Lecture 17: Two Sample Bootstrap, Wilcoxon Rank Sum Test

STAT 324

Ralph Trane
University of Wisconsin–Madison

Spring 2020

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

Two Sample Bootstrap

When sage crickets *Cyphoderris strepitans* mate, the male allows the female to eat part of his hind wings. It was thought that the hunger level of a female may influence desire to mate. An experiment was conducted where 24 females were randomly assigned to two groups. One group of 11 was starved for two days, and the other group of 13 was fed normally. Each female was presented with a male and the time to mating (in hours) was recorded. The primary research question was, ”Do starved females attempt mating more or less quickly than normally fed females?”

The primary research question was, "Do starved females attempt mating more or less quickly than normally fed females?"

The hypotheses are as follows:

$$H_0 : \mu_{\text{starved}} - \mu_{\text{fed}} = 0 \quad vs. \quad H_A : \mu_{\text{starved}} - \mu_{\text{fed}} \neq 0.$$

We will use $\alpha = 0.05$.

Two Sample Bootstrap

```
crickets <- tibble(hours = c(1.9, 2.1, 3.8, 9.0, 9.6, 13.0, 14.7, 17.9, 21.7, 29.0, 7
                             1.5, 1.7, 2.4, 3.6, 5.7, 22.6, 22.8, 39.0, 54.4, 72.1, 7
                   group = rep(c("starved", "fed"), c(11, 13)))

DT::datatable(crickets,
              options = list(pageLength = 7, dom = "tip"))
```

	hours	group
1	1.9	starved
2	2.1	starved
3	3.8	starved
4	9	starved
5	9.6	starved
6	13	starved
7	14.7	starved

Showing 1 to 7 of 24 entries

Previous 1 2 3 4 Next

Two Sample Bootstrap

```
crickets %>%
  group_by(group) %>%
  summarize(Mean = mean(hours),
            s = sd(hours),
            n = n())
```

```
## # A tibble: 2 x 4
##   group    Mean     s     n
## * <chr>   <dbl> <dbl> <int>
## 1 fed      36.0  33.6    13
## 2 starved  17.7  20.0    11
```

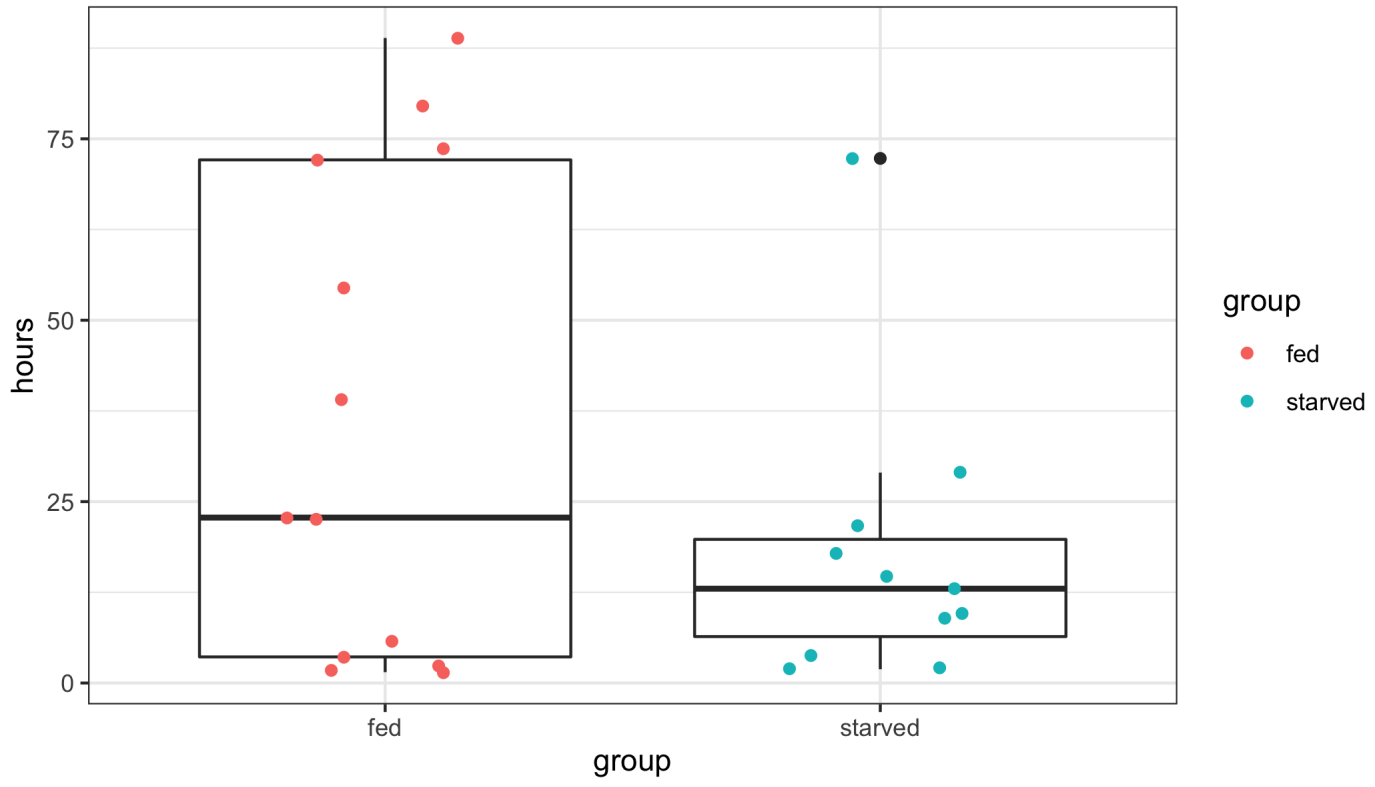
At this point, one might consider a two-sample t-test. You could argue both for and against assuming equal variances. Personally, I would not.

