

# R Lab #9c - Two-sample $t$ -test

## Contents

|  |           |
|--|-----------|
| <b>Two-sample <math>t</math>-test</b>                          | <b>1</b>  |
| <b>Chap12, Problem #15: mosquitoes and beer example</b>        | <b>1</b>  |
| Import and inspect data . . . . .                              | 1         |
| Using graphs to evaluate normality . . . . .                   | 2         |
| Formally testing for normality . . . . .                       | 7         |
| Assumption of equal variances in the two samples . . . . .     | 8         |
| Running a two-sample $t$ -test . . . . .                       | 9         |
| <b>Chap12, Problem #8: mate preferences in hybrid cichlids</b> | <b>9</b>  |
| Import and inspect data . . . . .                              | 9         |
| Using graphs to evaluate normality . . . . .                   | 10        |
| Formally testing for normality . . . . .                       | 10        |
| Assumption of equal variances in the two samples . . . . .     | 11        |
| Running a Welch's $t$ -test . . . . .                          | 12        |
| <b>R commands summary</b>                                      | <b>12</b> |

## Two-sample $t$ -test

When you have two samples of data from a study with an unpaired/independent design that means that each treatment is applied to completely different sampling units. For these cases, a two-sample  $t$ -test should be used to test for equal means between the two groups. Assumptions of this test include random sampling, normality for both samples, and equal variance/standard deviations in both samples.

## Chap12, Problem #15: mosquitoes and beer example

Problem #15 at the end of the chapter summarizes a study in which the researchers investigated whether mosquitoes showed equal attraction to people that recently drank water and beer. There were 43 participants in the study. 25 of these were randomly assigned to the beer group. In this treatment, 50 mosquitoes were released near these people before and after drinking beer, and the difference in the proportion of mosquitoes attracted to the person was measured. The remaining participants were assigned to the water group, with a similar design except that the people drank water instead of beer.

At first this may seem like a paired design because repeated measurements were made on the same people. If the study was only done with people before and after drinking beer then in fact it would be a paired design.

However, this is a two-sample design. The two treatments are beer and water, and each participant was assigned to only one of these treatments (i.e., no participant was in both treatment groups). Therefore, we have two independent samples (of difference in proportion of mosquitoes attracted)

## Import and inspect data

First, we should import the data and look at the data structure:

```

mosqData <- read.csv("chap12q15BeerAndMosquitoes.csv", stringsAsFactors = T)
str(mosqData)
## 'data.frame':    43 obs. of  4 variables:
## $ drink      : Factor w/ 2 levels "beer","water": 1 1 1 1 1 1 1 1 1 1 ...
## $ beforeDrink: num  0.13 0.13 0.21 0.25 0.25 0.32 0.43 0.44 0.46 0.5 ...
## $ afterDrink : num  0.49 0.59 0.27 0.43 0.5 0.5 0.37 0.3 0.58 0.89 ...
## $ change     : num  0.36 0.46 0.06 0.18 0.25 0.18 -0.06 -0.14 0.12 0.39 ...

```

We can see that there are 43 rows and 4 columns. The first column (drink) differentiates beer and water treatments, the second column (beforeDrink) has the proportion of mosquitoes attracted to the person before drinking, the third column (afterDrink) the proportion after, and the fourth column (change) is the difference in proportions (after minus before).

Recall from Chapter 1 that it is recommended that each row corresponds to an individual/unit. Thus, in a paired design when both samples are measured from the same units you have the numbers for each sample in separate columns (but “paired” to the same row). In this case, the beer and water samples were measured in different people, so the numbers for each sample are “stacked” in a single column.

