesis that the true mean guess is 62 at the 5% and not at the 1% level since $p = 0.012$.

Let's switch to a different data set. One where we have the 2005 graduates (Winter, Spring/Summer, or Fall) who did not start taking classes before Fall 2003. The first column **num_courses** has the number of courses they took (including things like SI 690). The second column has their specialization.

```
> classenrollment =
read.table("http://www-personal.umich.edu/~ladamic/si544f06/data/numcoursesforstudents.txt",head=T)
> summary(classenrollment)
  num_courses      specialization
Min.   :15.00   ARM           :13
1st Qu.:16.00   HCI           :42
Median :17.00   IEMP          : 8
Mean   :17.37   LIS            :25
3rd Qu.:18.00   tailored       :13
Max.   :32.00
```

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(classenrollment$num_courses,10,replace=F)

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

> t.test(sisample,mu=17)
```

One Sample t-test

```
data:  sisample
t = -1.765, df = 9, p-value = 0.1114
alternative hypothesis: true mean is not equal to 17
95 percent confidence interval:
 15.63101 17.16899
sample estimates:
mean of x
 16.4
```

Our sample had a mean of 16.4, but we still can't reject the hypothesis that the mean of the whole population could be 17, which is a good thing, because the mean is actually 17.37.

3 t-test for comparing the means of two samples

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(classenrollment$num_courses,classenrollment$specialization,mean)
  ARM      HCI      IEMP      LIS tailored
16.61538 17.59524 19.50000 16.88000 17.00000
> tapply(classenrollment$num_courses,classenrollment$specialization,sd)
  ARM      HCI      IEMP      LIS tailored
0.7679476 2.8802060 3.9279220 1.1298968 1.7320508
```

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. Do you remember when we were doing those loops to figure out averages by state for library data? Well, we could have saved ourselves some trouble by just using tapply:

```
> tapply(libraries$LIBRARIAN/libraries$POPU*1000,libraries$STABR,mean)
      AK      AL      AR      AZ      CA      CO      CT      DC
0.67199991 0.34353128 0.08338738 0.29672054 0.15043927 0.33314519 0.23951910 0.29097606
      DE      FL      GA      HI      IA      ID      IL      IN
0.18197145 0.16449086 0.07969554 0.13432197 0.41013489 0.45896533 0.38638645 0.36022637
      KS      KY      LA      MA      MD      ME      MI      MN
0.52989149 0.24299701 0.11846032 0.33807690 0.24109363 0.21917676 0.23081505 0.34201587
      MO      MS      MT      NC      ND      NE      NH      NJ
0.24776455 0.18437426 0.35308872 0.07551917 0.31643625 0.52267907 0.43155079 0.16754934
      NM      NV      NY      OH      OK      OR      PA      RI
0.47924908 0.45921956 0.34703159 0.31953871 0.43025236 0.30437731 0.15021074 0.19752814
      SC      SD      TN      TX      UT      VA      VT      WA
0.12093127 0.42058878 0.08314324 0.18973665 0.22884649 0.13233361 0.25347344 0.34842758
      WI      WV      WY
0.31433425 0.20893042 0.41440443
```

Now wasn't that much shorter (as an aside, I had to clean the data a bit, because the state abbreviations for IL had numeric codes after them, e.g. IL0001, but this approach will work if all your data is nicely categorized in one column).

OK, but let's stick to the class enrollment data for the time being. You'll notice that IEMP students take 19.5 courses on average but ARM students only 16.6. We want to know whether this difference is significant. One clue is the standard deviation of the samples. For IEMP, it is almost 4 (3.93), higher than for the other specializations. So IEMP students take more courses on average, but they are also more variable, and there are actually not that many of them (8). t-test to the rescue!

```
> IEMP.courses = classenrollment[classenrollment$specialization=="IEMP",1]
> IEMP.courses
[1] 19 19 17 27 24 17 17 16
> ARM.courses = classenrollment[classenrollment$specialization=="ARM",1]
> ARM.courses
[1] 17 16 17 17 16 17 15 17 16 18 17 16 17
> t.test(IEMP.courses,ARM.courses)
```

Welch Two Sample t-test

```
data: IEMP.courses and ARM.courses
t = 2.0532, df = 7.331, p-value = 0.07735
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.407455  6.176686
sample estimates:
mean of x mean of y
 19.50000  16.61538
```

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.

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/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=="T"],Num_Bookmarks[Group=="E"])

Welch Two Sample t-test

data: Num_Bookmarks[Group == "T"] and Num_Bookmarks[Group == "E"]
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:
 7.324365 20.675635
sample estimates:
mean of x mean of y
 23.4      9.4
```

Yup, definitely different. Same for folders:

```
> t.test(Num_Folders[Group=="T"],Num_Folders[Group=="E"])

Welch Two Sample t-test

data: Num_Folders[Group == "T"] and Num_Folders[Group == "E"]
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:
 0.3469416 5.1197251
sample estimates:
mean of x mean of y
 6.200000  3.466667
```

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 just Winter 2005 graduates (before we were taking anyone graduating in any semester in 2005) and look at the number of classes they took in the fall2003/winter2004 semesters and fall2004/winter2005 semesters.

```
> attach(byyear)
> t.test(first_year,second_year,paired=T)
```

Paired t-test

```
data: first_year and second_year
t = -3.0048, df = 81, p-value = 0.003536
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -1.0135163 -0.2059959
sample estimates:
mean of the differences
      -0.6097561
```

We have told R that the observations are paired, we have the number of courses taken by the student in the first year in one column and the number of courses taken by the very same student in the second year in the second column. From the `t.test` we can tell that the students took about 0.6 classes more in the first year compared to the second.

Normally if you would ignore that the data were paired, you would get a wider confidence interval and a larger p-value. And this is undesirable. In this case we get a lower p-value, but in general it is always more correct to do the paired test if the data are in fact paired. I'm going to speculate that there is a slight (although not significant) anti-correlation between number of classes taken in the first year and those taken in the second year. If you took a bunch your first year - you need to take fewer your second, and vice versa. Pairing then may reduce the significance in a test that is looking for a one-directional difference.

```
> t.test(first_year,second_year) # WRONG!
```

Welch Two Sample t-test

```
data: first_year and second_year
t = -3.1801, df = 161.894, p-value = 0.001765
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.9883955 -0.2311167
sample estimates:
mean of x mean of y
 8.292683  8.902439
```

If however we use the dataset from the Dalgaard book, we see the benefits of pairing data more clearly. The data is paired by woman, and has her pre- and post- menstrual energy intake.

```
> data(intake)
> attach(intake)
> intake
  pre post
1 5260 3910
2 5470 4220
3 5640 3885
4 6180 5160
5 6390 5645
6 6515 4680
7 6805 5265
8 7515 5975
9 7515 6790
10 8230 6900
11 8770 7335
> t.test(pre,post,paired=T)
```

Paired t-test

```
data: pre and post
t = 11.9414, df = 10, p-value = 3.059e-07
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 1074.072 1566.838
sample estimates:
mean of the differences
      1320.455
```

```
> t.test(pre,post) # WRONG!
```

Welch Two Sample t-test

```
data: pre and post
t = 2.6242, df = 19.92, p-value = 0.01629
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 270.5633 2370.3458
sample estimates:
mean of x mean of y
 6753.636 5433.182
```

The paired t-test gave a narrower confidence interval and a lower p-value, both of which are desirable.