

# Lecture 10: Bootstrap

STAT 324

Ralph Trane  
University of Wisconsin–Madison

Spring 2020



**WISCONSIN**  
UNIVERSITY OF WISCONSIN-MADISON

# Confidence Interval: The Short Story



To find confidence interval, we use the distribution of  $\frac{\bar{X} - E(\bar{X})}{SD(\bar{X})}$ .

If  $\bar{X} \sim N$ , and  $SD(\bar{X})$  is known, then  $\frac{\bar{X} - E(\bar{X})}{SD(\bar{X})} \sim N$ .

If  $\bar{X} \sim N$ , and  $SD(\bar{X})$  is unknown, then  $\frac{\bar{X} - E(\bar{X})}{\widehat{SD}(\bar{X})} \sim t_{n-1}$ .

So, when is  $\bar{X} \sim N$ ?

1. If the data are normal, i.e.  $X_1, \dots, X_n \sim N$ .
  - check using histogram and/or QQ-plot
2. If  $n \geq 30$ , then CLT tells us  $\bar{X} \sim N$  (in most real life scenarios...)

What if the data are not normal, and  $n < 30$ ?!?!?!?



What is the "gold standard" for finding the distribution of anything?

Sample from the population many, many times, and create a histogram.

That's all fun and games in theory, but in practice we cannot really do that.

Remember, all of statistics is build on one fundamental assumption: the sample looks like the population.

So what if we just... resample from the sample...?



This approach is called *bootstrapping*. How it works:

1. Grab your bootstraps
2. Pull yourself up!



This approach is called *bootstrapping*. How it works:

1. Given a sample, calculate  $\bar{x}$ .
2. Generate a new sample of size  $n$  from the original sample by sampling with replacement (!)
  - we call the first new sample  $x_{11}, x_{12}, \dots, x_{1n}$ , the second new sample  $x_{21}, x_{22}, \dots, x_{2n}$ , ..., the  $B$ 'th new sample  $x_{B1}, x_{B2}, \dots, x_{Bn}$
  - these new samples are called *bootstrap samples*
3. For each bootstrap sample, calculate  $t_j = \frac{\bar{x}_{\cdot j} - \bar{x}}{s_j / \sqrt{n}}$ .
  - here,  $\bar{x}_{\cdot j}$  is the average of the  $j$ 'th bootstrap sample, while  $\bar{x}$  is the average of the original sample.
4. Estimate the distribution of  $\frac{\bar{X} - E(\bar{X})}{\widehat{SD}(\bar{X})}$  by the distribution of  $t_1, t_2, \dots, t_B$ 
  - that is, the true distribution of  $\frac{\bar{X} - E(\bar{X})}{\widehat{SD}(\bar{X})}$  is approximately the histogram of the  $t_j$ 's.

# Confidence Intervals using Bootstrap



To find a confidence interval, we need to find  $x_1, x_2$  such that

$$P \left( x_1 \leq \frac{\bar{X} - E(\bar{X})}{\widehat{SD}(\bar{X})} \leq x_2 \right) = 1 - \alpha.$$

We can use the bootstrap samples to estimate the distribution of  $\frac{\bar{X} - E(\bar{X})}{\widehat{SD}(\bar{X})}$ , and find the cut-offs such that there's  $\alpha/2$  to the left of  $x_1$  and  $\alpha/2$  to the right of  $x_2$ .

We will call  $x_1 = \hat{t}_{1-\alpha/2}$ , and  $x_2 = \hat{t}_{\alpha/2}$  -- the  $1 - \alpha/2$  and  $\alpha/2$  critical values of the distribution of the  $\hat{t}_j$ 's.

Note: in this case, it is most likely that  $\hat{t}_{1-\alpha/2} \neq -\hat{t}_{\alpha/2}$ !!! There is no guarantee that the distribution is symmetrical, so no reason to think one of the values will be the negative of the other.



# Confidence Intervals using Bootstrap



This code takes an original sample (`orig_sample`), creates 5000 bootstrap samples, and calculates  $\hat{t}_1, \dots, \hat{t}_{5000}$ .

All of this will be in the data set `bootstrap_samples`.

