The Not so Tiny t-test

Week 10

Packages needed and a Note about Icons

Please load up the following packages. Remember to first install the ones you don't have.

You may come across the following icons. The table below lists what each means.

Icon	Description
	Indicates that an example continues on the following slide.
	Indicates that a section using common syntax has ended.
	Indicates that there is an active hyperlink on the slide.
	Indicates that a section covering a concept has ended.

Comparing the Means Between Groups of Things

The t-test is:

- One of the most common tests in statistics
- Used to determine whether the means of two groups are equal

Ideas

> **One-sample *t*-tests**: Compare the sample mean with a known value, when the variance of the population is unknown
>
> **Two-sample *t*-tests**: Compare the means of two groups under the assumption that both samples are random, independent, and normally distributed with unknown but equal variances
>
> **Paired *t*-tests**: Compare the means of two sets of paired samples, taken from two populations with unknown variance

Packages

Please load up the following packages

```r
library(tidyverse)
library(patchwork)
```

The Base R `t.test` command

Statistical Methods I

```
t.test(x, y = NULL,
       alternative = c("two.sided", "less", "greater"),
       mu = 0,
       paired = FALSE,
       var.equal = FALSE,
       conf.level = 0.95)
```

Option	Function
`x`	a numeric vector from a data set
`y`	an optional numeric vector from a data set
`mu`	a number indicating the true value of the mean
`alternative`	preference on type of test you wish to run
`paired`	preference on whether you wish to perform a paired *t*-test
`var.equal`	indicates whether or not to assume equal variances when performing a two-sample *t*-test
`conf.level`	the confidence level of the reported confidence interval

Notes

- The `var.equals` argument has a default setting of FALSE indicating unequal variances and applies the Welsch approximation to the degrees of freedom.
    - If you wish to have equal variances, this can be done by changing the setting to TRUE
- The `conf.level` argument is set to 95%, or where $\alpha = 0.05$.
    - The confidence interval is determined by
        - μ for the one-sample t-test
        - $\mu_1 - \mu_2$ for the two-sample t-test.

Be Aware!

> The `wilcox.test` function provides the same basic functionality and arguments

> However it is used when we ***do not want to assume the data to follow a normal distribution***

> We're assuming normality

> So please ignore it for now!

Assumptions

Random sampling

Data is derived from random sampling

Independent observations

Observations are independent from one another

Normality

Observations are from a normally distributed population

Homogeneity

*If more than one population is sampled from, then the populations have equal variances (aka **homogeneity of variances**)*

One- or Two-sample *t*-tests

If `y` is

– excluded, `t.test` will run as a one-sample *t*-test

– included, `t.test` will run as a two-sample *t*-test

- default `t.test` command will run as a two-sided *t*-test
- you can run a one-sided *t*-test by changing the `alternative` option to `greater` or `less`

Example

`t.test(x, alternative = "greater", mu = 47)` performs a one-sample *t*-test on the data contained in `x` where

$$H_0 : \mu = 47$$

$$H_1 : \mu > 47$$

Example

```r
midwest %>%
  head()
```

```
## # A tibble: 6 × 28
##     PID county    state  area poptotal popdensity popwhite popblack popamerindian
##   <int> <chr>     <chr> <dbl>    <int>      <dbl>    <int>    <int>         <int>
## 1   561 ADAMS     IL    0.052    66090      1271.    63917     1702            98
## 2   562 ALEXANDER IL    0.014    10626       759      7054     3496            19
## 3   563 BOND      IL    0.022    14991       681.    14477      429            35
## 4   564 BOONE     IL    0.017    30806      1812.    29344      127            46
## 5   565 BROWN     IL    0.018     5836       324.     5264      547            14
## 6   566 BUREAU    IL    0.05     35688       714.    35157       50            65
## # … with 19 more variables: popasian <int>, popother <int>, percwhite <dbl>,
## #   percblack <dbl>, percamerindan <dbl>, percasian <dbl>, percother <dbl>,
## #   popadults <int>, perchsd <dbl>, percollege <dbl>, percprof <dbl>,
## #   poppovertyknown <int>, percpovertyknown <dbl>, percbelowpoverty <dbl>,
## #   percchildbelowpovert <dbl>, percadultpoverty <dbl>, percelderlypoverty <dbl>,
## #   inmetro <int>, category <chr>
```