## Using graphs to evaluate normality

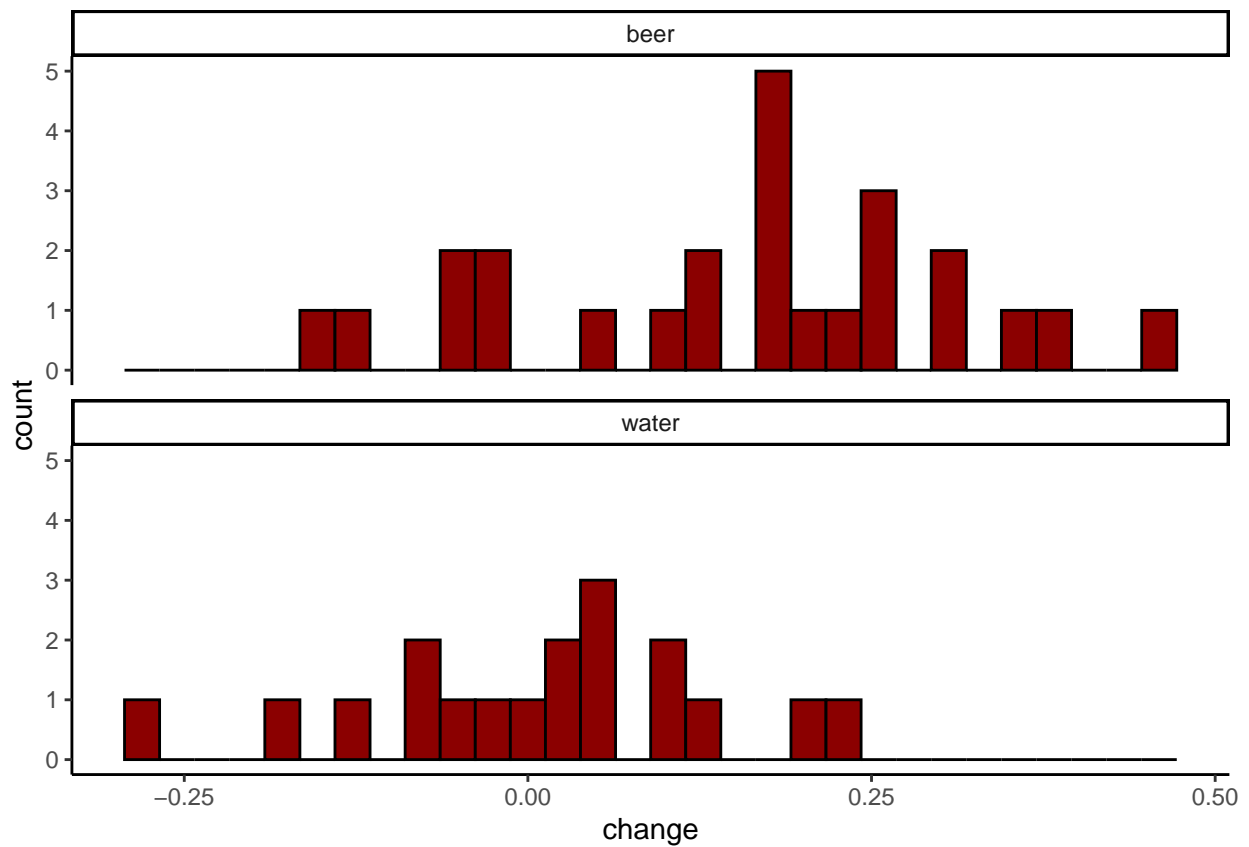
An assumption of the two-sample  $t$ -test is that each sample has a normal distribution. Generally, it is useful to graph the data to see if it appears normal.