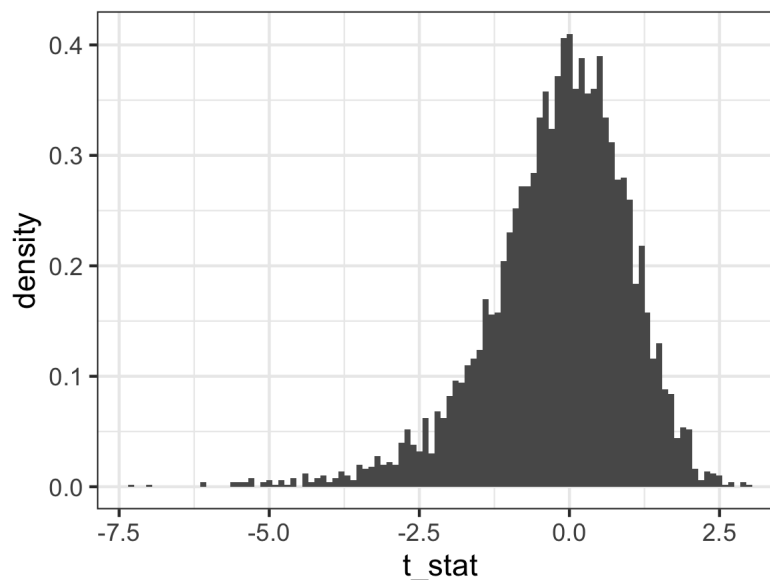
```
orig_sample <- tibble(x = c( 2.80, 16.47, 3.36,  9.31, 5.86,  
                           15.25, 27.58, 4.75, 36.20, 1.25,  
                           11.45, 10.01, 0.75,  0.59, 1.40,  
                           10.54, 20.69, 1.82, 10.16, 2.83))  
  
xbar <- mean(orig_sample$x)  
sample_size <- nrow(orig_sample)  
  
bootstrap_samples <- tibble(i = 1:5000) %>%  
  mutate(boot_sample = map(i, ~sample_n(orig_sample, size = sample_size, replace = TRUE)),  
         boot_mean = map_dbl(boot_sample, mean),  
         boot_sd = map_dbl(boot_sample, sd),  
         t_stat = (boot_mean - xbar)/(boot_sd/sqrt(sample_size)))
```

# Confidence Intervals using Bootstrap



We can then create a histogram of the  $\hat{t}_j$ 's:

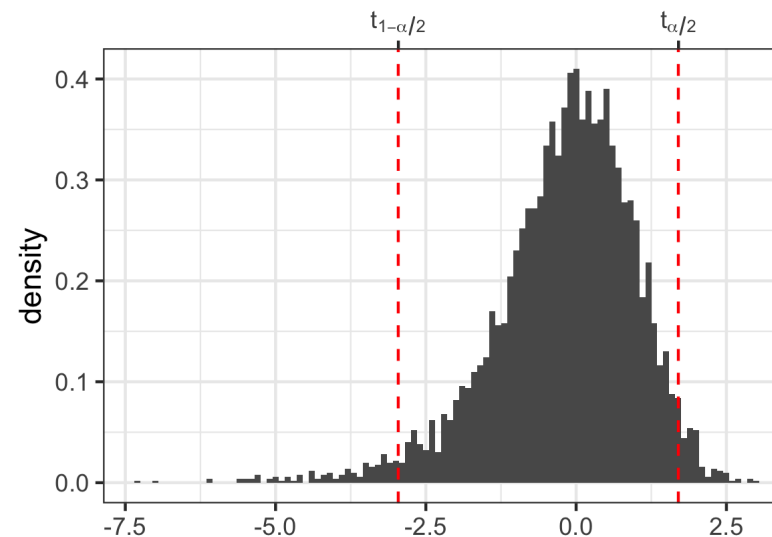
```
ggplot(bootstrap_samples,  
      aes(x = t_stat)) +  
  geom_histogram(binwidth = 0.1,  
                aes(y = ..density..))
```



# Confidence Intervals using Bootstrap



$\hat{t}_{1-\alpha/2}$  and  $\hat{t}_{\alpha/2}$  are by definition the numbers that cut-off  $1 - \alpha/2$  and  $\alpha/2$  of the area to the right, respectively.