Please use `?midwest` for more details on the variables

```r
midwest %>%
  filter(state == "OH" | state == "MI") %>%
  select(state, percollege)
```

```
## # A tibble: 171 × 2
##    state percollege
##    <chr>      <dbl>
##  1 MI          14.1
##  2 MI          16.3
##  3 MI          18.1
##  4 MI          18.9
##  5 MI          19.0
##  6 MI          11.8
##  7 MI          14.6
##  8 MI          17.3
##  9 MI          18.2
## 10 MI          21.4
## # … with 161 more rows
```

```
ohio_mi <-
  midwest %>%
  filter(state == "OH" | state == "MI") %>%
  select(state, percollege)
```

Descriptives

```
ohio_mi %>%
  filter(state == "OH") %>%
  summary()
```

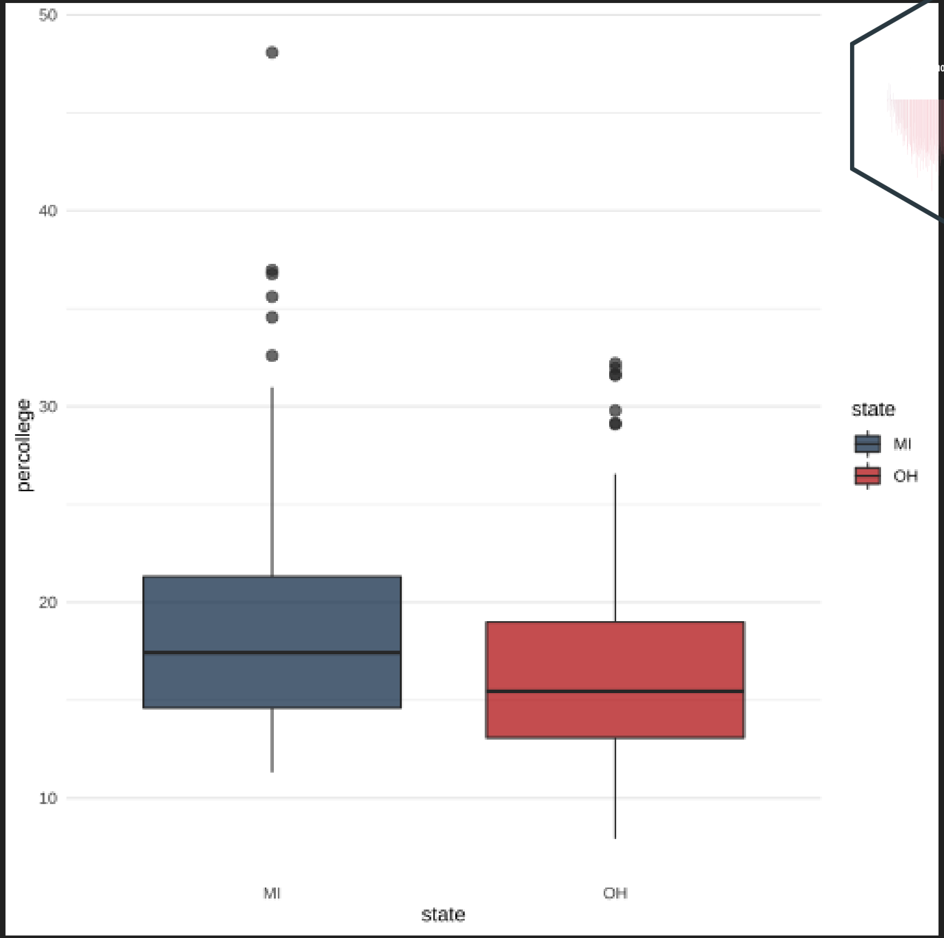
```
##     state             percollege    
##  Length:88          Min.   : 7.913  
##  Class :character   1st Qu.:13.089  
##  Mode  :character   Median :15.462  
##                     Mean   :16.890  
##                     3rd Qu.:18.995  
##                     Max.   :32.205  
```

```
ohio_mi %>%
  filter(state == "MI") %>%
  summary()
```