To actually perform a two-sample t-test, we need the following assumptions to hold:

1. Independent groups
2. Independent samples
3. Normal averages

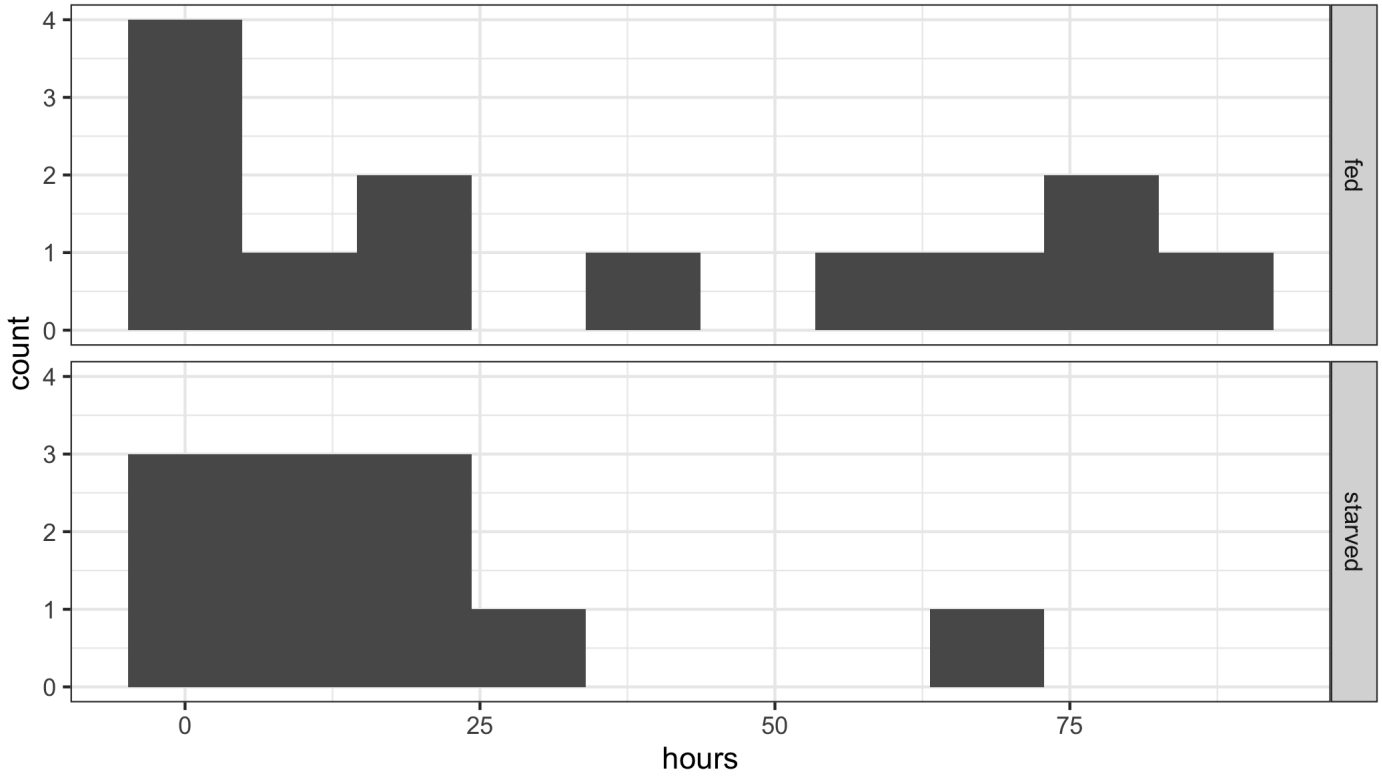
Two Sample Bootstrap

```
ggplot(crickets, aes(x = group, y = hours)) +
  geom_boxplot() +
  geom_jitter(width = 0.2, aes(color = group))
```



