Example paired t-tests

Blanca Lizarbe

2023-10-25

R Markdown

You need to proceed with the data analysis of the characteristics of several blood samples from patients of a disease that alters its viscosity . Blood samples were extracted before and two weeks after being treated (or not) to reduce the effects. One group of patients was treated with drug A, another with a different drug B, and a last batch only as a control (n = 15 from drug A, n = 15 from drug B, n = 15 controls). You want to test statistically if values of viscosity change after the administration (Drug A, Durg B or Control). The corresponding values are summarized in the following code:

```
rm(list = ls())
set.seed(1)

id <- vector(length = 45)
for (i in 1:45)
  id[i] <- paste(sample(x = letters, 10, replace = TRUE), collapse = "")

group = factor(c(rep("A",15),rep("B",15),rep("Control",15)))
Visc_before = c(round(rnorm(15, mean = 110, sd = 10)),
               round(rnorm(15, mean = 100, sd = 9)),
               round(rnorm(15, mean = 105, sd = 11)))
Visc_after = c(round(rnorm(15, mean = 59, sd = 8)),
              round(rnorm(15, mean = 80, sd = 25)),
              round(rnorm(15, mean = 100, sd = 12)))

df = data.frame("ID" = id,
                "Group" = group,
                "Visc_before" = Visc_before,
                "Visc_after" = Visc_after)
```

1. Perform the appropriate statistical tests to assess if Visc values are significantly modified by drugs/control. Do the tests separately for case A, case B and case Control.
2. Provide a CI graph of each test and interpret the results.

Results

1. T.tests

```r
# We will do the test in 4 different manners:

# 1. We start by sub-setting the data
df.A <- df[df$Group == "A",]
df.B <- df[df$Group == "B",]
df.C <- df[df$Group == "Control",]

# Then perform three separated PAIRED t.tests (same subjects observed more than
# once)
(test.A <- t.test(df.A$Visc_before, df.A$Visc_after, paired = TRUE))
```

```
##
##  Paired t-test
##
## data:  df.A$Visc_before and df.A$Visc_after
## t = 15.689, df = 14, p-value = 2.806e-10
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##  44.08542 58.04791
## sample estimates:
## mean difference
##        51.06667
```

```r
(test.B <- t.test(df.B$Visc_before, df.B$Visc_after, paired = TRUE))
```

```
##
##  Paired t-test
##
## data:  df.B$Visc_before and df.B$Visc_after
## t = 4.931, df = 14, p-value = 0.0002211
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##  12.80761 32.52573
## sample estimates:
## mean difference
##        22.66667
```

```r
(test.C <- t.test(df.C$Visc_before, df.C$Visc_after, paired = TRUE))
```

```
##
##  Paired t-test
##
## data:  df.C$Visc_before and df.C$Visc_after
## t = 0.23662, df = 14, p-value = 0.8164
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##  -6.451307  8.051307
## sample estimates:
## mean difference
##             0.8
```

```r
# 2. by generating an additional column that contains the difference between before and after
df$diff <- with(df, Visc_before - Visc_after)

# and applying three t.tests

t.test(df[df$Group == "A",]$diff, alternative ='two.sided',  mu=0.0,
       conf.level=.95)
```

```
##
##  One Sample t-test
##
## data:  df[df$Group == "A", ]$diff
## t = 15.689, df = 14, p-value = 2.806e-10
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  44.08542 58.04791
## sample estimates:
## mean of x
##  51.06667
```

```r
t.test(df[df$Group == "B",]$diff,
       alternative ='two.sided',
       mu=0.0,  conf.level=.95)
```

```
##
##  One Sample t-test
##
## data:  df[df$Group == "B", ]$diff
## t = 4.931, df = 14, p-value = 0.0002211
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  12.80761 32.52573
## sample estimates:
## mean of x
##  22.66667
```

```r
t.test(df[df$Group == "Control",]$diff,
       alternative ='two.sided',
       mu=0.0,  conf.level=.95)
```

```
##
##  One Sample t-test
##
## data:  df[df$Group == "Control", ]$diff
## t = 0.23662, df = 14, p-value = 0.8164
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -6.451307  8.051307
## sample estimates:
## mean of x
##       0.8
```

```r
# 3. by a for loop

groups <- levels(df$Group)
tests.loops <- vector("list", length = 3)

for (i in 1:3)
  tests.loops [[i]] <- t.test(df[df$Group == groups[i],]$diff,
                              alternative ='two.sided',
                              mu=0.0,  conf.level=.95)

# 4. aggregate function

tests.aggregate <- aggregate(df$diff,
                             by = list(group),
                             FUN = function(x) t.test(x,
                                                      alternative ='two.sided',
                                                      mu=0.0,  conf.level=.95))
```

2. CI

In order to obtain the CI graphs of the t-tests, we first extract the estimate of the differences of means. This can be obtained directly fro the t.test results. We will create a "means" vector containing the three differences of means

```r
means <- c(test.A$estimate, test.B$estimate, test.C$estimate)

# we plot the three of them in the same graph
plot(means,
     c(1,2,3),
     xlim=c(-20 ,60),
     ylim=c(0,3),
     yaxt = "n",
     xlab="mean viscosity changes", ylab="Type of drug" )
text(x = means[1:3],
     y = c(0.9,1.9,2.9),
     labels = c("Drug A", "Drug B", "Control"))
# We next use the "arrows" function, as we did in class, to add to the previous
# plot the arrows expressing the CI information.

# We can obtain directly the CI from the t.test results.
arrows(x0 = test.A$conf.int[1],
       y0 = 1,
       x1 = test.A$conf.int[2],
       y1 = 1,
       code = 3, col="red", lwd = 1)

arrows(test.B$conf.int[1],
       y0 = 2,
       test.B$conf.int[2],
       y1 = 2,
       code = 3, col="red", lwd = 1)

arrows(test.C$conf.int[1],
```

```
       y0 = 3,
       test.C$conf.int[2],
       y1 = 3,
       code = 3, col="red", lwd = 1)
# We add a vertical line on x = 0 to highlight the line that should
# "not be crossed" to reject the null hypothesis
abline(v = 0, lwd = 2, lty = 2)
```