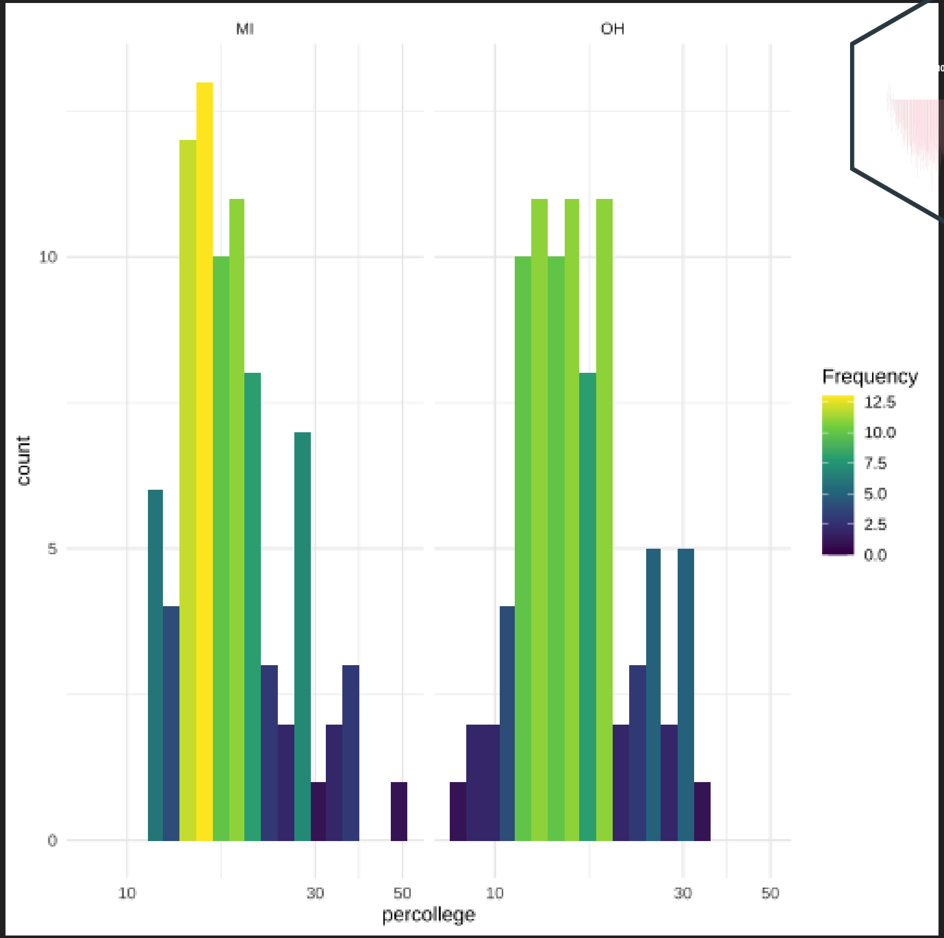
```
##     state             percollege   
##  Length:83          Min.   :11.31  
##  Class :character   1st Qu.:14.61  
##  Mode  :character   Median :17.43  
##                     Mean   :19.42  
##                     3rd Qu.:21.31  
##                     Max.   :48.08  
```

Ohio appears to have slightly less college educated adults than Michigan but let's see if that's actually true

