Chapter 9. Comparing Two Means

Daehyeog Lee

2023-11-12

Table of contents

9.1. What will this chapter tell me?

Comparing two means and checking for the evidence that one is different from the other.

9.2. Packages used in this chapter

ggplot2 is for graphs, pastecs is for descriptive statistics, and WRS2 is for robust methods.

```
library(ggplot2); library(pastecs); library(WRS2)
library(tidyr); library(dplyr); library(boot)
```

```
Attaching package: 'tidyr'

The following object is masked from 'package:pastecs':

    extract

Attaching package: 'dplyr'

The following objects are masked from 'package:pastecs':

    first, last

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

9.3. Looking at differences

There are two different ways of collecting data: we can either expose different people to different experimental manipulation (between group or independent design), or take a single group of people and expose them to different experimental manipulations at different points of time (repeated-measures design).

9.3.1. A problem with error bar graphs of repeated-measures designs

SELF-TEST

24 arachnophobes were used in this data.

```
Group <- gl(2, 12, labels = c( "Picture", "Real Spider"))
Anxiety <- c(30, 35, 45, 40, 50, 35, 55, 25, 30, 45, 40, 50, 40, 35, 50, 55,
             65, 55, 50, 35, 30, 50, 60, 39)
spiderLong <- data.frame(Group, Anxiety)
head(spiderLong)
```

```
   Group Anxiety
1 Picture      30
2 Picture      35
3 Picture      45
4 Picture      40
5 Picture      50
6 Picture      35
```

```
bar <- ggplot(spiderLong, aes(Group, Anxiety))
bar +
  stat_summary(fun.y = mean, geom = "bar", fill = "White", colour = "Black") +
  stat_summary(
    fun.data = mean_cl_normal, geom = "pointrange"
    ) +
  labs(
    x = "Type of Stimulus", y = "Anxiety"
    ) +
  scale_y_continuous(
    limits = c(0, 70), breaks = seq(from = 0, to = 70, by = 10)
    )
```

Warning: The `fun.y` argument of `stat_summary()` is deprecated as of ggplot2 3.3.0.
i Please use the `fun` argument instead.
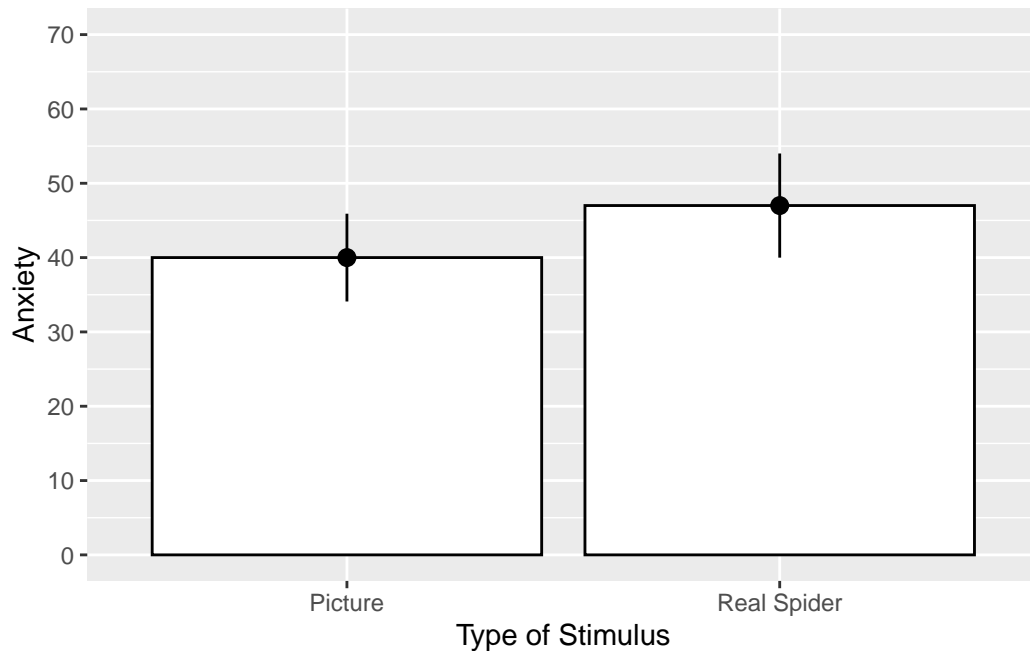
SELF-TEST

12 same participants' data were used. All that has changed is whether the design used the same participants (repeated measures) or different (independent). However, two graphs are same.

If we have repeated measures groups, we should not use error bar graphs. Or if we do, we have to adjust the data before we plot the graph:

```
spiderWide <- read.delim("spiderWide.dat", header = TRUE)
head(spiderWide)
```

```
  picture real
1      30   40
2      35   35
3      45   50
4      40   55
5      50   65
6      35   55
```

9.3.2. Step 1: calculate the mean for each participant

To begin with, we need to calculate the average anxiety for each participant by executing:

```
spiderWide$pMean <- (spiderWide$picture + spiderWide$real) / 2
spiderWide$pMean
```