First, let’s make frequency distribution histograms for each sample:

```

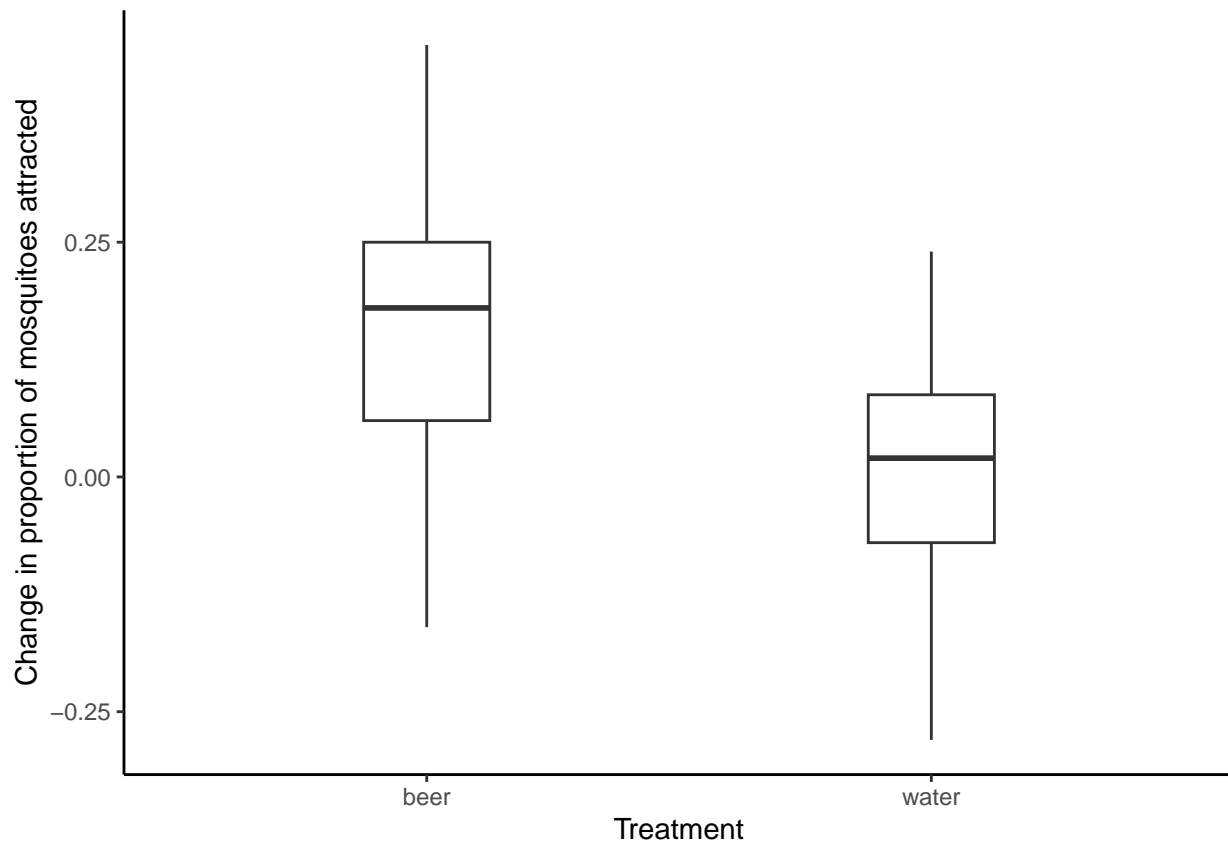
library(ggplot2)
ggplot(mosqData, aes(x = change)) + geom_histogram(color = "black",
  fill = "darkred") + facet_wrap(~drink, ncol = 1) + theme_classic()
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

```



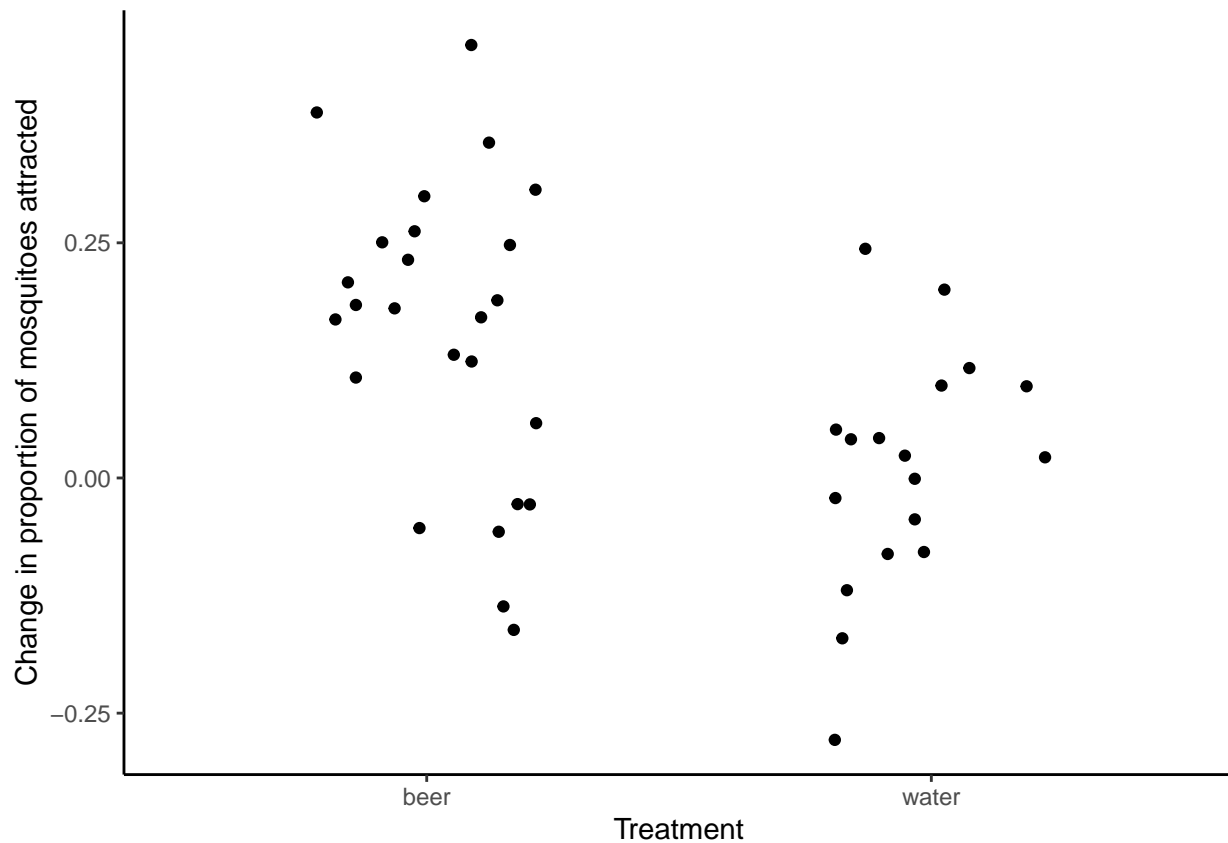
Now a box plot for each group:

```
ggplot(mosqData, aes(x = drink, y = change)) + geom_boxplot(width = 0.25) +
  xlab("Treatment") + ylab("Change in proportion of mosquitoes attracted") +
  theme_classic()
```



