Practicing T-test in R

Part 2: 2-sample and paired t-tests

Nathan Brouwer\T1\textbar{}brouwern@gmail.com/@lobrowR

October 2017

Outline

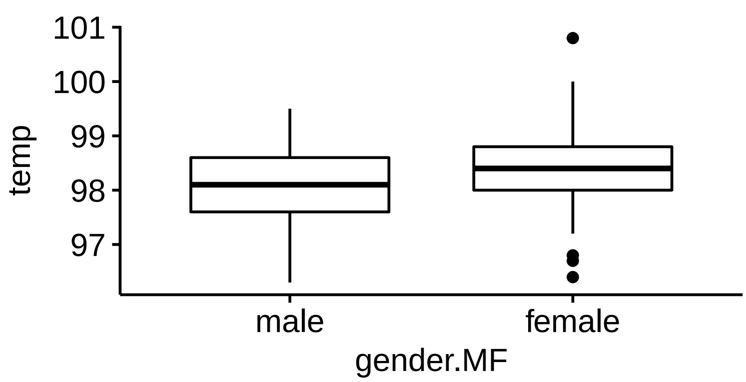
- Part 2: 2-sample t-test & paired t-test
- Question 2: Is the mean human body statistically different from 98.6?
- Question 3: How do Male & Female Body Temps compare?
- Question 4: How is a paired t-test similar to a 1-sample t-test?
- OPTIONAL: Question 5: What is the relationship between resting heart rate & body temperature

Question 3: How do Male and Female Body Temps compare?

Graph 3: How do males and female compare?

An excellent way to compare data that is in two groups (eg male and female) is with a boxplot

```
#library(ggpubr)
ggboxplot(data = bodytemp,
          y = "temp",
          x = "gender.MF")
```



We can code by the color of the lines like this using color =

```
ggboxplot(data = bodytemp,
          y = "temp",
          x = "gender.MF",
          color = "gender.MF")
```

And we can change the fill using fill (note, only do color or fill, not both)

```
ggboxplot(data = bodytemp,
          y = "temp",
          x = "gender.MF",
          fill = "gender.MF")
```

We can improve the labels using xlab and ylab

```r
ggboxplot(data = bodytemp,
          y = "temp",
          x = "gender.MF",
          fill = "gender.MF",
          xlab = "Gender",
          ylab = "Body temperature (F)")
```

Boxplots tell us a lot about the data, but they can be further improved by overlaying the raw data. We can do this in ggpubr using add = "jitter"

```r
ggboxplot(data = bodytemp,
          y = "temp",
          x = "gender.MF",
          fill = "gender.MF",
          xlab = "Gender",
          ylab = "Body temperature (F)",
          add = "jitter")
```

OPTIONAL: OLD SCHOOL - boxplot

```r
      boxplot(temp ~ gender.MF, data = bodytemp)
```

Evaluting the graphs

Males appear to have a lower median temp than female. Is this biologically based or could this difference just be due to chance (eg, just b/c we have a small sample)?

2-sample t-test

- We can assess this using a 2-sample t-test.

- Before, we had all the data pooled together, hence we did a "1-sample" test.
- Now we are splitting the data into two groups and therefore its a **two sample** test
- Two sample tests probably the most common type of t-test
- Therefore, if someones says that they did a t-test, the probably mean a 2-sample test
- For a 2-sample test we don't specify "mu"
- We specify the test using the equation notation just like the boxplot, using the tilde symbol (~)

```r
t.test(temp ~ gender.MF,
       data = bodytemp)
```

```
##
##  Welch Two Sample t-test
##
## data:  temp by gender.MF
## t = 2.2854, df = 127.51, p-value = 0.02394
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.03881298 0.53964856
## sample estimates:
## mean in group female   mean in group male
##             98.39385             98.10462
```