Two Sample Bootstrap

```
ggplot(crickets,
       aes(x = hours)) +
  geom_histogram(bins = 10) +
  facet_grid(group ~ .)
```



Two Sample Bootstrap

- sample sizes both less than 30, and data do not look normal (skewed)
- variances may or may not be equal
- potential outlier in starved group

Two sample t-test still reasonable? No. Non normal averages.

What to do instead? One option is a two-sample bootstrap test.

Two Sample Bootstrap

```
crickets %>%
  group_by(group) %>%
  summarize(ave = mean(hours),
            s = sd(hours),
            n = n()) %>% print %>%
  summarize(diff_ave = ave[2] - ave[1],
            sd_ave = sqrt(sum(s^2/n)),
            t_obs = diff_ave/sd_ave)
```

```
## # A tibble: 2 x 4
##   group     ave     s     n
## * <chr>   <dbl> <dbl> <int>
## 1 fed      36.0  33.6    13
## 2 starved  17.7  20.0    11

## # A tibble: 1 x 3
##   diff_ave sd_ave t_obs
##      <dbl>  <dbl> <dbl>
## 1    -18.3   11.1 -1.64
```

Two Sample Bootstrap

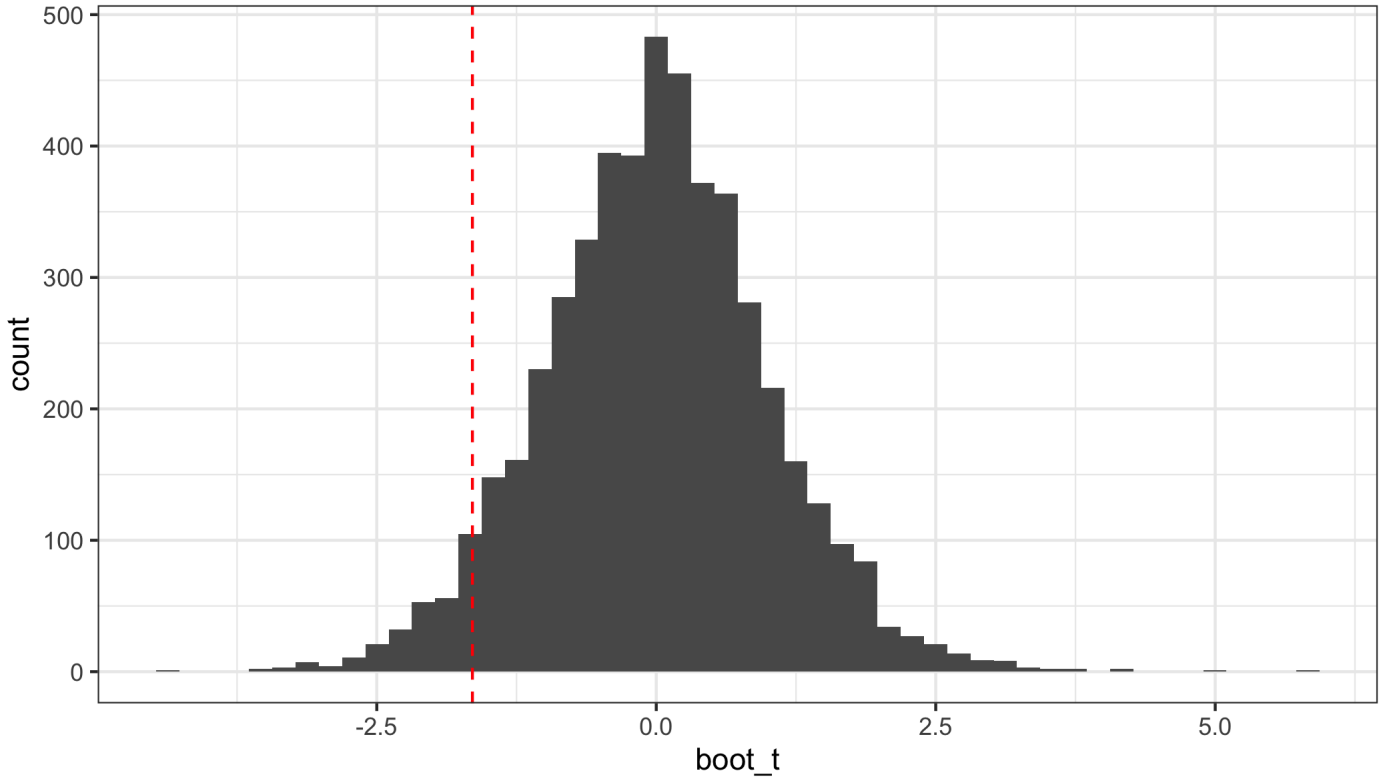
```
starved_crickets <- crickets %>%
  filter(group == "starved")

fed_crickets <- crickets %>%
  filter(group == "fed")


bootstrap_samples <- tibble(i = 1:5000) %>%