In this case, the numbers are:

```
bootstrap_samples %>%  
  summarize(t_left = quantile(t_stat, 0.025),  
            t_right = quantile(t_stat, 0.975))
```

```
## # A tibble: 1 x 2  
##   t_left t_right  
##   <dbl> <dbl>  
## 1  -2.96    1.70
```

# Confidence Intervals using Bootstrap



So,

$$\begin{aligned} 1 - \alpha &= P \left( \hat{t}_{1-\alpha/2} \leq \frac{\bar{X} - E(\bar{X})}{\widehat{SD}(\bar{X})} \leq \hat{t}_{\alpha/2} \right) \\ &= P \left( \hat{t}_{1-\alpha/2} \widehat{SD}(\bar{X}) \leq \bar{X} - \mu \leq \hat{t}_{\alpha/2} \widehat{SD}(\bar{X}) \right) \\ &= P \left( -\bar{X} + \hat{t}_{1-\alpha/2} \widehat{SD}(\bar{X}) \leq -\mu \leq -\bar{X} + \hat{t}_{\alpha/2} \widehat{SD}(\bar{X}) \right) \\ &= P \left( \bar{X} - \hat{t}_{1-\alpha/2} \widehat{SD}(\bar{X}) \geq \mu \geq \bar{X} - \hat{t}_{\alpha/2} \widehat{SD}(\bar{X}) \right) \\ &= P \left( \bar{X} - \hat{t}_{\alpha/2} \widehat{SD}(\bar{X}) \leq \mu \leq \bar{X} - \hat{t}_{1-\alpha/2} \widehat{SD}(\bar{X}) \right) \end{aligned}$$

# Confidence Intervals using Bootstrap



A  $(1 - \alpha) \cdot 100\%$  Confidence Interval for the true mean  $\mu$  is

$$[\bar{X} - \hat{t}_{\alpha/2} \widehat{SD}(\bar{X}), \bar{X} - \hat{t}_{1-\alpha/2} \widehat{SD}(\bar{X})].$$

We are  $(1 - \alpha) \cdot 100\%$  confident that the true value is in this interval.

# Confidence Intervals: Chick Birth Weights



The ChickWeight data have data regarding the effect of diet on early growth of chicks.

```
ChickWeight
```

```
## # A tibble: 578 x 4
##   weight Time Chick Diet
##   <dbl> <dbl> <ord> <fct>
## 1     42     0  1     1
## 2     51     2  1     1
## 3     59     4  1     1
## 4     64     6  1     1
## 5     76     8  1     1
## 6     93    10  1     1
## 7    106    12  1     1
## 8    125    14  1     1
## 9    149    16  1     1
## 10   171    18  1     1
## # ... with 568 more rows
```

We are interested in the mean birth weight of the chicks. This would not be affected by the diet, so treat as one big sample.

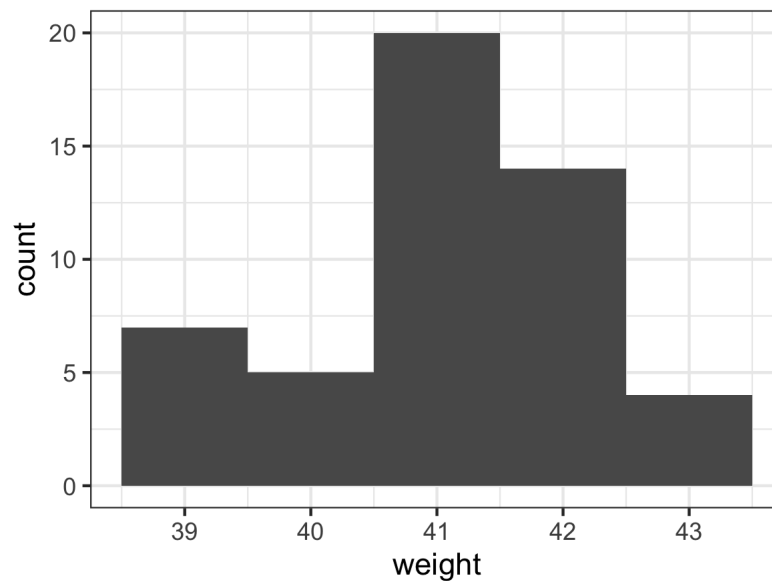
# Confidence Intervals: Chick Birth Weights