Now a strip plot for each group:

```
ggplot(mosqData, aes(x = drink, y = change)) + geom_jitter(width = 0.25) +  
  xlab("Treatment") + ylab("Change in proportion of mosquitoes attracted") +  
  theme_classic()
```

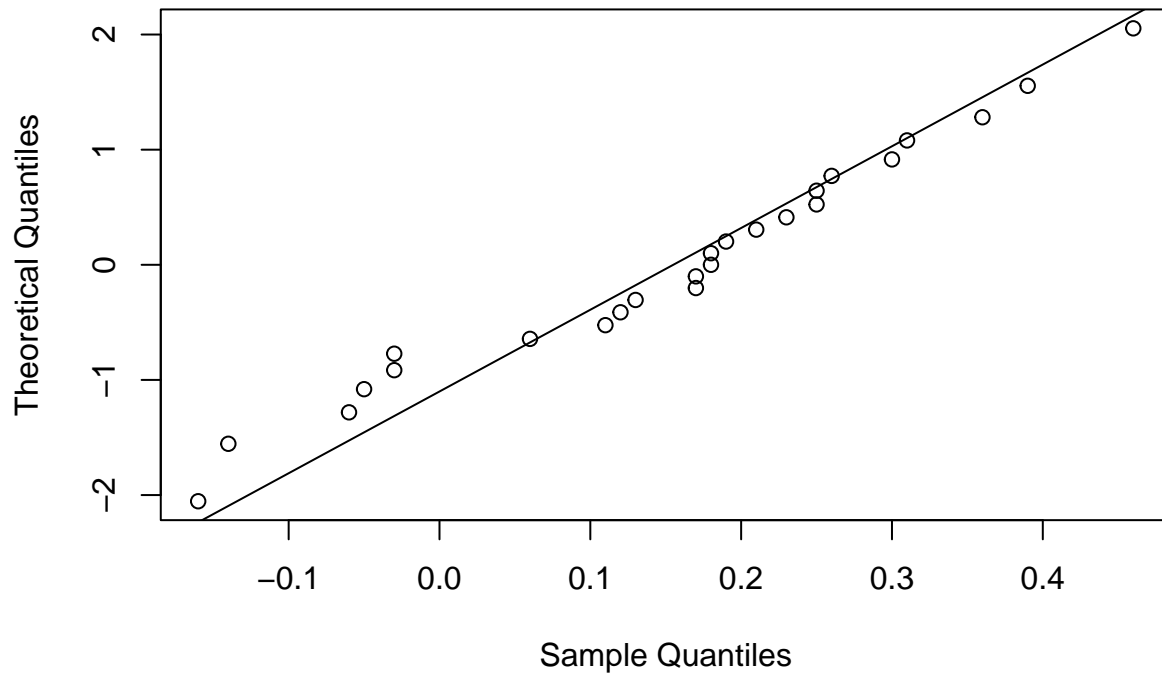


Now a normal quantile plot for each group (see Chapter 13). Since both samples are in the same column we need to filter one sample at a time. This can be done using the command `filter()` in the `dplyr` library that was introduced in an earlier R lab.