  mutate(bootstrap_fed = map(i, ~sample_n(fed_crickets,
                                          size = 13, replace = TRUE)$hours),
         bootstrap_starved = map(i, ~sample_n(starved_crickets,
                                              size = 11, replace = TRUE)$hours),
         boot_mean_fed = map_dbl(bootstrap_fed, mean),
         boot_mean_starved = map_dbl(bootstrap_starved, mean),
         boot_sd_fed = map_dbl(bootstrap_fed, sd),
         boot_sd_starved = map_dbl(bootstrap_starved, sd),
         boot_t = (boot_mean_starved - boot_mean_fed - (-18.25734))/
           (sqrt(boot_sd_starved^2/13 + boot_sd_fed^2/11)))
```

Two Sample Bootstrap

```
ggplot(bootstrap_samples,
       aes(x = boot_t)) +
  geom_histogram(bins = 50) +
  geom_vline(xintercept = -1.64476,
             linetype = "dashed",
             color = "red")
```



Two Sample Bootstrap

p-value, and 95% Confidence Interval for the difference in means:

```
bootstrap_samples %>%
  summarize(LL = -18.25734 - quantile(boot_t, 0.975)*11.10031,
            UL = -18.25734 - quantile(boot_t, 0.025)*11.10031,
            p_value = (sum(boot_t < -1.64476) + sum(boot_t > 1.64476))/5000)
```