Want to find a confidence interval for  $\mu$  = true mean birth weight.

```
birth_weights <- ChickWeight %>% filter(Time == 0)

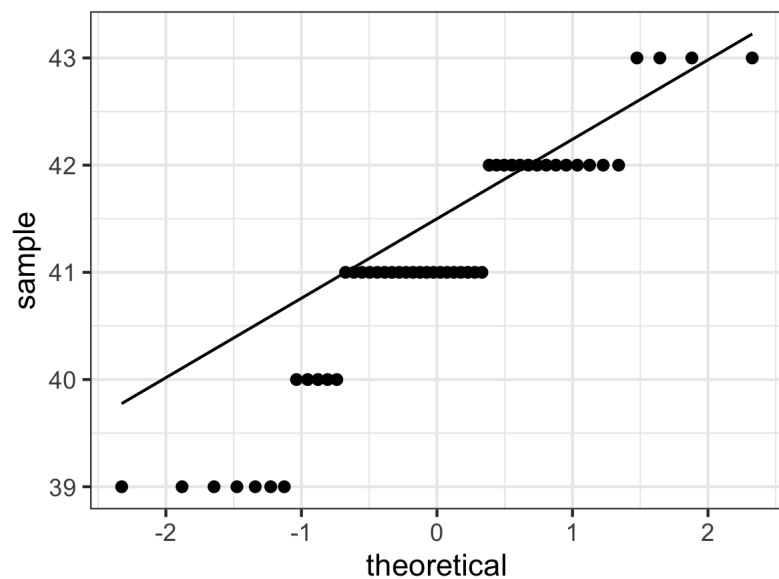
ggplot(birth_weights,
       aes(x = weight)) +
  geom_histogram(binwidth = 1)
```



# Confidence Intervals: Chick Birth Weights



```
ggplot(birth_weights,  
       aes(sample = weight)) +  
  geom_qq() +  
  geom_qq_line()
```



Definitely not normal.

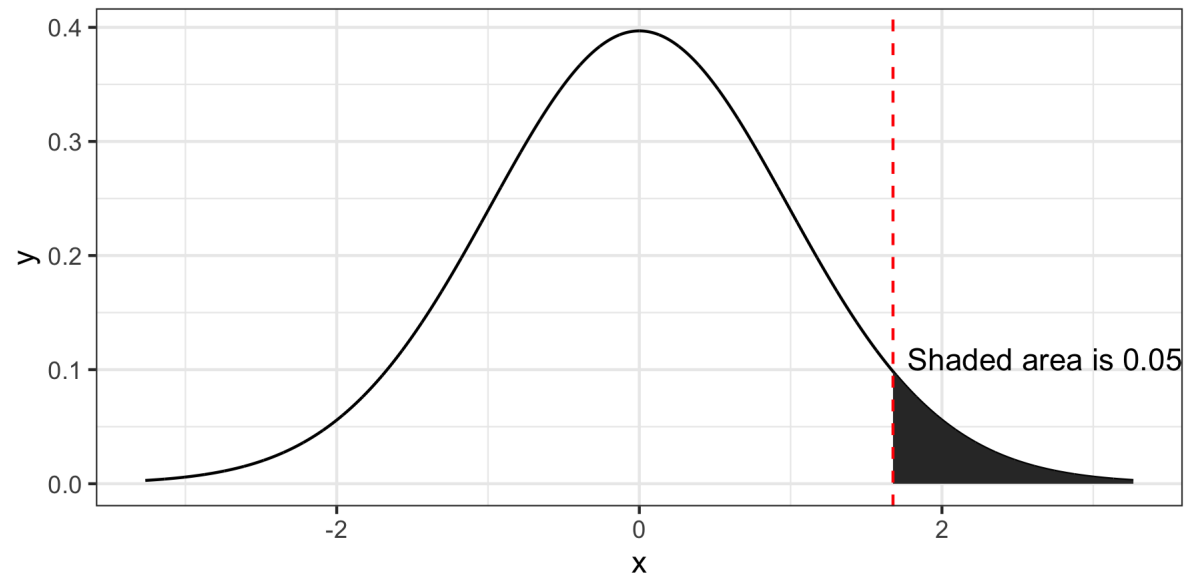


# Confidence Intervals: Chick Birth Weights



BUT...  $n = 50$ ! So, by CLT  $\bar{X} \sim N$ . Therefore, can construct a CI. Since we do not know true  $\sigma$ , find a 90% CI as  $\bar{x} \pm t_{n-1,0.05} \frac{s}{\sqrt{n}}$ .

