

# Lab 6 – Fundamental Principles of Parametric Statistics

## SE, CI

Questions? montwe@ualberta.ca or isaacren@ualberta.ca

This lab provides more theoretical background than previous labs. There is not much new if you understood everything that we interactively developed during the lectures. However, here is a chance to review what you learned or to catch up. There are also a few additional challenges that you can try out to see if you now understand the principles of parametric statistics.

### 6.1 Standard Error

The standard error of the mean ( $SEM_x$ ) is the standard deviation of a sample-based estimate of the population mean. If you repeated your experiment over and over again, you would each time get a slightly different value. How much this value varies is a measure of precision of your mean estimate. The beauty of the  $SEM_x$  is that you don't actually have to repeat your experiment over and over again to get this estimate. Let's see if we can confirm this empirically ...

R has a random number generator `rnorm()` that generates a string of random numbers with a mean of 0 and a standard deviation of 1. For example, `rnorm(10)` will “sample” 10 random numbers that come from a theoretical distribution where the population mean is 0 and the standard deviation is 1.

You can modify this command to sample from a different distribution with a user-defined mean and standard deviation by specifying these terms within the command. Repeatedly run the code below (each time you get a slightly different random `s1` dataset) and fill out the table below.

```
s1=rnorm(10,mean=15,sd=5)
s1
mean(s1)
sd(s1)
se1=sd(s1)/sqrt(10)
```

Sample	$\bar{x}$	n	SD	SE=SD/ $\sqrt{n}$
s1		10		
s2		10		
s3		10		
s4		10		
s5		10		
s6		10		

**SD**

**Mean**

The standard deviation of the means (column  $\bar{x}$ ) should be approximately the same as any SE-estimate, and very close to the mean of multiple  $SEM_x$  estimates.

You can double-check this yourself for your own table above with code along the lines of:

```
sd(c(your_mean_of_s1, your_mean_of_s2, your_mean_of_s3, etc...))
mean(c(your_SE_of_s1, your_SE_of_s2, your_SE_of_s3, etc...))
```

## 6.2 Confidence Intervals

The standard error of the mean (SEM<sub>x</sub>) for large sample sizes is equivalent to the 68% confidence interval of the mean. (This roughly means that you could be 68% confident that the true population mean is represented within this range.) Confirm this with the functions `qnorm()` and `pnorm()`, which convert between units of standard deviations (standard error in this case) and percentiles (= probabilities) based on a normal distribution.

The `pnorm()` function changes standard error to percentiles. If the user-specified number inside the brackets is positive, it indicates the standard error on the right side of the distribution. If the number inside is negative, it indicates the standard error on the left side of the distribution.

```
pnorm(1) # converts the SE on the right side of the mean to percentiles
pnorm(-1) # converts the SE on the left side of the mean to percentiles
qnorm(0.16) #converts percentiles back to SE (note effect of rounding)
qnorm(0.84) #converts percentiles back to SE (note effect of rounding)
```

Calculate 95% confidence intervals:`pnorm(1.96)`

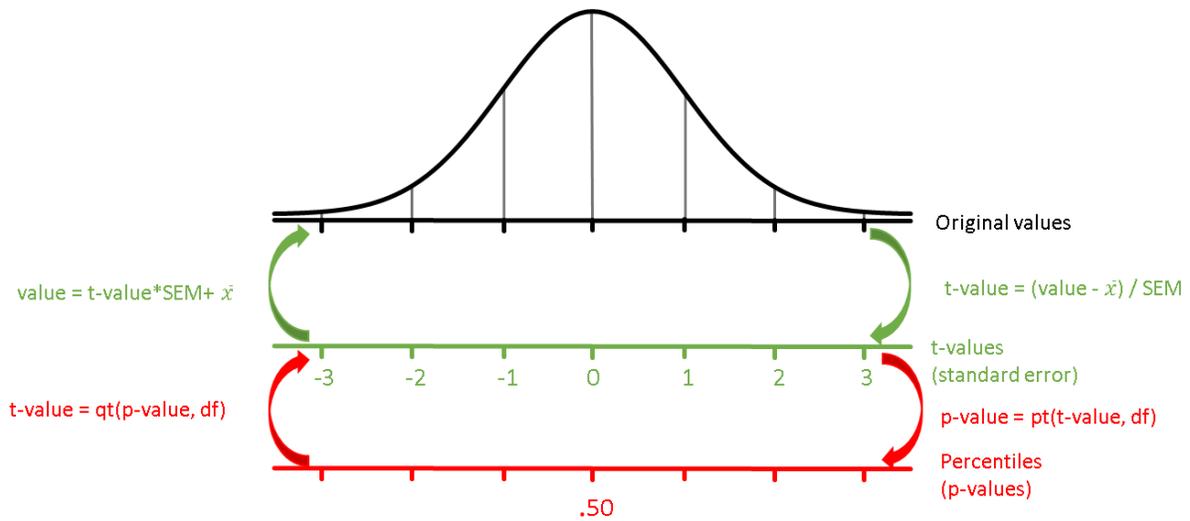
```
pnorm(-1.96)
qnorm(0.025)
qnorm(0.975)
```

Now, let's do this directly for sample-based confidence intervals using a T-distribution and the `r` functions `qt(p, df)` and `pt(sd, df)` rather than using a normal distribution. The confidence interval of a sample is then calculated as  $CI = \bar{x} \pm qt() * SE$

```
pt(1.96, 1000) #should be similar to pnorm(1.96)
pt(1.96, 9)
pt(-1.96, 9)

qt(0.025, 1000) #should be similar to qnorm(0.025, 1000)
qt(0.025, 9)
qt(0.975, 9)
```

Now, let's try to calculate the 90% confidence interval of the mean for data from a previous lab for lentil variety A (VarA). You can use the following figure to visualize your calculation:



```
VarA=c(720,740,690,760)
      mean(VarA)
sd(VarA)
sd(VarA)/sqrt(4)
qt(0.90,3)
```

```
mean(VarA) + sd(VarA)/sqrt(4) * qt(0.90,3)
mean(VarA) - sd(VarA)/sqrt(4) * qt(0.90,3)
```

You can check your results developed from first principles with the `t.test` function in R, which also provides the confidence interval of a sample if specified as an option:

```
t.test(VarA, conf.level=0.90)
```