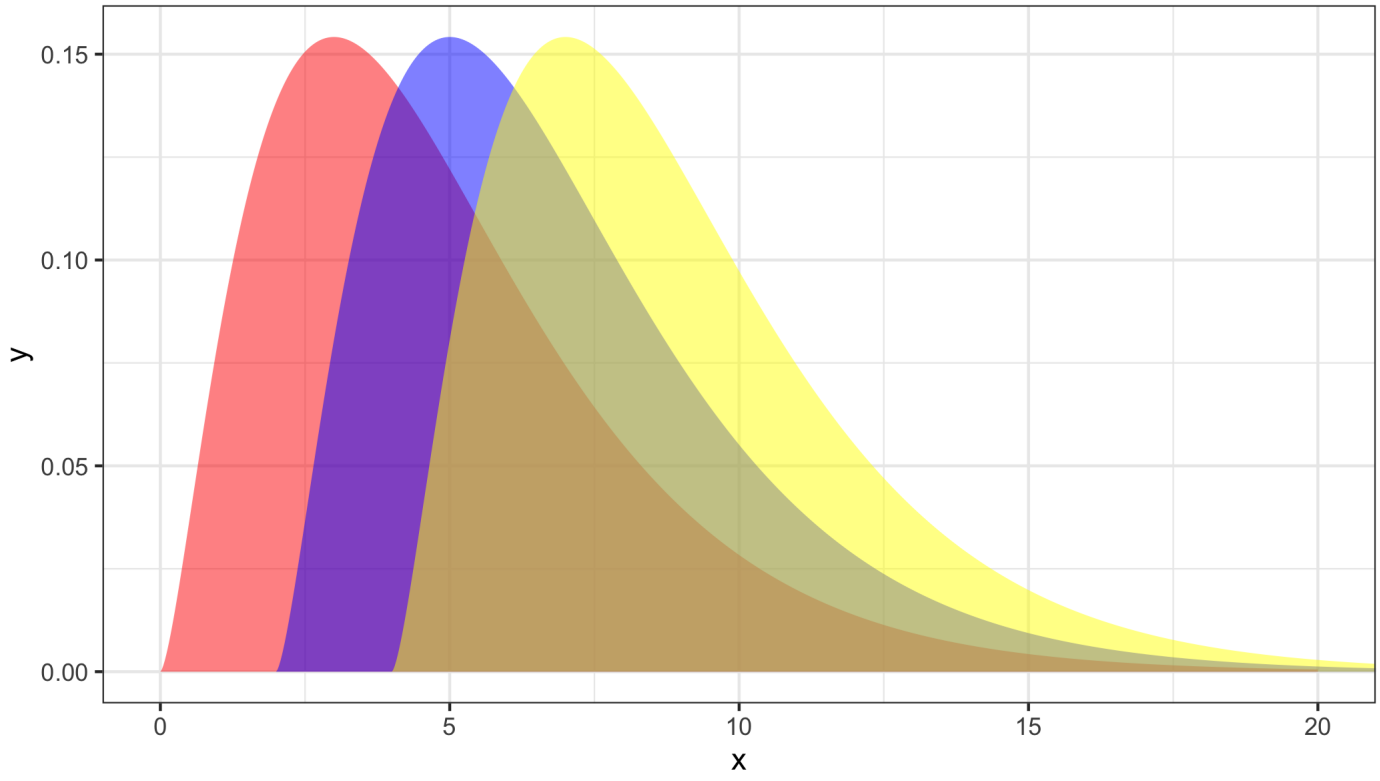
```
## # A tibble: 1 x 3
##       LL    UL p_value
##    <dbl> <dbl>   <dbl>
## 1 -40.2  3.98   0.104
```

We do not reject the null hypothesis - the data does not provide sufficient evidence to convince us that there is a difference in mean time to mating.

Wilcoxon Rank Sum Test

Different approach: change the question from "difference in means" to "shift in location".

General example:



Wilcoxon Rank Sum Test

If we compare red and yellow curves:

- generally expect values from red curve to be smaller than values from yellow
- even more sure that
    - smallest values from red < smallest values from yellow
    - middle values from red < middle values from yellow
    - largest values from red < largest values from yellow

What if there is no shift? Whether largest values from red are greater than largest values from yellow or not is basically a coin toss - could go either way.

So, if we rank all values (both those from red and yellow):

- if there is a shift, values from red will generally have lower rank than values from yellow
- if there is no shift, the ranks will be randomly distributed between the two groups.