What is  $t_{n-1,0.05}$ ? The value on x-axis such that we cut-off 0.05 to the right.



# Confidence Intervals: Chick Birth Weights



In R: remember that `quantile` finds the cut-off that cuts off to the left. To cut off 0.05 to the right, we cut off 0.95 to the left:

```
T_49 <- StudentsT(df = 49) # n-1  
  
(t_crit <- quantile(T_49, 0.95))
```

```
## [1] 1.676551
```

So, 90% CI is

```
birth_weights %>%  
  summarize(mean = mean(weight),  
            sd = sd(weight),  
            LL = mean - t_crit * sd/sqrt(50),  
            UL = mean + t_crit * sd/sqrt(50))
```

```
## # A tibble: 1 x 4  
##   mean    sd    LL    UL  
##   <dbl> <dbl> <dbl> <dbl>  
## 1  41.1  1.13  40.8  41.3
```

# Confidence Intervals: Biochemical Oxygen Demand



Measuring water quality over time. Done by measuring biochemical oxygen demand.

Data:

BOD

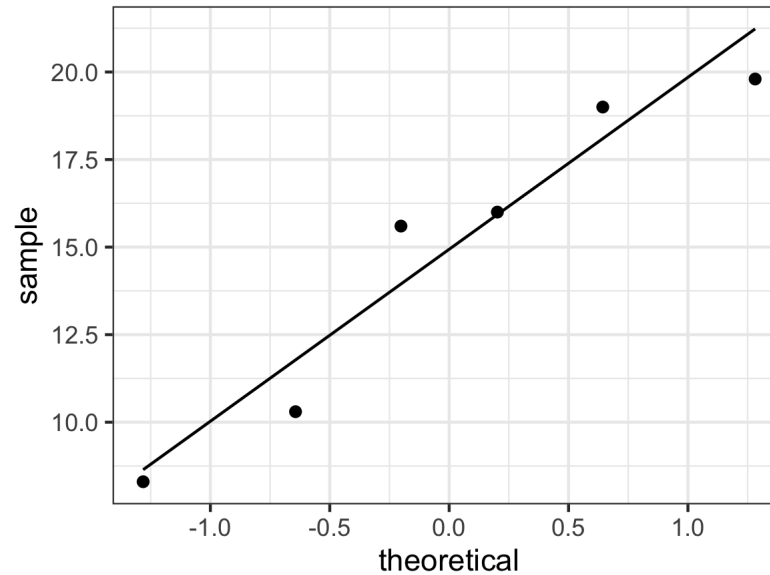
##	Time	demand
## 1	1	8.3
## 2	2	10.3
## 3	3	19.0
## 4	4	16.0
## 5	5	15.6
## 6	7	19.8

$n$  small, so cannot use CLT to conclude that  $\bar{X} \sim N$ . However, if the data is normal, we can still get to that same conclusion!

# Confidence Intervals: Biochemical Oxygen Demand



```
ggplot(BOD, aes(sample = demand)) +  
  geom_qq() +  
  geom_qq_line()
```