```
 [1] 35.0 35.0 47.5 47.5 57.5 45.0 52.5 30.0 30.0 47.5 50.0 44.5
```

9.3.3. Step 2: calculate the grand mean

A grand mean is the mean of all scores and so for the current data this value will be the mean of all 24 scores:

```
grandMean <- mean(c(spiderWide$picture, spiderWide$real))
grandMean
```

```
[1] 43.5
```

Executing this command creates a variable called **grandMean**, which is the mean of **picture** and **real** combined into a single variable. In other words, it's the mean of all scores.

9.3.4. Step 3: calculate the adjustment factor

The fact that participants' mean anxiety scores differ represents individual differences between different people. We should equalize the means between participants. To do this, we need to calculate an adjustment factor by substracting each participant's mean score (**pMean**) from the grand mean (**grandMean**):

```
spiderWide$adj <- grandMean - spiderWide$pMean
```

The scores in column **adj** represent the difference between each participant's mean anxiety and the mean anxiety level across all participants.

5

9.3.5. Step 4: create adjusted values for each variable

So far, we have calculated the difference between each participant's mean score and the mean score of all participants (the grand mean). This difference can be used to adjust the existing scores for each participant:

```
spiderWide$picture_adj <- spiderWide$picture + spiderWide$adj
spiderWide$real_adj <- spiderWide$real + spiderWide$adj
head(spiderWide)
```

```
  picture real pMean   adj picture_adj real_adj
1      30   40  35.0   8.5        38.5     48.5
2      35   35  35.0   8.5        43.5     43.5
3      45   50  47.5  -4.0        41.0     46.0
4      40   55  47.5  -4.0        36.0     51.0
5      50   65  57.5 -14.0        36.0     51.0
6      35   55  45.0  -1.5        33.5     53.5
```

```
pMean3 <- mean(c(spiderWide$real_adj, spiderWide$picture_adj))
pMean3 # Same as the grand mean!
```

```
[1] 43.5
```

SELF-TEST

```
spiderWide$pMean <-
  c(35.0, 35.0, 47.5, 47.5, 57.5, 45.0, 52.5, 30.0, 30.0, 47.5, 50.0, 44.5)
spiderWide$adj <-
  c(8.5, 8.5, -4.0, -4.0, -14.0, -1.5, -9.0, 13.5, 13.5, -4.0, -6.5, -1.0)
spiderWide$picture_adj <-
  c(38.5, 43.5, 41.0, 36.0, 36.0, 33.5, 46.0, 38.5, 43.5, 41.0, 33.5, 49.0)
spiderWide$real_adj <-
  c(48.5, 43.5, 46.0, 51.0, 51.0, 53.5, 41.0, 48.5, 43.5, 46.0, 53.5, 38.0)
spiderWide$pMean2 <-
  c(43.5, 43.5, 43.5, 43.5, 43.5, 43.5, 43.5, 43.5, 43.5, 43.5, 43.5, 43.5)
spiderWide$id <- gl(12, 1, labels = c(paste("P", 1:12, sep = "_")))
head(spiderWide)
```

```
  picture real pMean   adj picture_adj real_adj pMean2  id
1      30   40  35.0   8.5        38.5     48.5   43.5 P_1
2      35   35  35.0   8.5        43.5     43.5   43.5 P_2
3      45   50  47.5  -4.0        41.0     46.0   43.5 P_3
4      40   55  47.5  -4.0        36.0     51.0   43.5 P_4
5      50   65  57.5 -14.0        36.0     51.0   43.5 P_5
6      35   55  45.0  -1.5        33.5     53.5   43.5 P_6
```

```r
adjustedData <- spiderWide %>%
  pivot_longer(
    cols = c("picture_adj", "real_adj"),
    names_to = "variable",
    values_to = "value"
  )

head(adjustedData)
```

```
# A tibble: 6 x 8
  picture  real pMean   adj pMean2 id    variable    value
    <int> <int> <dbl> <dbl>  <dbl> <fct> <chr>       <dbl>
1      30    40    35   8.5   43.5 P_1   picture_adj  38.5
2      30    40    35   8.5   43.5 P_1   real_adj     48.5
3      35    35    35   8.5   43.5 P_2   picture_adj  43.5
4      35    35    35   8.5   43.5 P_2   real_adj     43.5
5      45    50  47.5  -4     43.5 P_3   picture_adj  41
6      45    50  47.5  -4     43.5 P_3   real_adj     46
```

```r
adjustedData<-adjustedData[, -c(1:5)]
head(adjustedData)
```

```
# A tibble: 6 x 3
  id    variable    value
  <fct> <chr>       <dbl>
1 P_1   picture_adj  38.5
2 P_1   real_adj     48.5
3 P_2   picture_adj  43.5
4 P_2   real_adj     43.5
5 P_3   picture_adj  41
6 P_3   real_adj     46
```

```r
names(adjustedData)<-c("id", "Group", "Anxiety_Adj")
adjustedData$Group <-
  factor(adjustedData$Group, labels = c("Spider Picture", "Real Spider"))
head(adjustedData)
```