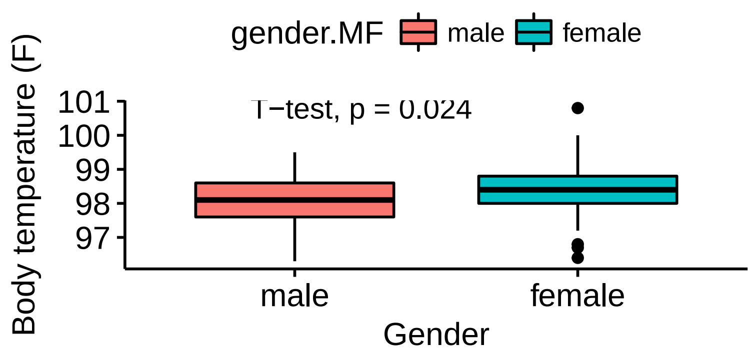
- What does this test indicate about male and female body temp?
- Female body temp is higher - is it closer to 98.6?

Plotting the 2-sample t-test

Plot boxplots w/ p value

ggpubr has a cool feature for adding p-values to plots called stat_compare_means(). Remember the "+" between ggboxplot() and stat_compare_means()!

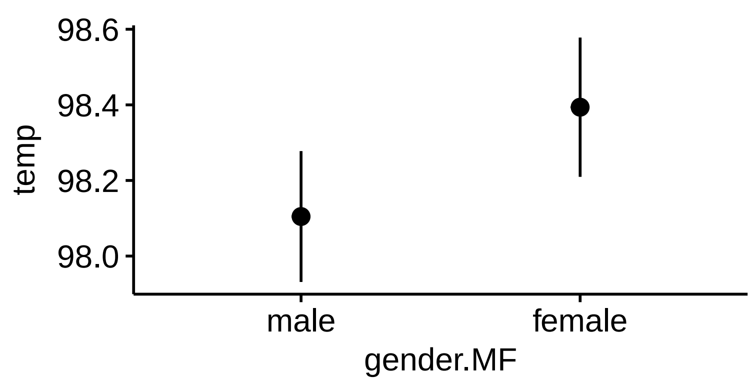
```
ggboxplot(data = bodytemp,
          y = "temp",
          x = "gender.MF",
          fill = "gender.MF",
          xlab = "Gender",
          ylab = "Body temperature (F)") +
  stat_compare_means(method = "t.test")
```



Plot means and error bars

We can quickly plot means and error bars using ggerrorplot(). The default is to plot means with error bars based on the standard error; we'll change this to 95% CIs by setting desc_stat = mean_ci

```
ggerrorplot(data = bodytemp,
           desc_stat =  "mean_ci",
          y = "temp",
          x = "gender.MF")
```

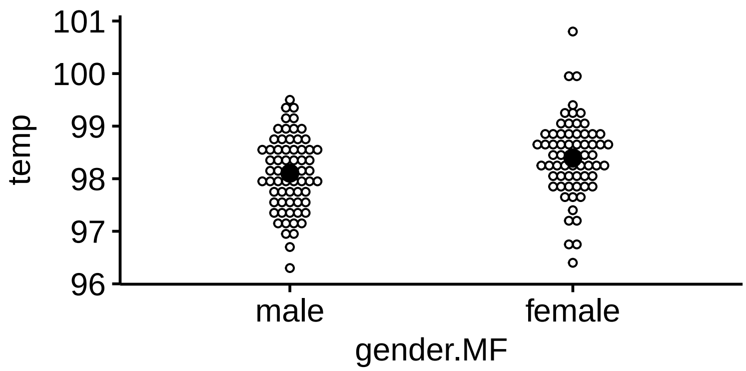


Note that the error bars overlap a fair bit, even those the t-test gives us a value of $p < 0.05$.