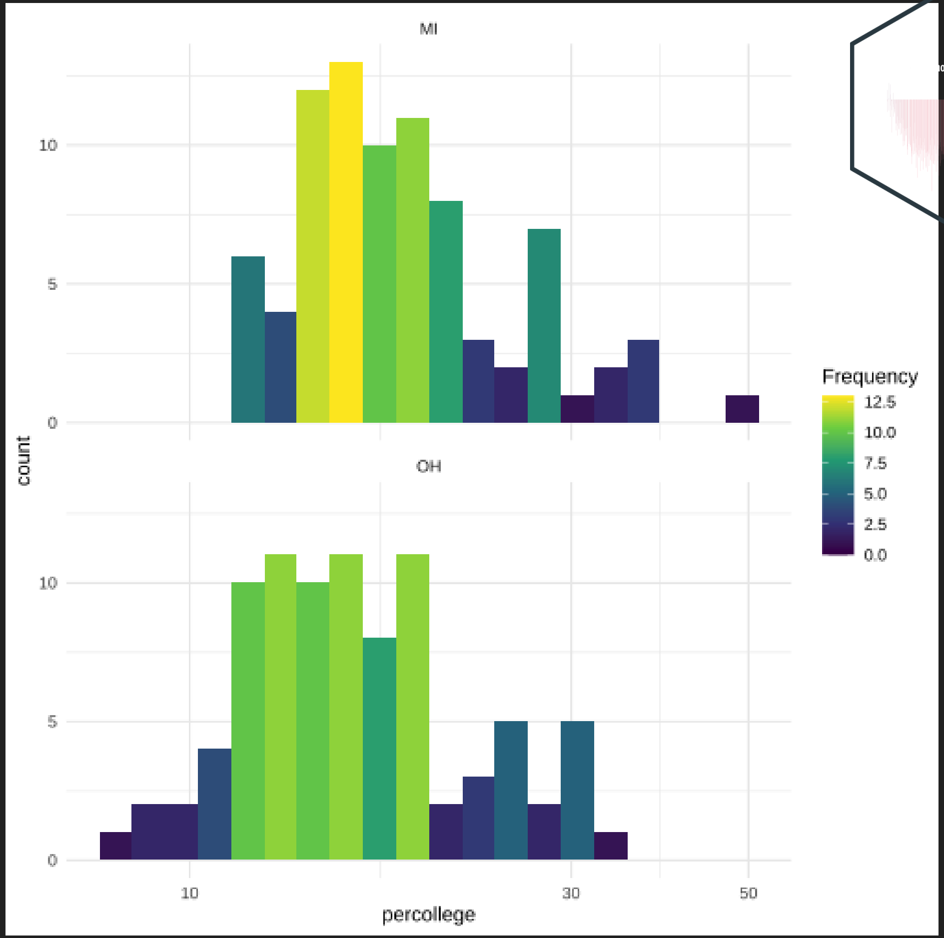
```r
ggplot(ohio_mi, aes(x = state,
                    y = percollege,
                    fill = state)) +
        geom_boxplot(alpha = 0.7,
                     outlier.size = 2.5) +
        scale_fill_manual(values = c("#00274C",
                                     "#BB0000")) +
        theme_minimal() +
        theme(panel.grid.major.x = element_blank(),
              panel.grid.minor.x = element_blank())
```



```r
ggplot(ohio_mi, aes(x = percollege)) +
        geom_histogram(aes(fill = ..count..),
                       bins = 20) +
        scale_fill_viridis_c("Frequency") +
        facet_wrap(. ~ state) +
        theme_minimal() +
        scale_x_log10()
```



```
ggplot(ohio_mi, aes(x = percollege)) +
        geom_histogram(aes(fill = ..count..),
                       bins = 20) +
        scale_fill_viridis_c("Frequency") +
        facet_wrap(. ~ state, ncol = 1) +
        theme_minimal() +
        scale_x_log10()
```