```
# A tibble: 6 x 3
  id    Group          Anxiety_Adj
  <fct> <fct>                <dbl>
1 P_1   Spider Picture        38.5
2 P_1   Real Spider           48.5
3 P_2   Spider Picture        43.5
4 P_2   Real Spider           43.5
5 P_3   Spider Picture        41
6 P_3   Real Spider           46
```

```r
bar <- ggplot(adjustedData, aes(Group, Anxiety_Adj))
bar +
  stat_summary(fun.y = mean, geom = "bar", fill = "White", colour = "Black") +
  stat_summary(fun.data = mean_cl_normal, geom = "pointrange") +
  labs(x = "Type of Stimulus", y = "Anxiety") +
  scale_y_continuous(
    limits = c(0, 70), breaks = seq(from = 0, to = 70, by = 10)
    )
```
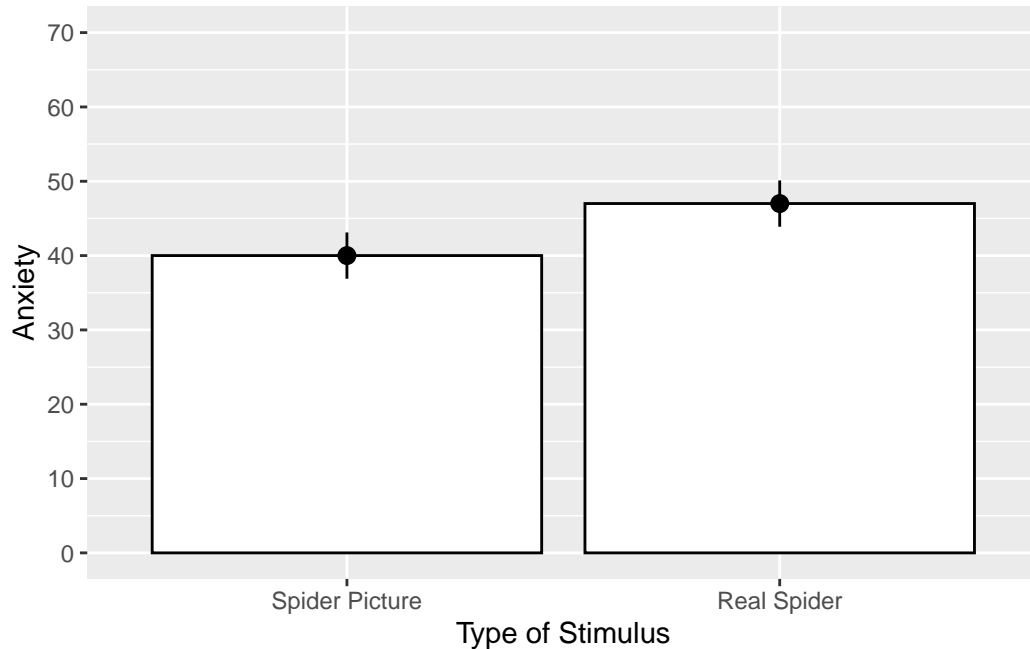
9.4. The *t*-test

t-test is used in various situations such as:
testing whether a correlation coefficient is different from 0; testing whether a regression coefficient, *b*, is different from 0; testing whether two group means are different.

There are two different *t*-tests depending on whether the independent variable was manipulated using the same participants or different:

- **Independent-means *t*-test (Independent measures)** is used when there are two experimental conditions and different participants were assigned to each condition.

- **Dependent-means *t*-test (Paired-samples *t*-test)** is used when there are two experimental conditions and the same participants took part in both conditions of the experiment.

9.4.1. Rationale for the *t*-test

Both *t*-tests have a similar rationale.

- Two samples of data are collected and the sample means calculated.

- If the samples come from the same population, then we expect their means to be roughly equal. Under the null hypothesis we assume that the experimental manipulation has no effect on the participants. Therefore, we expect the sample means to be very similar.

- We compare the difference between the sample means that we collected to the difference between the sample means that we would expect to obtain if there were no effect. If the difference between the samples we have collected is larger than we would expect then we can assume one of the two things:

  - We have, by chance, collected two samples that are atypical of the population from which they came.

  - The two samples means differ because of the different experimental manipulation imposed on each sample.

Test statistics can be thought as the 'variance explained by the model' divided by the 'variance that the model can't explain'. In other words, 'effect/error'. We can use the standard error of the differences between the two means as an estimate of the error in our model.

$$t = \frac{observed\,difference\,b/w\,sample\,means - expected\,difference\,b/w\,population\,means}{estimate\,of\,the\,SE\,of\,the\,difference\,b/w\,two\,sample\,means}$$

The exact form that this equation takes depends on whether the same or different participants were used in each experimental condition.

9.4.2. The *t*-test as a general linear model

All statistical procedures have simple model: outcome = (model) + error

$$A_i = (b_0 + b_1 G_i) + \epsilon_i$$

b_0 (the intercept) is equal to the mean of the baseline group. b_1 represents the difference between the group means (put 0 and 1 to G_i). In regression, the *t*-test is used to ascertain whether the regression coefficient b_1 is equal to 0, and when we carry out a *t*-test on grouped data, we test whether the difference between group means is equal to 0.

SELF-TEST