# Confidence Intervals: Biochemical Oxygen Demand

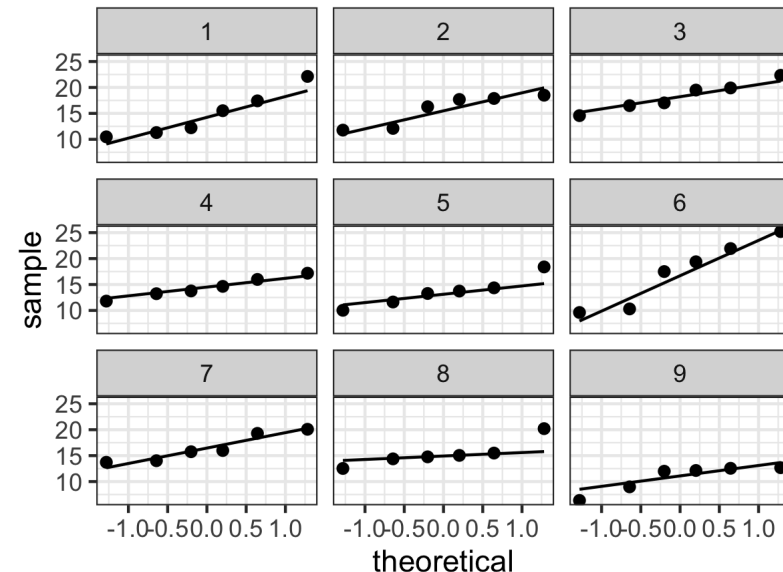


Is it straight enough? Not sure. Compare to other samples of same size that are *actually* from a normal with same mean and SD as our sample. Then ask: does our sample seem that much different?

```
X <- Normal(mu = mean(BOD$demand), sigma = sd(BOD$demand))

normal_samples <- tibble(i = 1:9) %>%
  mutate(data = map(i, ~random(X, n = nrow(BOD)))) %>%
  unnest_longer(data)
```

```
ggplot(normal_samples,
  aes(sample = data)) +
  geom_qq() +
  geom_qq_line() +
  facet_wrap(~i)
```



# Confidence Intervals: Biochemical Oxygen Demand



I would probably say no.

So, we assume  $X_1, \dots, X_6 \sim N$ . We do not know  $\sigma$ , so find 99% CI as  $\bar{x} \pm t_{n-1, 0.005} \frac{s}{\sqrt{n}}$ .

```
T_5 <- StudentsT(df = 5) # n-1  
(t_crit <- quantile(T_5, 0.995))
```

```
## [1] 4.032143
```

So, 99% CI is

```
BOD %>%  
  summarize(mean = mean(demand),  
            sd = sd(demand),  
            LL = mean - t_crit * sd/sqrt(50),  
            UL = mean + t_crit * sd/sqrt(50))
```

```
##      mean      sd      LL      UL  
## 1 14.83333 4.630623 12.19281 17.47386
```

# Confidence Intervals: Effect of Drug on Sleep



Scientists are interested in the effect of soporific drugs on amount of sleep. Data actually has data for 10 patients in 2 groups, but we will only consider one of the groups.

```
sleep1 <- sleep %>% filter(group == 1)
sleep1
```

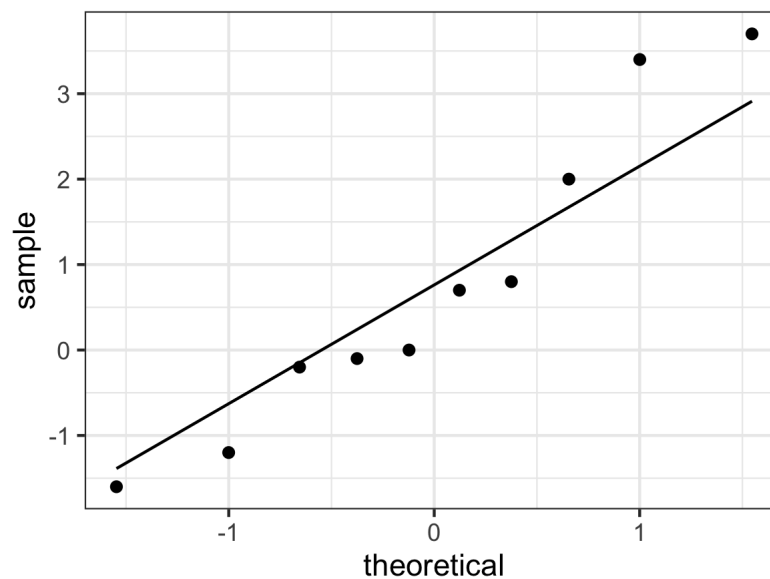
##	extra	group	ID
## 1	0.7	1	1
## 2	-1.6	1	2
## 3	-0.2	1	3
## 4	-1.2	1	4
## 5	-0.1	1	5
## 6	3.4	1	6
## 7	3.7	1	7
## 8	0.8	1	8
## 9	0.0	1	9
## 10	2.0	1	10

Small  $n$ , so no CLT. Is it normal?