Plot means, error bars and raw data

A create type of plot is a "beeswarm plot", which ggpubr just calls a "dotplot". We can use add = "mean_ci" to add the means and CIs. Note how tiny the CIs are relative to the full range of the data.

```
ggdotplot(data = bodytemp,
          y = "temp",
          x = "gender.MF",
          add = "mean_ci")
```



For comparison, change it to add = mean_sd. What do you notice?

```
ggdotplot(data = bodytemp,
          y = "temp",
          x = "gender.MF",
          add = "mean_sd")
```

The SD is a measure of the variation in the data, while the SE/95% CI is a measure of how precisely we have measured the mean. Here, the SD bars are wide b/c there is a lot of variation in the data, but the 95% CI are narrow b/c the sample size is pretty big.

Question 5: How is a paired t-test similar to a 1-sample t-test?

- A paired t-test is a very common type of t-test
- A common type of pairing is before and after a treatment occurred
- A paired treatment can be set up a couple different ways in R
- Mathematically, a paired t-test is similar to a 1-sample t-test.
- With a paired t-test we are not interested in whether the overall means of each group or time point (ie before, after) are different
- We are interested in whether the difference between each pair of measurements is consistently different from zero
- **A paired t-test is therefore equivalent to a 1-sample t-test where the population parameter mu=0**

Simulate data

- We will look at paired t-tests with some fake data.

- I've simulated some data representing a person's body temp before the experimental treatment occurred (before)
- I've also simulated body temps after a stress treatment occurs (stressed)
- The hypothesis (Ha) is that being stressed changes your body temp.
- The null hypothesis (Ho) is that there is no consistent impact of stress on body temp.
- That is, while some peoples body temp goes up after the treatment, other people's go down, and on average the change in temp is about 0.
- (NOTE: I intentionally made the means to be different, so the difference is not surprising!)

Simulate fake paired data

This code makes the fake data; ignore it

```
      # ignore this
      # lm.out <- lm(temp ~ heartrate, data = bodytemp)
      # temp.orig <- simulate(lm.out, seed = 100)
      # temp.stress <- simulate(lm.out, seed = 101) + 0.5+ rnorm(length(bodytemp), mean = 0, sd = sd(bo
      # new.bodytemp <- data.frame(before = temp.orig$sim_1,stressed = temp.stress$sim_1)
      # write.csv(new.bodytemp, "fake_paired_temp_data.csv",row.names = F)
```

Load the fake paired temp data

Follow the steps use previously to load the data. I'm going to use the command read.csv(), but you can do it however you want.

```
new.bodytemp <- read.csv(file = "fake_paired_temp_data.csv")
```

Calculate the difference between "before"

- We can do some simple math to calculate the difference between before and after
- we use the dollar sign ($) to select the columns we want

```
difference1 <- new.bodytemp$before -  new.bodytemp$stressed
```

We can add this to our dataframe like this

```
new.bodytemp$difference1 <- difference1
```

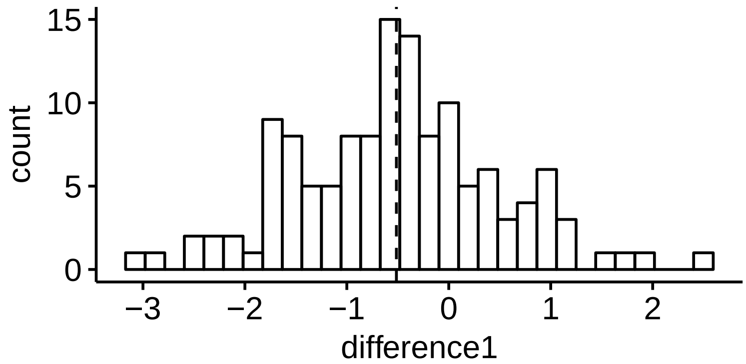
Note that we could do this all in 1 step if we wanted. Lets flip the order of the values.

```
new.bodytemp$difference2<- new.bodytemp$stressed - new.bodytemp$before
```

Visualize difference

Look at distribution of differences

```
gghistogram(new.bodytemp,
            x = "difference1",
            add = "mean")
```



What is the mean represent?