```
lmspider <- lm(spiderLong$Anxiety ~ spiderLong$Group)
summary(lmspider)
```

```
Call:
lm(formula = spiderLong$Anxiety ~ spiderLong$Group)

Residuals:
```

```
    Min     1Q Median     3Q     Max
  -17.0    -8.5    1.5     8.0    18.0
```

Coefficients:
```
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)                 40.000      2.944  13.587 3.53e-12 ***
spiderLong$GroupReal Spider  7.000      4.163   1.681    0.107
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 10.2 on 22 degrees of freedom
Multiple R-squared:  0.1139,     Adjusted R-squared:  0.07359
F-statistic: 2.827 on 1 and 22 DF,   p-value: 0.1068
```

```r
mean_picture <- mean(c(30, 35, 45, 40, 50, 35, 55, 25, 30, 45, 40, 50))
mean_picture
```

```
[1] 40
```

```r
mean_spider <- mean(c(40, 35, 50, 55, 65, 55, 50, 35, 30, 50, 60, 39))
mean_spider
```

```
[1] 47
```

```r
mean_spider - mean_picture
```

```
[1] 7
```

The first thing to notice is the value of the constant b_0: its value is 40, the same as the mean of the base category (the picture group). The second thing to notice is that the value of the regression coefficient is 7, which is the difference between the two group means (`mean_spider - mean_picture`). Finally, the t-statistic, which tests whether b_1 is significantly different from zero, is not significant, indicating that b_1 is not significantly different from zero.

9.4.3. Assumptions of the *t*-test

Given that the *t*-test is basically regression, it has much the same assumptions. Both the independent
t-test and the dependent *t*-test are parametric tests based on the normal distribution. Therefore, they assume:

- The sampling distribution is normally distributed. In the dependent *t*-test this means that the sampling distribution of the *differences* between scores should be normal, not the scores themselves.

The independent *t*-test, because it is used to test different groups of people, also assumes:

- Scores in different treatment conditions are independent.

- Homogeneity of variance. However, statisticians recently have stopped using this approach for three reasons. First, if we don't have unequal group sizes, the assumption is pretty much irrelevant and can be ignored. Second, the tests of homogeneity of variance don't work with unequal group sizes and smaller samples. Third, there is an adjustment called Welch's *t*-test, which is able to correct for violation of this assumption.

9.5. The independent *t*-test

9.5.1. The independent *t*-test equation explained

We are dealing with the situation in which different entities have been tested in the different conditions of our experiment. When different participants participate in different conditions, pairs of scores will differ not just because of the experimental manipulation, but also because of other sources of variance. Therefore, we have to make comparisons on a *per condition* basis.

For the independent *t*-test we are looking at differences between groups and so we divide by the standard deviation of differences between groups. The majority of samples from a population will have fairly similar means. If we plot a sampling distribution of the differences between every pair of sample means that could be taken from two populations, we would find that it had a normal distribution with a mean equal to the difference between population means $(\mu_1 - \mu_2)$. The SD of the sampling distribution (the standard error) tells us how variable the differences between sample means can occur by chance. We divide the difference between sample means by the SD of the sampling distribution. The **variance sum law** states that the variance of a difference between two independent variables is equal to the sum of their variances. To restate, the variance of the sampling distribution is equal to the sum of the variances of the two populations from which the samples were taken.

(Equations are in p.374~375)

The pooled variance estimate t-test is used, which takes account of the difference in sample size by *weighting* the variance of each sample. The pooled variance estimate is simply a weighted average in which each variance is multiplied (weighted) by its degrees of freedom, and then we divide by the sum of weights.

We can compare the obtained value of t against the maximum value we would expect to get by chance alone in a t-distribution with the same degrees of freedom. If the value we obtain exceeds this critical value we can be confident that this reflects an effect of our independent variable.

R's Souls' Tip

```
# Defining a function which calculates the t-statistic
x1 <- mean(spiderLong[spiderLong$Group == "Real Spider", ]$Anxiety)
x2 <- mean(spiderLong[spiderLong$Group == "Picture", ]$Anxiety)
sd1 <- sd(spiderLong[spiderLong$Group == "Real Spider", ]$Anxiety)
sd2 <- sd(spiderLong[spiderLong$Group == "Picture", ]$Anxiety)
n1 <- length(spiderLong[spiderLong$Group == "Real Spider", ]$Anxiety)
n2 <- length(spiderLong[spiderLong$Group == "Picture", ]$Anxiety)

ttestfromMeans<-function(x1, x2, sd1, sd2, n1, n2)
{
  df <- n1 + n2 - 2
  poolvar <- (((n1 - 1) * sd1^2) + ((n2 - 1) * sd2^2)) / df
  t <- (x1 - x2) / sqrt(poolvar * ((1 / n1) + (1 / n2)))
  sig <- 2*(1 - (pt(abs(t), df)))
  paste("t(df = ", df, ") = ", t, ", p = ", sig, sep = "")
}