First beer:

```
library(dplyr)
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##   filter, lag
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
qqnorm(filter(mosqData, drink == "beer")$change, datax = T)
qqline(filter(mosqData, drink == "beer")$change, datax = T)
```

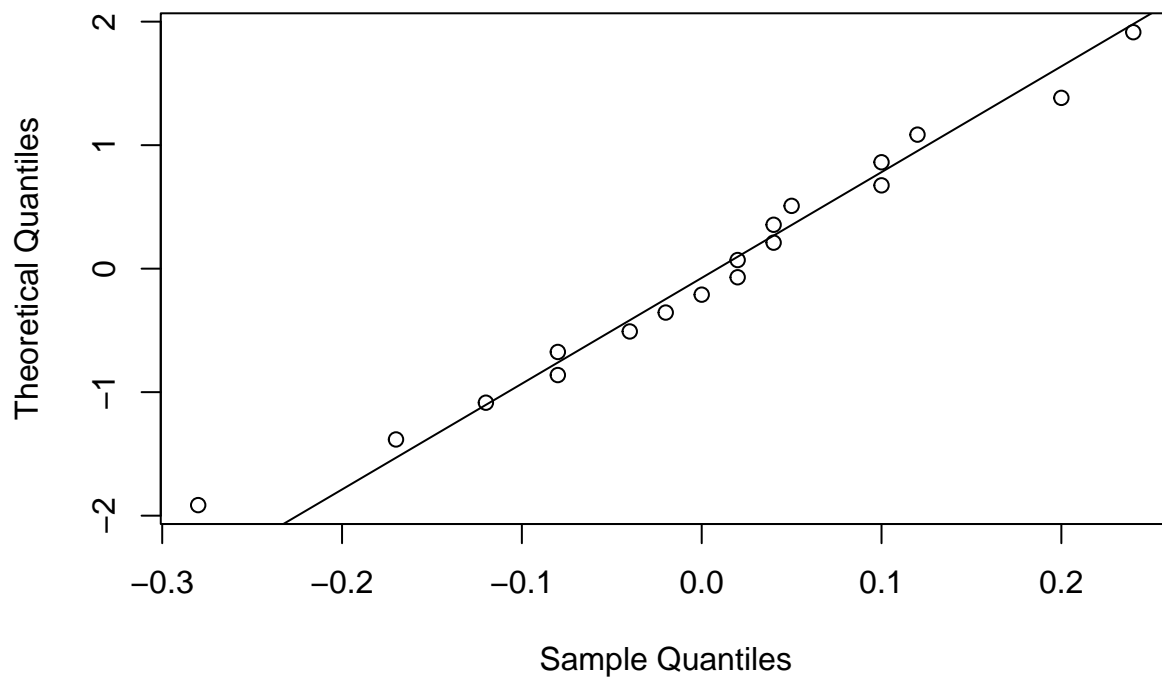
## Normal Q-Q Plot



And now water:

```
qqnorm(filter(mosqData, drink == "water")$change, datax = T)  
qqline(filter(mosqData, drink == "water")$change, datax = T)
```

## Normal Q-Q Plot



In the interest of showing you different ways to do things, here is another syntax that you can use to select

one of the samples:

```
qqnorm(mosqData$change[b$drink == "water"], datax = T)
## Error in qqnorm(mosqData$change[b$drink == "water"], datax = T): object 'b' not found
qqline(mosqData$change[b$drink == "water"], datax = T)
## Error in quantile(y, probs, names = FALSE, type = qtype, na.rm = TRUE): object 'b' not found
```

These graphs give you a qualitative view of whether or not the samples are normally distributed. In practice, the graphs are redundant and people generally use their favorite graph type to view their data.

## Formally testing for normality

Fortunately, there are tests that quantitatively test if a sample is consistent with a normal distribution, and the most common is the Shapiro-Wilk test (see Chap 13). This test calculates the mean and standard deviation of your sample, uses these estimates to derive a corresponding normal distribution, and then does a goodness-of-fit between the two.

Note that BOTH samples (beer and water) need to have a normal distribution!

The command to run the test is `shapiro.test(x)`, where `x` is a vector of numbers. The null hypothesis is that the sample has a normal distribution. Again, we can use `filter()` to test each sample separately.

```
shapiro.test(filter(mosqData, drink == "beer")$change)
##
##  Shapiro-Wilk normality test
##
## data:  filter(mosqData, drink == "beer")$change
## W = 0.96785, p-value = 0.5912
shapiro.test(filter(mosqData, drink == "water")$change)
##
##  Shapiro-Wilk normality test
##
## data:  filter(mosqData, drink == "water")$change
## W = 0.98162, p-value = 0.9652
```

The  $P$ -value in both tests is above an alpha of 0.05, and thus we fail to reject the null hypothesis that the samples have a normal distribution. We can thus proceed with the assumption of normality in both sample being fulfilled. In Chapter 13 we will discuss options if one or both samples does not have a normal distribution.