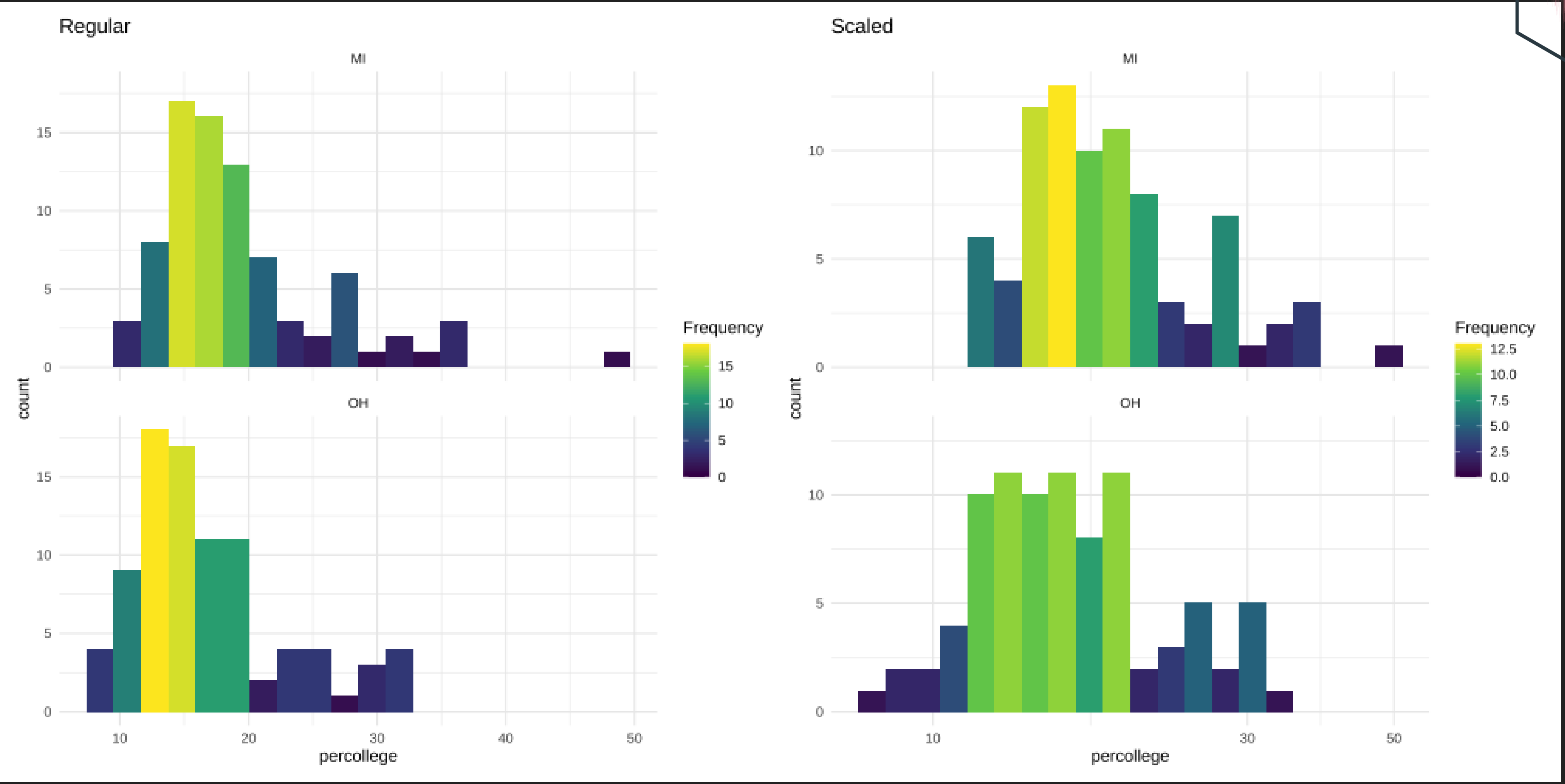


```r
regularplot <-
  ggplot(ohio_mi, aes(x = percollege)) +
  geom_histogram(aes(fill = ..count..),
                 bins = 20) +
  scale_fill_viridis_c("Frequency") +
  facet_wrap(~ state, ncol = 1) +
  theme_minimal() +
  ggtitle("Regular")
```

```r
scaledplot <-
  ggplot(ohio_mi, aes(x = percollege)) +
  geom_histogram(aes(fill = ..count..),
                 bins = 20) +
  scale_fill_viridis_c("Frequency") +
  facet_wrap(~ state, ncol = 1) +
  theme_minimal() +
  ggtitle("Scaled") +
  scale_x_log10() # this line was added
```

Statistical Methods I

```
regularplot + scaledplot
```



Testing as is

```
t.test(percollege ~ state, data = ohio_mi)
```

```
##
##  Welch Two Sample t-test
##
## data:  percollege by state
## t = 2.5953, df = 161.27, p-value = 0.01032
## alternative hypothesis: true difference in means between group MI and group OH is not equal to 0
## 95 percent confidence interval:
##  0.6051571 4.4568579
## sample estimates:
## mean in group MI mean in group OH
##         19.42146         16.89045
```

> Results show a *p*-value < .01 so **there is a statistical difference between the two means**

> This supports the alternative hypothesis that there is a difference between the average percent of college educated adults in Ohio versus Michigan

Testing using a `log` function

```r
t.test(log(percollege) ~ state, data = ohio_mi)
```

```
##
##  Welch Two Sample t-test
##
## data:  log(percollege) by state
## t = 2.9556, df = 168.98, p-value = 0.003567
## alternative hypothesis: true difference in means between group MI and group OH is not equal to 0
## 95 percent confidence interval:
##  0.04724892 0.23732151
## sample estimates:
## mean in group MI mean in group OH
##         2.915873         2.773587
```