ttestfromMeans(x1, x2, sd1, sd2, n1, n2)
```

```
[1] "t(df = 22) = 1.68134561495341, p = 0.106839192382597"
```

9.5.2. Doing the independent t-test

9.5.2.1. General procedure for the independent t-test

This is a general procedure of the independent t-test.

1. Enter data.
2. Explore data by graphing or computing some descriptive statistics. Check distributional assumptions.

13

3. Run the *t*-test.
4. Calculate an effect size.

9.5.2.2. Entering data

The `t.test()` function contains an option `paired = TRUE/FALSE`, which tells it whether to treat data as dependent or independent. However, R Commander does care, so we will enter the data in a long format.

```
Group <- gl(2, 12, labels = c("Picture", "Real Spider"))
Anxiety <- c(30, 35, 45, 40, 50, 35, 55, 25, 30, 45, 40, 50, 40, 35, 50, 55,
             65, 55, 50, 35, 30, 50, 60, 39)
spiderLong <- data.frame(Group, Anxiety)
```

9.5.2.3. The independent *t*-test using R Commander

Import data -> Statistics -> Means -> Independent samples *t*-test.

### 9.5.2.4. Exploring data and testing assumptions

```
spiderBoxplot <- ggplot(spiderLong, aes(Group, Anxiety))
spiderBoxplot +
  geom_boxplot() +
  labs(x = "Type of Stimulus", y = "Anxiety") +
  scale_y_continuous(
    limits = c(0, 100), breaks = seq(from = 0, to = 100, by = 10)
    )
```
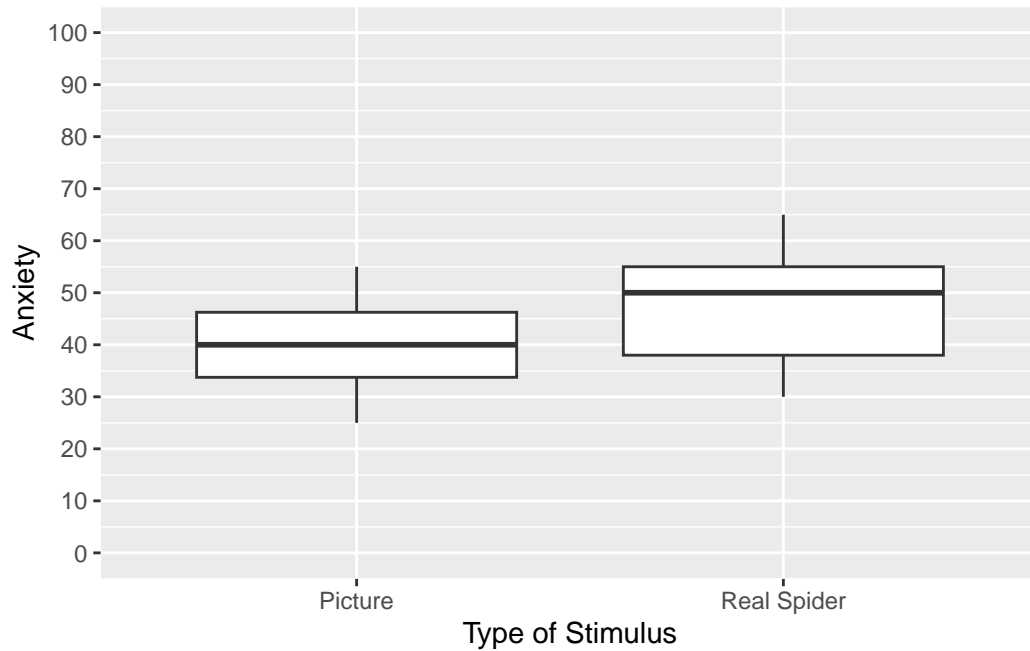
14