Note that we could also have used the `tapply()` command to do both Shapiro-Wilk tests with one line of code:

```
tapply(mosqData$change, mosqData$drink, shapiro.test)
## $beer
##
##  Shapiro-Wilk normality test
##
## data:  X[[i]]
## W = 0.96785, p-value = 0.5912
##
##
## $water
##
##  Shapiro-Wilk normality test
##
## data:  X[[i]]
## W = 0.98162, p-value = 0.9652
```

## Assumption of equal variances in the two samples

Another assumption of a two-sample  $t$ -test is that variance (or standard deviations) are equal in the two groups. The test is robust to some departure from this assumption, but there are limits.

First, if the sample sizes are above 30 and somewhat even then an up to 3x difference in standard deviation is allowed. In this case we know that the sample sizes are less than 30 in both groups. If wanted then you can quickly check the length of each sample with `tapply()`:

```
tapply(mosqData$change, mosqData$drink, length)
## beer water
## 25 18
```

With a low sample size study the textbook is subjective and says that the two-sample test should not be used of there is *some* difference in standard deviations.

First, let's calculate the standard deviation for the beer and water group separately with `tapply()`.

```
tapply(mosqData$change, mosqData$drink, sd)
## beer water
## 0.1622519 0.1269347
```

These standard deviations are fairly close (1.28x diff), but it's subject whether or not this qualifies as *some* difference.

We can, however, use the Levene's test to formally test if the beer and water groups have equal variance.

The test is done with `levineTest()`, which is part of the package "car". I think that this package was installed as a dependency previously. If not then you can use `install.packages("car")` to install it.

The syntax of `levineTest()` is three arguments. First you need to give the function the data frame with data. Next is a formula, which has a format of  $y \sim x$  where  $y$  is the numerical variable and  $x$  is the categorical treatment/group variable. Finally, you should specify that the "center" of each group be computed with the "mean" method. Thus:

```
library(car)
## Loading required package: carData
##
## Attaching package: 'car'
## The following object is masked from 'package:dplyr':
##
## recode
levineTest(data = mosqData, change ~ drink, center = "mean")
## Levene's Test for Homogeneity of Variance (center = "mean")
## Df F value Pr(>F)
## group 1 1.3689 0.2488
## 41
```

Note that the test can also be run with a syntax of `df$variable` for each variable in the formula. That is:

```
levineTest(mosqData$change ~ mosqData$drink, center = "mean")
## Levene's Test for Homogeneity of Variance (center = "mean")
## Df F value Pr(>F)
## group 1 1.3689 0.2488
## 41
```

Here the  $P$ -value is above 0.05, so we fail to reject the null hypothesis of equal variances and can proceed with the typical two-sample  $t$ -test. In Chapter 13 we will discuss options if the variances are not equal.



## Running a two-sample $t$ -test

The command to run a two-sample  $t$ -test in R is `t.test(x,y,var.equal=T)`, where `x` and `y` are the vectors of your two samples. In most cases, however, your samples will be found in the same column like we have here. An alternative syntax that efficiently separates the two samples is `t.test(numerical~categorical,data=dataframe,var.equal=T)`, where `numerical` is the name of the numerical variable/column and `categorical` is the name of the categorical variable/column. Thus, we use the following command:

```
t.test(change ~ drink, data = mosqData, var.equal = T)
##
## Two Sample t-test
##
## data: change by drink
## t = 3.1913, df = 41, p-value = 0.002717
## alternative hypothesis: true difference in means between group beer and group water is not equal to 0
## 95 percent confidence interval:
## 0.05383517 0.23940928
## sample estimates:
## mean in group beer mean in group water
## 0.154400000 0.007777778
```

Notice that the output shows that you are running a two-sample  $t$ -test. The  $P$ -value of the test is 0.002717, which is below the alpha of 0.05. Therefore, you reject the null hypothesis that the change in proportion of mosquitoes attracted is equal in the two groups.

The output of the test shows the mean change for each group, and you can see that the number is higher for the beer group. Thus, you can say that participants that had the beer treatment had significantly greater increases in proportion of mosquitoes attracted compared to those that had the water treatment.

## Chap12, Problem #8: mate preferences in hybrid cichlids

Problem #8 at the end of the chapter summarizes a study in which the researchers investigated female mate preferences in hybrids between closely related cichlid (fish) species. First, the species were crossed to produce F1 hybrids and the mate preference toward each of the male parental species was measured in female F1 hybrids. Then they crossed the F1 individuals to produce F2 hybrids and tested mate preferences again.