# Confidence Intervals: Effect of Drug on Sleep



```
ggplot(sleep1,  
      aes(sample = extra)) +  
  geom_qq() +  
  geom_qq_line()
```





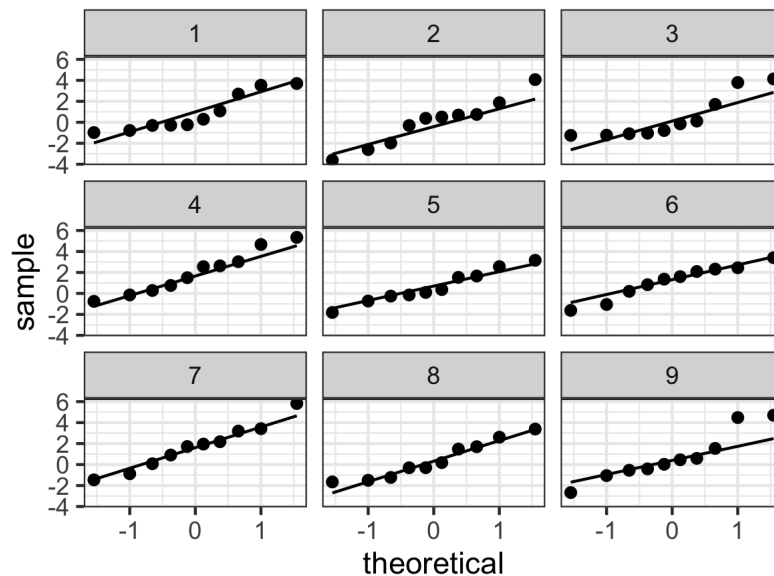
# Confidence Intervals: Effect of Drug on Sleep



Is it straight enough? Not sure. Compare to other samples of same size that are *actually* from a normal with same mean and SD as our sample. Then ask: does our sample seem that much different?

```
X <- Normal(mu = mean(sleep1$extra), sigma = sd(sleep1$extra))  
  
normal_samples <- tibble(i = 1:9) %>%  
  mutate(data = map(i, ~random(X, n = nrow(sleep1)))) %>%  
  unnest_longer(data)
```

```
ggplot(normal_samples,  
  aes(sample = data)) +  
  geom_qq() +  
  geom_qq_line() +  
  facet_wrap(~i)
```



# Confidence Intervals: Effect of Drug on Sleep



Not sure. After consulting with the scientists in charge of the study, it is decided that we do **NOT** want to assume normality.

So, non-normal data, and  $n$  too small for CLT. So, we opt for a bootstrap approach:

```
xbar <- mean(sleep1$extra)

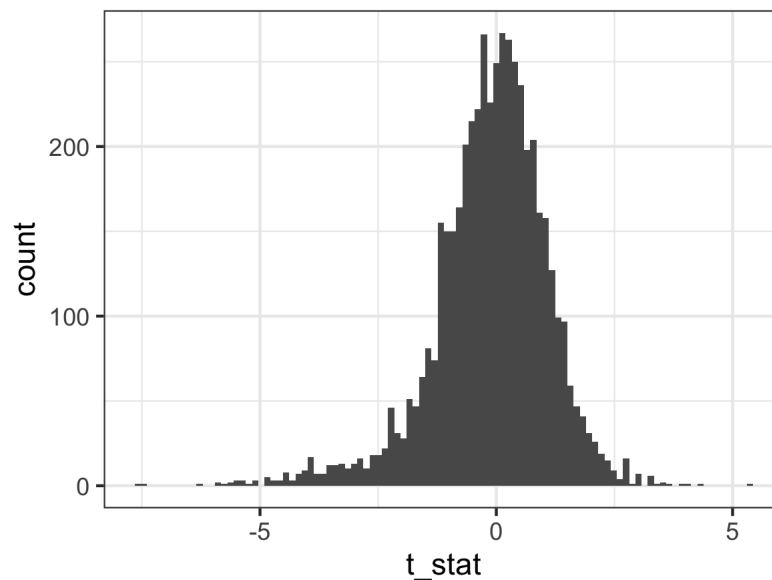
bootstrap_samples <- tibble(i = 1:5000) %>%
  mutate(boot_samples = map(i, ~sample_n(sleep1, size = 10, replace = TRUE)$extra),
         boot_mean = map_dbl(boot_samples, mean),
         boot_sd = map_dbl(boot_samples, sd),
         t_stat = (boot_mean - xbar)/(boot_sd/sqrt(10)))
```