```
bar <- ggplot(spiderLong, aes(Group, Anxiety))
bar +
  stat_summary(fun.y = mean, geom = "bar", fill = "White", colour = "Black") +
  stat_summary(fun.data = mean_cl_normal, geom = "pointrange") +
  labs(x = "Type of Stimulus", y = "Anxiety") +
  scale_y_continuous(
    limits = c(0, 60), breaks = seq(from = 0, to = 60, by = 10)
    )
```

Warning: Removed 1 rows containing non-finite values (`stat_summary()`).
Removed 1 rows containing non-finite values (`stat_summary()`).
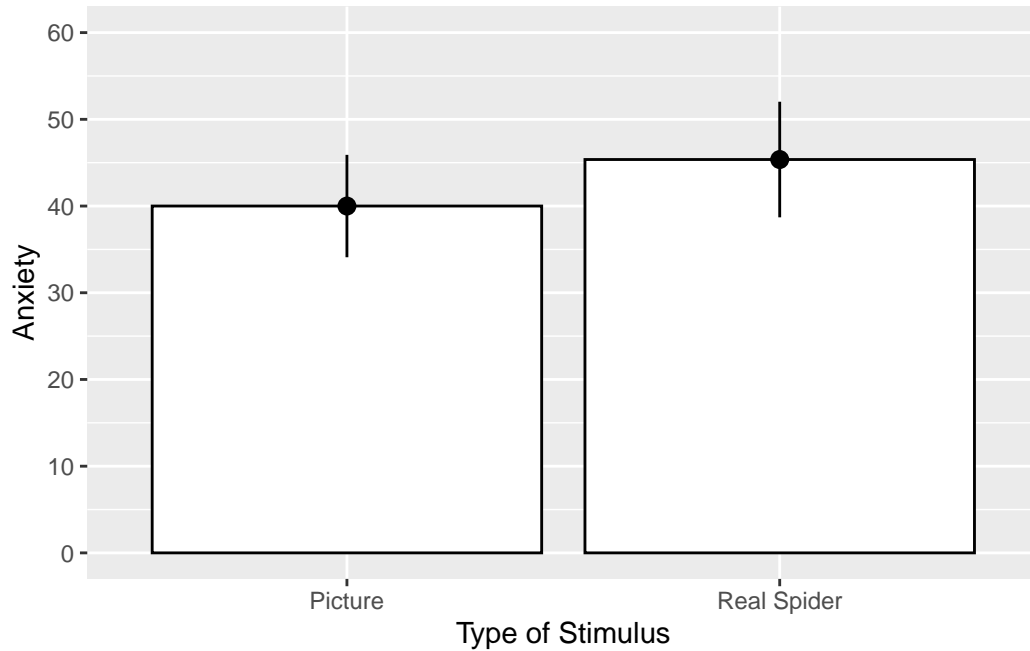
On face value, there are no between-group differences. To get some descriptive statistics for each group we can use the `by()` function:

```
by(spiderLong$Anxiety, spiderLong$Group, stat.desc, basic = FALSE, norm = TRUE)
```

```
spiderLong$Group: Picture
      median          mean      SE.mean CI.mean.0.95          var       std.dev
  40.0000000    40.0000000    2.6827168    5.9046200   86.3636364     9.2932038
    coef.var      skewness     skew.2SE     kurtosis     kurt.2SE   normtest.W
   0.2323301     0.0000000    0.0000000   -1.3939289   -0.5656047    0.9650165
   normtest.p
   0.8522870
-------------------------------------------------------------
spiderLong$Group: Real Spider
       median           mean       SE.mean  CI.mean.0.95               var
  50.000000000    47.000000000    3.183765638    7.007420922 121.636363636
      std.dev       coef.var       skewness      skew.2SE       kurtosis
  11.028887688    0.234657185   -0.005590699   -0.004386224   -1.459758279
     kurt.2SE      normtest.W     normtest.p
  -0.592315868    0.948872904    0.620569431
```

Both normality tests are non-significant, implying that we can probably assume normality of

errors in the model.

9.5.2.5. The independent *t*-test using R

```
ind.t.test <- t.test(Anxiety ~ Group, data = spiderLong)
ind.t.test
```

```
	Welch Two Sample t-test

data:  Anxiety by Group
t = -1.6813, df = 21.385, p-value = 0.1072
alternative hypothesis: true difference in means between group Picture and group Real Spider
95 percent confidence interval:
 -15.648641    1.648641
sample estimates:
    mean in group Picture mean in group Real Spider
                       40                        47
```

The above code created a model called `ind.t.test` based on predicting anxiety scores (**Anxiety**) from group membership (**Group**).

Alternatively, if we use the data `spiderWide` as an input, we would need to run the *t*-test by executing:

```
ind.t.test <- t.test(spiderWide$real, spiderWide$picture)
ind.t.test
```

```
	Welch Two Sample t-test

data:  spiderWide$real and spiderWide$picture
t = 1.6813, df = 21.385, p-value = 0.1072
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -1.648641 15.648641
sample estimates:
mean of x mean of y
       47        40
```

9.5.2.6. Output from the independent t-test

The degrees of freedom are calculated by adding the two sample sizes and then subtracting the number of samples. However, the output reports 21.39. This discrepancy is because this function uses a **Welch's t-test**, which does not make the assumption of homogeneity of variance. The Welch uses a correction which adjusts the degrees of freedom based on the homogeneity of variance.

9.5.2.7. Robust methods to compare independent means

Wilcox (2005) describes some robust procedures for comparing two means from independent groups.

The first robust function, `yuen()`, is based on a trimmed mean. It takes the general form: `yuen(scores group 1, scores group 2, tr = .2, alpha = .05)`, in which:

- `scores group 1(2)` is a variable that contains the scores for the first(second) group.

- `tr` is the proportion of trimming to be done. The default is 20%.

- `alpha` sets the alpha level for the test.