Will use this fact for a hypothesis test for location shift. More formally:

$$H_0 : \text{two populations follow same distribution}$$

$$\text{vs.}$$

$$H_A : \text{two populations follow distributions with same shape, but on is shifted}$$

Wilcoxon Rank Sum Test

```
crickets_ranked <- crickets %>%
  mutate(rank = rank(hours))

DT::datatable(crickets_ranked,
              options = list(dom = "t",
                             paging = FA
                             scrollY = "
```

```
ggplot(crickets_ranked,
       aes(x = rank, y = group)) +
  geom_point() +
  scale_x_continuous(minor_breaks = 1:24
```

If the distribution of the observations from the starved group is shifted right compared to the distribution of the observations from the fed group, the ranks in the starved group would generally be large. I.e. the *sum of the ranks* in the starved group would be large.

	hours	group	rank
1	1.9	starved	3
2	2.1	starved	4
3	3.8	starved	7
4	9	starved	9
5	9.6	starved	10
6	13	starved	11
7	14.7	starved	12
8	17.9	starved	13

Wilcoxon Rank Sum Test

So, we somehow have to find out if the sum of the ranks in the starved group is large.

A natural thing to compare to is the *smallest* possible sum of ranks. The smallest possible sum would be if all observations in the starved group are smaller than all observations in the fed group. If this is the case, the ranks of observations in the starved group would be $1, 2, 3, \ldots, 11$. So, the sum would be $1 + 2 + \ldots + 11 = 66$.

Our *test statistic* is the difference between the observed sum of ranks, and the smallest possible sum of ranks: $U = R - R_{\text{min}}$.

```
crickets_ranked %>%
  filter(group == "starved") %>%
  summarize(R_obs = sum(rank),
            R_min = sum(1:n()),
            U_obs = R_obs - R_min)
```

```
## # A tibble: 1 x 3
##   R_obs R_min U_obs
##   <dbl> <int> <dbl>
## 1   121    66    55
```

Wilcoxon Rank Sum Test

The next question we would like to answer: "is U_{obs} unexpectedly large or small, if the null hypothesis is true?" To answer this question, we need the distribution of U so that we can find the p-value. This is where things become very tricky...

To illustrate how this is done, we will consider a much simpler scenario:

```
simple_example <- tibble(group = rep(c("
                         observations =
  mutate(rank = rank(observations))

simple_example
```

```
## # A tibble: 5 x 3
##   group observations  rank
##   <chr>        <dbl> <dbl>
## 1 A              4.8     5
## 2 A              2.2     2
## 3 B              3       3
## 4 B              1.5     1
## 5 B              3.5     4
```

For the Wilcoxon Rank Sum Test, we only focus on one of the groups. Let's pick group A.

The observed rank sum for group A:
$R_{\text{obs}} = 2 + 5 = 7.$

Smallest possible rank sum for group A:
$R_{\text{min}} = 1 + 2 = 3.$

Observed value of the test statistic:
$U_{\text{obs}} = 7 - 3 = 4.$

Wilcoxon Rank Sum Test

If the null hypothesis H_0 : the two groups follow identical distributions is true, then the ranks in group A might as well have been 3 and 4. Or 1 and 6. In fact, if H_0 is true, any possible ranking is equally likely.

How many different rankings can we get, when we are only interested in group A? $\binom{5}{2} = 10$ (choose 2 locations out of 5 possibilities.)

When the sample size is small enough, as is the case here, we can write out all possible combinations, and for each of them calculate the value of the test statistic.

This will allow us to get the pmf of the test statistic, which we then can use to find the p-value!