> Results show a *p*-value < .01 so **there is a statistical difference between the two means**

> So **there is a statistical difference between the two means**

Paired-samples *t*-test

Same `t.test` command as in the previous sections but just change your option to `paired = TRUE`

```
t.test(x, y = NULL,
       alternative = c("two.sided", "less", "greater"),
       mu = 0,
       paired = TRUE,
       var.equal = FALSE,
       conf.level = 0.95)
```

Example

```
sleep %>%
  head()
```

```
##   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
```

Please use `?sleep` for more details on the variables

```r
sleep %>%
  select(-ID)
```

```
##    extra group
## 1    0.7     1
## 2   -1.6     1
## 3   -0.2     1
## 4   -1.2     1
## 5   -0.1     1
## 6    3.4     1
## 7    3.7     1
## 8    0.8     1
## 9    0.0     1
## 10   2.0     1
## 11   1.9     2
## 12   0.8     2
## 13   1.1     2
## 14   0.1     2
## 15  -0.1     2
## 16   4.4     2
## 17   5.5     2
## 18   1.6     2
## 19   4.6     2
## 20   3.4     2
```

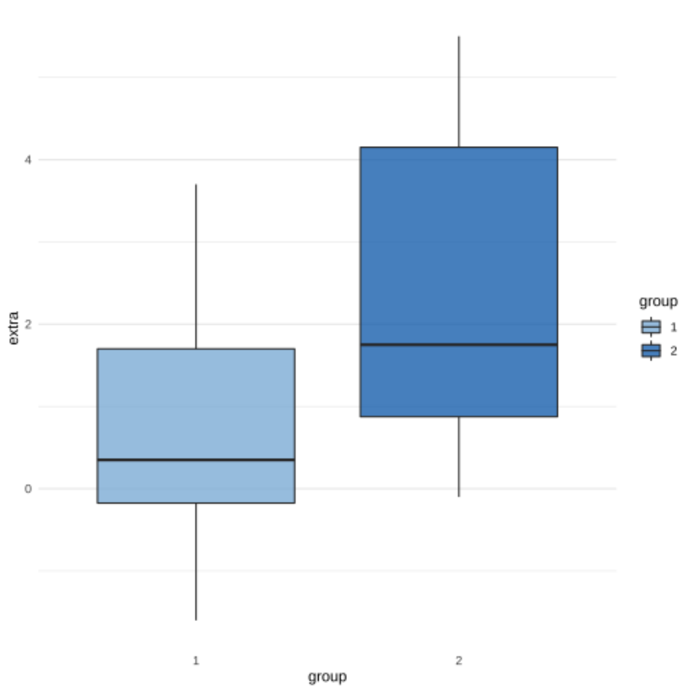
Descriptives

```r
sleep %>%
  summary()
```

```
##      extra          group        ID   
##  Min.   :-1.600   1:10   1      :2  
##  1st Qu.:-0.025   2:10   2      :2  
##  Median : 0.950          3      :2  
##  Mean   : 1.540          4      :2  
##  3rd Qu.: 3.400          5      :2  
##  Max.   : 5.500          6      :2  
##                          (Other):8  
```

Boxplot

```
sleep %>%
  ggplot(aes(group, extra, fill = group)) +
  geom_boxplot(alpha = 0.8) +
  scale_fill_manual(
    values = c("#428bca", "#d9534f")
    ) +
  theme_minimal() +
  theme(
    panel.grid.major.x = element_blank(),
    panel.grid.minor.x = element_blank()
    )
```



Asessing if there is a statistically significant effect of a particular drug on sleep (increase in hours of sleep compared to control) for 10 patients

Testing

We want to see if the mean values for the extra variable differs between group 1 and group 2

```
t.test(extra ~ group, data = sleep, paired = TRUE)
```

```
## 
##      Paired t-test
## 
## data:  extra by group
## t = -4.0621, df = 9, p-value = 0.002833
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.4598858 -0.7001142
## sample estimates:
## mean of the differences 
##                   -1.58
```

> Results show a *p*-value < .01 so **there is a statistical difference between the two means**

> This supports the alternative hypothesis that suggesting that the drug increases sleep on average by 1.58 hours

Thats it!