```
any(is.na(spiderWide$real))
```

```
[1] FALSE
```

```
any(is.na(spiderWide$picture))
```

```
[1] FALSE
```

```
class(spiderWide$real)
```

```
[1] "integer"
```

```
class(spiderWide$picture)
```

```
[1] "integer"
```

```
sd(spiderWide$real)
```

[1] 11.02889

```
sd(spiderWide$picture)
```

[1] 9.293204

```
# yuen(formula = real ~ picture, tr = 0.1, data = spiderWide)

# yuen(spiderWide$real ~ spiderWide$picture)
# yuen(spiderWide$real ~ spiderWide$picture, tr = .1)
# yuenbt(spiderWide$real, spiderWide$picture, nboot = 2000)
# pb2gen(spiderWide$real, spiderWide$picture, alpha = .05, nboot = 2000, est = mom)
# pb2gen(spiderWide$real, spiderWide$picture, nboot = 2000)
```

9.5.2.8. Calculating the effect size

$r = \sqrt{\frac{t^2}{t^2+df}}$ converts a t-value into an r-value. Even though our t-statistic is statistically significant, this doesn't mean our effect is important in practical terms. We can also calculate r using **R** by executing:

```
t <- ind.t.test$statistic[[1]]
t
```

[1] 1.681346

```
df <- ind.t.test$parameter[[1]]
df
```

[1] 21.38502

```
r <- sqrt(t^2 / (t^2 + df))
r
```

[1] 0.3416978

```
round(r, 3)
```

[1] 0.342

The '[[1]]' tells **R** that we want the first value only.

9.5.2.9. Reporting the independent *t*-test

- On average, participants experienced great anxiety from real spiders (M = 47.00, SE = 3.18), than from pictures of spiders (M = 40.00, SE = 2.68). This difference was not significant t(21.39) = -1.68, p > .05; however, it did represent a medium-sized effect r = .34

9.6. The dependent *t*-test

$t = \frac{D - \mu_D}{\frac{s_D}{\sqrt{N}}}$ D : The mean difference between our samples

μ_D : The difference that we would expect to find between population means
Denominator: The SE of the differences

This equation compares he mean difference between our samples to the difference that we would expect to find between population means, and then takes into account the standard error of the differences. If the null hypothesis is true, then we expect there to be no difference between the population means.

9.6.1. Sampling distributions and the standard error

The standard error is the standard deviation of the sampling distribution.

Three properties of sampling distribution:

- If the population is normally distributed, the sampling distribution is normally distributed.

- The mean of the sampling distribution is equal to the mean of the population.

- The SD of sampling distribution is equal to the standard deviation of the population divided by the square root of the number of observations in the sample. This SD is known as the standard error.

Most of the time, the difference between sample means from the same population will be zero, or close to zero. The **standard error of differences** is the SD of the sampling distribution. A small SE tells us that most pairs of samples from a population will have very similar means. A large SE tells us that sample means can deviate quite a lot from the population mean.

9.6.2. The dependent *t*-test equation explained

The average difference is calculated by adding the differences between each person's score in each condition (total amount of difference) and dividing this by the number of participants. This average difference is an indicator of the systematic variation in the data, which represents the experimental effect. We need to compare this systematic variation against some kind of measure of the 'systematic variation that we could naturally expect to find'.

The SE is a measure of the error in the mean as a model of the population. If we had taken two random samples from a population, then the means could be different just by chance. The SE tells us by how much these samples could differ. Our model is the average difference between condition means, and we divide by the SE which represents the error associated with this model.

By dividing by the SE we are doing two things:

- Standardizing the average difference between conditions.
- Contrasting the difference between means that we have against the difference that we could *expect* to get based on how well the samples represent the populations from which they came.

In a perfect world, we could calculate the SE by taking all possible pairs of samples from a population, but it's impossible. Therefore, we estimate the standard error from the SD of differences obtained within the sample s_D and the sample size N.

If the SE of differences is a measure of the unsystematic variation within the data, and the sum of difference scores represents the systematic variation, then it should be clear that the *t*-statistic is simply the ratio of the systematic variation in the experiment to the unsystematic variation.

- Unsystematic variation: Difference created by unknown factors.
- Systematic variation: Difference created by a specific experimental manipulation.

If the experimental manipulation creates any kind of effect, then we would expect the systematic variation to be much greater than the unsystematic variation (therefore, $t > 1$). We can compare the obtained value of t against the maximum value we would expect to get by chance alone in a t-distribution with the same degrees of freedom. If the value we obtain exceeds this critical value, we can be confident that this reflects an effect of our independent variable.

9.6.3. Dependent *t*-tests using R

9.6.3.1. General procedure for the dependent *t*-test

- Enter data.
- Graph the data, compute some descriptive statistics, check distributional assumptions.
- Run the *t*-test.
- Calculate an effect size.

9.6.3.2. Entering data

```
picture <- c(30, 35, 45, 40, 50, 35, 55, 25, 30, 45, 40, 50)
real <- c(40, 35, 50, 55, 65, 55, 50, 35, 30, 50, 60 ,39)
spiderWide <- data.frame(picture, real)
spiderWide
```