Wilcoxon Rank Sum Test

Rank of obs 1	Rank of obs 2	R	R_min	U	P(U)
1	2	3	3	0	0.1
1	3	4	3	1	0.1
1	4	5	3	2	0.1
1	5	6	3	3	0.1
2	3	5	3	2	0.1
2	4	6	3	3	0.1
2	5	7	3	4	0.1
3	4	7	3	4	0.1
3	5	8	3	5	0.1
4	5	9	3	6	0.1

Wilcoxon Rank Sum Test

If our alternative hypothesis is H_A : "group A is shifted to the left of B", then

$$\text{p-value} = P(U \leq U_{\text{obs}}) = \frac{8}{10} = 0.8$$

Rank of obs 1	Rank of obs 2	R	R_min	U	P(U)
1	2	3	3	0	0.1
1	3	4	3	1	0.1
1	4	5	3	2	0.1
1	5	6	3	3	0.1
2	3	5	3	2	0.1
2	4	6	3	3	0.1
2	5	7	3	4	0.1
3	4	7	3	4	0.1
3	5	8	3	5	0.1
4	5	9	3	6	0.1

Wilcoxon Rank Sum Test

If our alternative hypothesis is H_A : "group A is shifted to the left of B", then

$$\text{p-value} = P(U \leq U_{\text{obs}}) = \frac{8}{10} = 0.8$$

If our alternative hypothesis is H_A : "group A is shifted to the right of B", then

$$\text{p-value} = P(U \geq U_{\text{obs}}) = \frac{4}{10} = 0.4$$

Rank of obs 1	Rank of obs 2	R	R_min	U	P(U)
1	2	3	3	0	0.1
1	3	4	3	1	0.1
1	4	5	3	2	0.1
1	5	6	3	3	0.1
2	3	5	3	2	0.1
2	4	6	3	3	0.1
2	5	7	3	4	0.1
3	4	7	3	4	0.1
3	5	8	3	5	0.1
4	5	9	3	6	0.1

Wilcoxon Rank Sum Test

If our alternative hypothesis is H_A : "group A is shifted to the left of B", then

$$\text{p-value} = P(U \leq U_{\text{obs}}) = \frac{8}{10} = 0.8$$

If our alternative hypothesis is H_A : "group A is shifted to the right of B", then

$$\text{p-value} = P(U \geq U_{\text{obs}}) = \frac{4}{10} = 0.4$$

If our alternative hypothesis is H_A : "group A is shifted from B", then

$$\begin{aligned}\text{p-value} &= 2 \cdot \min\{P(U \leq U_{\text{obs}}), P(U \geq U_{\text{obs}}\} \\ &= 2 \cdot \frac{4}{10} = 0.8\end{aligned}$$

Rank of obs 1	Rank of obs 2	R	R_min	U	P(U)
1	2	3	3	0	0.1
1	3	4	3	1	0.1
1	4	5	3	2	0.1
1	5	6	3	3	0.1
2	3	5	3	2	0.1
2	4	6	3	3	0.1
2	5	7	3	4	0.1
3	4	7	3	4	0.1
3	5	8	3	5	0.1
4	5	9	3	6	0.1

Wilcoxon Rank Sum Test

Returning to the crickets: we have 24 crickets in total, and 11 in the starved group. I.e. there's $\binom{24}{11} = 2,496,144$ possible combinations...

We're obviously not going to calculate all possible combinations of ranks, but rather rely on R to do so for us:

```
crickets <- crickets %>%
  mutate(group = factor(group,
                        levels = c("starved", "fed")))

wilcox.test(data = crickets,
            hours ~ group)
```

```
##
##     Wilcoxon rank sum test
##
## data:  hours by group
## W = 55, p-value = 0.3607
## alternative hypothesis: true location shift is not equal to 0
```

Notice how $W = U_{\text{obs}}$.

Wilcoxon Rank Sum Test

Quick note: if we change the group we focus on, we get same p-value when the alternative is two-sided:

```
crickets <- crickets %>%
  mutate(group = factor(group,
                        levels = c("fed", "starved")))

wilcox.test(data = crickets,
            hours ~ group)
```

```
##
##     Wilcoxon rank sum test
##
## data:  hours by group
## W = 88, p-value = 0.3607
## alternative hypothesis: true location shift is not equal to 0
```

For one-sided, the p-vales are switched:

Wilcoxon Rank Sum Test

```
crickets <- crickets %>%
  mutate(group = factor(group,
                        levels = c("fed", "starved")))

wilcox.test(data = crickets,
            hours ~ group, alternative = "less")
```

```
##
##     Wilcoxon rank sum test
##
## data:  hours by group
## W = 88, p-value = 0.8344
## alternative hypothesis: true location shift is less than 0
```

```
wilcox.test(data = crickets,
            hours ~ group, alternative = "greater")
```

```
##
##     Wilcoxon rank sum test
##
## data:  hours by group
## W = 88, p-value = 0.1804
## alternative hypothesis: true location shift is greater than 0
```