# Confidence Intervals: Effect of Drug on Sleep



Distribution from bootstrap:

```
ggplot(data = bootstrap_samples,  
       aes(x = t_stat)) +  
  geom_histogram(bins = 100)
```

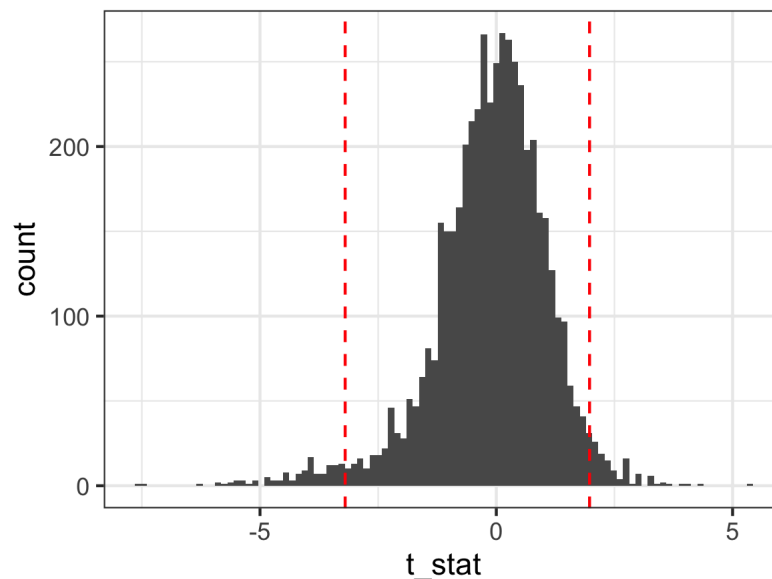


# Confidence Intervals: Effect of Drug on Sleep



Want to create a 95% CI for the true mean. Find  $\hat{t}_{0.025}$  and  $\hat{t}_{0.975}$ :

```
ggplot(data = bootstrap_samples,  
       aes(x = t_stat)) +  
  geom_histogram(bins = 100) +  
  geom_vline(xintercept = quantile(bootstrap_samples$t_stat,  
                                   p = c(0.025, 0.975)),  
            color = "red", linetype = "dashed")
```



# Confidence Intervals: Effect of Drug on Sleep



We find these, estimated mean (  $\bar{x}$  ), and standard deviation (  $s$  ) in R

```
bootstrap_samples %>%  
  summarize(t_left = quantile(t_stat, 0.025),  
            t_right = quantile(t_stat, 0.975))
```

```
## # A tibble: 1 x 2  
##   t_left t_right  
##   <dbl> <dbl>  
## 1  -3.20    1.97
```

```
sleep1 %>%  
  summarize(mean = mean(extra),  
            sd = sd(extra),  
            n = n())
```

```
##   mean      sd    n  
## 1  0.75 1.78901 10
```

# Confidence Intervals: Effect of Drug on Sleep



So, we find the lower limit of 95% CI as

$$\begin{aligned}\bar{x} - \hat{t}_{0.975} \frac{s}{\sqrt{n}} &= 0.75 - 1.973 \frac{1.789}{\sqrt{10}} \\ &= -0.37\end{aligned}$$

and the upper limit as

So, we find the lower limit of 95% CI as

$$\begin{aligned}\bar{x} - \hat{t}_{0.025} \frac{s}{\sqrt{n}} &= 0.75 - (-3.198) \frac{1.789}{\sqrt{10}} \\ &= 2.56\end{aligned}$$