This study focuses on the question of whether a small or large number of genes are involved in mate preferences. If a small number of genes determine mate preferences then the variance in preference should be greater in F2 compared to F1 individuals.

The data are preference scores for F1 and F2 individuals tested. Since different fish were measured for each sample, they are independent. The question asks about differences in variance, but we can also run a two-sample  $t$ -test to see if the mean preference score is equal in the two groups.

## Import and inspect data

First, we should import the data and look at the data structure:

```
fishData <- read.csv("chap12q08Cichlids.csv", stringsAsFactors = T)
str(fishData)
## 'data.frame': 53 obs. of 2 variables:
## $ genotype : Factor w/ 2 levels "F1","F2": 1 1 1 1 1 1 1 1 1 1 ...
## $ preference: num 0.108 0.098 0.078 0.072 0.062 0.047 0.007 0.007 0.007 0.007 ...
```

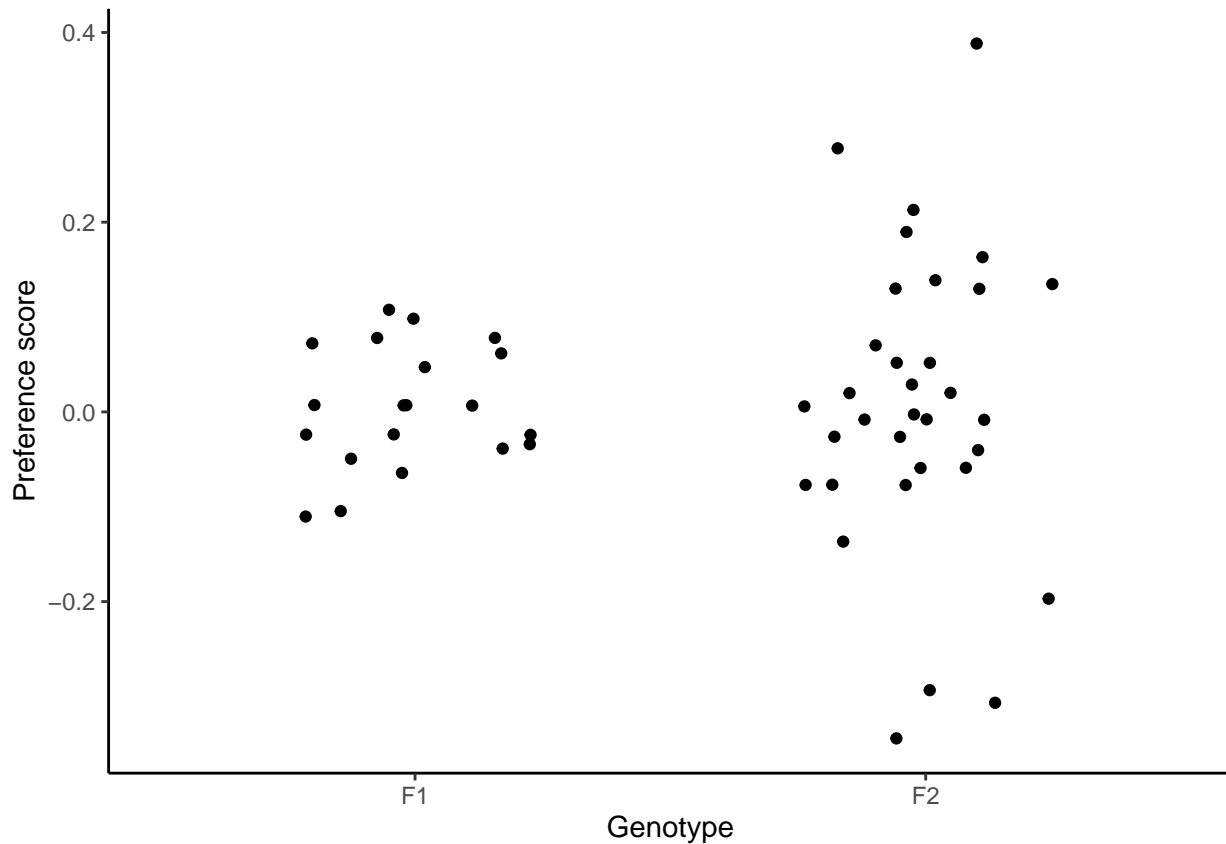
We can see that there are 53 rows and 2 columns. The first column (genotype) differentiates F1 and F2 samples, and the second column (preference) has the preference scores. See above for notes on data frame

organization differences in paired versus two-sample designs.