```
   picture real
1       30   40
2       35   35
3       45   50
4       40   55
5       50   65
6       35   55
7       55   50
8       25   35
9       30   30
10      45   50
11      40   60
12      50   39
```

9.6.3.3. The dependent *t*-test using R Commander

Statistics -> Means -> Paired *t*-test

9.6.3.4. Exploring data and testing assumptions

By executing the following command, we can get descriptive statistics:

```
stat.desc(spiderWide, basic = FALSE, norm = TRUE)
```

```
                  picture          real
median         40.0000000   50.000000000
mean           40.0000000   47.000000000
SE.mean         2.6827168    3.183765638
CI.mean.0.95    5.9046200    7.007420922
var            86.3636364  121.636363636
std.dev         9.2932038   11.028887688
coef.var        0.2323301    0.234657185
skewness        0.0000000   -0.005590699
skew.2SE        0.0000000   -0.004386224
kurtosis       -1.3939289   -1.459758279
kurt.2SE       -0.5656047   -0.592315868
normtest.W      0.9650165    0.948872904
normtest.p      0.8522870    0.620569431
```

Parametric tests assume that the sampling distribution is normal. This should be true in large samples, but in small samples we often check the normality of our data:

```
spiderWide$diff<-spiderWide$real-spiderWide$picture
spiderWide
```

```
     picture real diff
1         30   40   10
2         35   35    0
3         45   50    5
4         40   55   15
5         50   65   15
6         35   55   20
7         55   50   -5
8         25   35   10
9         30   30    0
10        45   50    5
11        40   60   20
12        50   39  -11
```

```
stat.desc(spiderWide$diff, basic = FALSE, desc = FALSE, norm = TRUE)
```

```
  skewness    skew.2SE    kurtosis    kurt.2SE normtest.W normtest.p
-0.2464810 -0.1933785 -1.2342159 -0.5007991  0.9557903  0.7224801
```

The output shows that the distribution of differences is not significantly different form normal, W = 0.956, p > .05.

9.6.3.5. The dependent *t*-test using R

We again use the function `t.test()`, but this time we include the option `paired = TRUE`:

```
dep.t.test <- t.test(spiderWide$real, spiderWide$picture, paired = TRUE)
dep.t.test
```

```
	Paired t-test

data:  spiderWide$real and spiderWide$picture
t = 2.4725, df = 11, p-value = 0.03098
alternative hypothesis: true mean difference is not equal to 0
95 percent confidence interval:
  0.7687815 13.2312185
sample estimates:
mean difference
              7
```

If we had our data stored in long format so that our group scores are in a single column and group membership is expressed in a second column, we can still run a dependent *t*-test by executing:

```
dep.t.test <- t.test(Anxiety ~ Group, data = spiderLong, paired = TRUE)
dep.t.test
```

```
	Paired t-test

data:  Anxiety by Group
t = -2.4725, df = 11, p-value = 0.03098
alternative hypothesis: true mean difference is not equal to 0
95 percent confidence interval:
```

```
 -13.2312185  -0.7687815
sample estimates:
mean difference
            -7
```

9.6.3.6. Output from the dependent *t*-test

The probability for the spider data is very low (p = .031) and in fact it tells us that there is only a 3.1% chance that a value of t this big could happen if the null hypothesis were true. Therefore, the result
$t(11) = 2.47$, p < .05 is significant.

The output also provides a 95% confidence interval **for the mean difference**. The confidence interval tells us the boundaries within which the true mean difference is likely to lie. **So, assuming this sample's confidence interval is one of the 95 out of 100 that contains the population value, we can say that the true mean difference lies between 0.77 and 13.23.** The importance of this interval is that it does not contain zero (both limits are positive) because this tells us that the true value of the mean difference is unlikely to be zero. If we were to compare pairs of random samples from a population, we would expect most of the differences between sample means to be zero. But since our interval, based on our two samples, does not contain zero, we can be confident that our two samples do not represent random samples from the same population. Instead, they represent samples from different populations induced by the experimental manipulation.

9.6.3.7. Robust methods to compare dependent means

```
# yuend(spiderWide$real, spiderWide$picture)
# ydbt(spiderWide$real, spiderWide$picture, nboot = 2000)
# bootdcpi(spiderWide$real, spiderWide$picture, est = tmean, nboot = 2000)
```

9.6.3.8. Calculating the effect size

```
t <- dep.t.test$statistic[[1]]
df <- dep.t.test$parameter[[1]]
r <- sqrt(t^2 / (t^2 + df))
round(r, 3)
```

```
[1] 0.598
```

The r is calculated as 0.598, which indicates that the effect is large.

9.6.3.9. Reporting the dependent *t*-test

- On average, experienced significantly greater anxiety from real spiders (M = 47.00, SE = 3.18) than from pictures of spiders (M = 40.00, SE = 2.68), t(11) = 2.47, p < .05, r = .60.

9.7. Between groups or repeated measures?

There was a significant difference between means when the data was analysed from the same participants, but the difference between means was not significant when the data was analysed from different participants. When the same participants are used across conditions the unsystematic variance (error variance) is reduced dramatically, making it easier to detect any systematic variance.