Wilcoxon Rank Sum Test

```
crickets <- crickets %>%
  mutate(group = factor(group,
                        levels = c("starved", "fed")))

wilcox.test(data = crickets,
            hours ~ group, alternative = "less")
```

```
##
##     Wilcoxon rank sum test
##
## data:  hours by group
## W = 55, p-value = 0.1804
## alternative hypothesis: true location shift is less than 0
```

```
wilcox.test(data = crickets,
            hours ~ group, alternative = "greater")
```

```
##
##     Wilcoxon rank sum test
##
## data:  hours by group
## W = 55, p-value = 0.8344
## alternative hypothesis: true location shift is greater than 0
```

Wilcoxon Rank Sum Test

Assumptions made:

- independent groups
- independent samples
- shapes of the two groups approximately the same
    - hard to check!!

Pros/Cons

Before discussing pros and cons, let's just for good measure see what would have happened had we used a two sample t-test:

```
t.test(data = crickets, hours ~ group, var.equal = FALSE)
```

```
##
##     Welch Two Sample t-test
##
## data:  hours by group
## t = -1.6448, df = 19.926, p-value = 0.1157
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -41.417704   4.903019
## sample estimates:
## mean in group starved     mean in group fed
##               17.72727               35.98462
```

I chose to NOT assume equal variances:

- if variances are in fact equal, but not assumed equal, some power is lost
- if variances are NOT equal, but assumed equal, test might lead to crazy conclusions.

Therefore, when in doubt, don't assume equal variance. Here, ratios of SDs border line (1.685), so the safe choice is to NOT assume equal variances.

Pros/Cons

T-test:

- Need normality of averages
- If normal averages, more powerful
- Dealing with means, so impacted by outliers

Bootstrap:

- No need for normality
- Still means, still impacted by outliers

Wilcoxon:

- No need for normality
- Assumes similar shapes, which is hard to check
- Dealing with ranks, so more *robust* towards outliers
    - changing extreme values doesn't change ranks

Pros/Cons

```
original <- crickets %>%
  summarize(Analysis = "Original",
            t_test_p_value = t.test(hours ~ group)$p.value,
            t_test_LL = t.test(hours ~ group)$conf.int[1],
            t_test_UL = t.test(hours ~ group)$conf.int[2],
            wilcox_p_value = wilcox.test(hours ~ group)$p.value)

remove_outliers <- crickets %>%
  group_by(group) %>%
  filter(hours < quantile(hours, 0.75) + IQR(hours) * 1.5,
         hours > quantile(hours, 0.25) - IQR(hours) * 1.5) %>%
  ungroup() %>%
  summarize(Analysis = "Remove Outliers",
            t_test_p_value = t.test(hours ~ group)$p.value,
            t_test_LL = t.test(hours ~ group)$conf.int[1],
            t_test_UL = t.test(hours ~ group)$conf.int[2],
            wilcox_p_value = wilcox.test(hours ~ group)$p.value)

change_max <- crickets %>%
  mutate(hours = if_else(hours == max(hours), 10000, hours)) %>%
  summarize(Analysis = "Change Max",
            t_test_p_value = t.test(hours ~ group)$p.value,
            t_test_LL = t.test(hours ~ group)$conf.int[1],
            t_test_UL = t.test(hours ~ group)$conf.int[2],
            wilcox_p_value = wilcox.test(hours ~ group)$p.value)
```

Pros/Cons

```
bind_rows(original, remove_outliers, change_max) %>%
  knitr::kable(format = "html")
```

Analysis	t_test_p_value	t_test_LL	t_test_UL	wilcox_p_value
Original	0.1157041	-41.41770	4.903019	0.3607035
Remove Outliers	0.0287529	-44.58883	-2.840403	0.2315775
Change Max	0.3287755	-2451.49212	890.192821	0.3607035