OPTIONAL - OLDSCHOOL

```
    hist(new.bodytemp$difference1)
```

1-sample t-test on difference

- We'll complete the paired t-test process by conducting a 1-sample test, setting mu = 0.
- We are therefore testing the hypothesis that the average **difference** between before and stressed is greater than zero

```
t.test(new.bodytemp$difference1 ,
       mu = 0)
```

```
##
##  One Sample t-test
##
## data:  new.bodytemp$difference1
## t = -5.8386, df = 129, p-value = 4.032e-08
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.6877954 -0.3396314
## sample estimates:
##  mean of x
## -0.5137134
```

What's the difference between these two results?

- Our 1st difference column difference1 was calculated as “new.bodytemp$before - new.bodytemp$stressed”
- This is, “before” - “stressed”
- Our 2nd difference column difference2 was calculated as " new.bodytemp$stressed - new.bodytemp$before"
- That is, “stressed - before”

Both of these columns contain equivalent data, just the signs (pos or neg) are different depending on what gets subtracted from what

```
#raw data
mean(new.bodytemp$difference1)
```

```
## [1] -0.5137134
```

```
mean(new.bodytemp$difference2)
```

```
## [1] 0.5137134
```

We can over the top check this using the abs() command to get the absolute values

```
#absolute values with abs()
mean(abs(new.bodytemp$difference1))
```

```
## [1] 0.8992502
```

```
mean(abs(new.bodytemp$difference2))
```

```
## [1] 0.8992502
```

Compare the 2 t-tests

```
#difference1 column
t.test(new.bodytemp$difference1 ,
       mu = 0)

#difference2 column
t.test(new.bodytemp$difference2 ,
       mu = 0)
```

The results are the same in terms of the p-value but different in other ways. Can you spot the differences? There are 2 (2 types of differences, 3 numbers that are different)

Paired t-test the normal way

- Its actually not necessary to calculate the difference - the t.test function can do it on the fly.
- To do things directly, give t.test() two things: the before column and the stressed column
- then, tell t.test() "paired = TRUE"
- This tells t.test() that the two columns of data are directly paired; that is, each row of data contains two numbers that are paired.

```
t.test(new.bodytemp$before,
       new.bodytemp$stressed,
       paired = TRUE)
```

OPTIONAL Question 5: Introducing scatter plots by considering the relationship between resting heart rate & temp

Making a scatterplot

```
ggscatter(y = "temp",x = "heartrate",
          data = bodytemp)
```

There might be some relationship here. We could explore this further if we wanted with linear regression.

OPTIONAL-OLDSCHOOL: Making a scatterplot using plot()

```
      plot(temp ~ heartrate,data = bodytemp)
```

Males vs. Female

Do male and females have different relationships between resting hear rate and body temp?

```
ggscatter(y = "temp",x = "heartrate",
          data = bodytemp,
          color = "gender.MF",
          shape = "gender.MF",
          add = "reg.line")
```

The patterns appear to be similar. We could explore this further with multiple linear regression.

OPTIONAL-OLDSCHOOL

Subset the data by males and females

```
males <- subset(bodytemp,  gender.MF == "male")
females <- subset(bodytemp,  gender.MF == "female")
```

Plot males and females separately

```
par(mfrow = c(1,1))
#set xlims
xlims <- c(min(bodytemp$heartrate),max(bodytemp$heartrate))
plot(temp ~ heartrate, data = males,xlim = xlims)
points(temp ~ heartrate, data = females, col = "green")
```

Appendices

Appendix 1: Code to rebuild the data from scratch

This code allows you to re-build the data without loading any data.

```
temp <- c(96.3,96.7,96.9,97,97.1,97.1,97.1,97.2,97.3,97.4,97.4,97.4,97.4,97.5,97.5,97.6,97.6,97.6,97.7,
gender <- c(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
heartrate <- c(70,71,74,80,73,75,82,64,69,70,68,72,78,70,75,74,69,73,77,58,73,65,74,76,72,78,71,74,67,6

bodytemp <- data.frame(temp = temp,gender = gender,heartrate = heartrate)
```