## Using graphs to evaluate normality

Let's again graph the data to see if each sample appears normal, but we'll keep it simple and only make a strip plot.

```
ggplot(fishData, aes(x = genotype, y = preference)) + geom_jitter(width = 0.25) +  
  xlab("Genotype") + ylab("Preference score") + theme_classic()
```



## Formally testing for normality

The plot shows that both samples appear consistent with normality, but let's again run the Shapiro-Wilk test (see Chap 13). This test calculates the mean and standard deviation of your sample, uses these estimates to derive a corresponding normal distribution, and then does a goodness-of-fit between the two.

The command to run the test is `shapiro.test(x)`, where `x` is a vector of numbers. The null hypothesis is that the sample has a normal distribution. Again, we could use `filter()` to test each sample separately.

```
shapiro.test(filter(fishData, genotype == "F1")$preference)  
##  
##  Shapiro-Wilk normality test  
##  
## data:  filter(fishData, genotype == "F1")$preference  
## W = 0.95417, p-value = 0.4348  
shapiro.test(filter(fishData, genotype == "F2")$preference)  
##  
##  Shapiro-Wilk normality test  
##
```

```
## data: filter(fishData, genotype == "F2")$preference
## W = 0.96735, p-value = 0.4108
```

Or we could run both tests in one command using `tapply()`.

```
tapply(fishData$preference, fishData$genotype, shapiro.test)
## $F1
##
## Shapiro-Wilk normality test
##
## data: X[[i]]
## W = 0.95417, p-value = 0.4348
##
##
## $F2
##
## Shapiro-Wilk normality test
##
## data: X[[i]]
## W = 0.96735, p-value = 0.4108
```

For both tests the  $P$ -value is above 0.05, so we can have met the assumptions of normality and can proceed.

## Assumption of equal variances in the two samples

Another assumption of a two-sample  $t$ -test is that variance (or standard deviations) are equal in the two groups. The test is robust to some departure from this assumption, but there are limits.

First, let's look at the sample size and standard deviation for each sample:

```
tapply(fishData$preference, fishData$genotype, length)
## F1 F2
## 20 33
tapply(fishData$preference, fishData$genotype, sd)
##      F1      F2
## 0.06424288 0.15820460
```

This shows that the sample sizes are 20 for F1 and 33 for F2. Since one is below 30 we should treat this as a small sample size study.

The amount of difference in standard deviations to allow is subjective when sample sizes are small, and the textbook says that the standard two-sample test should not be used if there is *some* difference between the groups. Here the difference is  $\sim 2.5x$ , which seems considerable. Let's run the Levene test to formally test the null of equal variances in the two groups.

```
leveneTest(data = fishData, preference ~ genotype, center = "mean")
## Levene's Test for Homogeneity of Variance (center = "mean")
##      Df F value Pr(>F)
## group 1  6.3182 0.01515 *
##      51
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Here the  $P$ -value is 0.01515, which is below 0.05. So we reject the null hypothesis of equal variances in the two groups.

So we meet the assumption of normality in both samples, but since the sample size is small and we also reject equal variances with the Levene's test. So we should not run a standard two-sample  $t$ -test on these data.

## Running a Welch's $t$ -test

Since the groups variances are not equal we should run Welch's  $t$ -test to analyze if the mean preference score is the same in both groups. To do this in R we run the `t.test()` command, but with the argument `equal.var=F`. This is how it can be run in R (see above for notes on syntax when numerical data for both samples are in the same column):

```
t.test(preference ~ genotype, data = fishData, var.equal = F)
##
##  Welch Two Sample t-test
##
## data:  preference by genotype
## t = -0.10468, df = 46.042, p-value = 0.9171
## alternative hypothesis: true difference in means between group F1 and group F2 is not equal to 0
## 95 percent confidence interval:
##  -0.06577309  0.05927006
## sample estimates:
## mean in group F1 mean in group F2
##      0.004900000      0.008151515
```

Notice that the output shows that you are running a Welch's  $t$ -test. The  $P$ -value of the test is 0.9171, which is above the alpha of 0.05. Therefore, you fail to reject the null hypothesis that the mean preference score is equal in F1 and F2 fish.

So in this data set there is a significant difference in variance in preference between the groups (F1 and F2), but the mean preferences are equal.

## R commands summary

- **Two-sample  $t$ -test (standard)**
  - `t.test(x,y,var.equal=T)`
- **Two-sample  $t$ -test (Welch's)**
  - `t.test(x,y,var.equal=F)`
- **Levene's test**
  - `leveneTest(y~x,center="mean")`