Two-sample t-tests

Patrick Breheny

October 20, 2016

Today's lab will focus on the two-sample t-test: how to carry it out in R, and comparing the equal-variance and unequal-variance approaches in terms of power and validity (proper coverage and type I error rates).

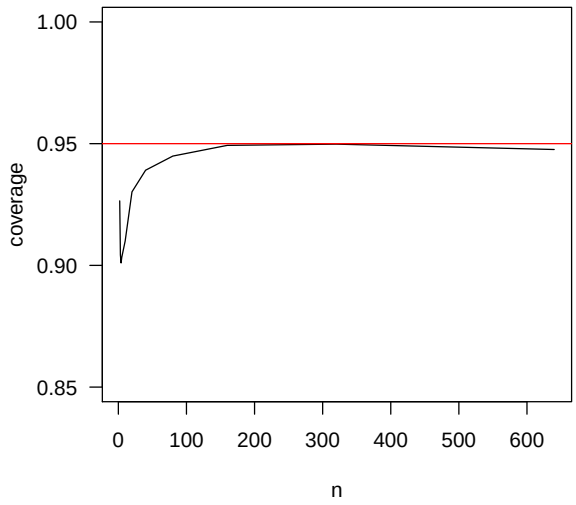
1 How well do t intervals do when the data is not normally distributed: Part 2

First, I just wanted to briefly revisit our simulation from last time looking at the coverage of the t-interval for data that does not follow a perfectly normal distribution. Mainly, I wanted to plot coverage as a function of n, and also show how to present nonlinear axes in R plots.

Let's carry out the simulation, but loop over several values of n:

```
> lipids <- read.delim("http://myweb.uiowa.edu/pbreheny/data/lipids.txt")
> pop <- lipids$TRG
> N <- 10000
> n <- c(2, 3, 4, 5, 10, 20, 40, 80, 160, 320, 640)
> covered <- matrix(NA, N, length(n), dimnames=list(1:N, n))
> pb <- txtProgressBar(1, N, style=3)
> for (i in 1:N) {
+   for (j in 1:length(n)) {
+     sam <- sample(pop, n[j], replace=TRUE)
+     if (sd(sam)==0) {
+       covered[i,j] <- 0
+       next
+     }
+     covered[i,j] <- t.test(sam, mu=mean(pop))$p.value > 0.05
+   }
+   setTxtProgressBar(pb, i)
+ }
```

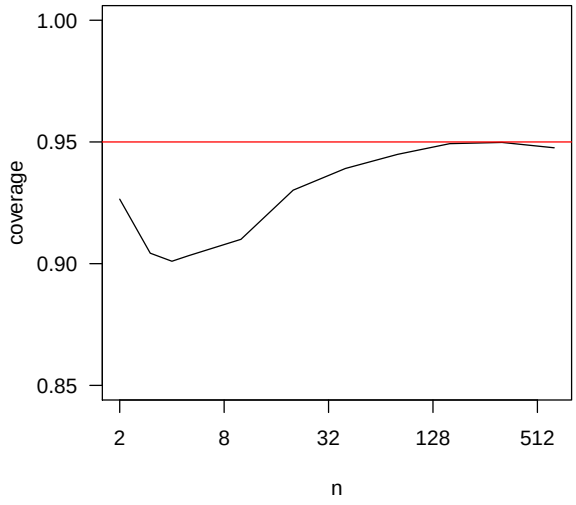
```
> coverage <- apply(covered, 2, mean)
> plot(n, coverage, type="l", ylim=c(0.85,1), las=1)
> abline(h=0.95, col="red")
```



So as we saw last week, the t intervals are pretty robust to non-normality. Even with fairly skewed data, coverage is pretty good – a little erratic for small sample sizes, but never too far below the nominal 95% coverage.

One thing that's not ideal about this plot is that the "interesting" part of the graph (where coverage is changing) is squeezed into a very small part of the plot. One way to show that part of the plot in greater detail, but still show what happens for large sample sizes, is to use a nonlinear axis. There are a few different ways to set this up in `R`, but the most flexible is to use the `axis` function:

```
> plot(log2(n), coverage, type="l", ylim=c(0.85,1), xaxt="n", xlab="n", las=1)
> x <- seq(1, 9, 2)
> axis(1, at=x, labels=2^x)
> abline(h=0.95, col="red")
```



Personally, I like this plot a little better, as it allows you to see the exact sample sizes at which the interesting dip in coverage occurs.

2 Two sample t-tests

As we saw last week, one can carry out t-tests in `R` using the `t.test` function, which can perform two-sample t-tests as well as paired t-tests. To see how they work, let's look at the lead smelter data that we discussed in class. Starting with Student's approach (the equal variance assumption), we have:

```
> lead <- read.delim("http://myweb.uiowa.edu/pbreheny/data/lead-iq.txt")
> head(lead)

  Smelter IQ
1     Far 70
2     Far 85
3     Far 86
4     Far 76
5     Far 96
6     Far 94

> t.test(IQ~Smelter, data=lead, var.equal=TRUE)


Two Sample t-test

data:  IQ by Smelter
t = 1.3505, df = 122, p-value = 0.1793
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -1.627295  8.614465
sample estimates:
 mean in group Far mean in group Near
          92.68657           89.19298

> ## By hand:
> n <- with(lead, by(IQ, Smelter, length))
> m <- with(lead, by(IQ, Smelter, mean))
> v <- with(lead, by(IQ, Smelter, var))
> Sp <- sqrt(weighted.mean(v, n-1))
> SE <- Sp*sqrt(1/n[1] + 1/n[2])
> 2*pt((m[1]-m[2])/SE, sum(n)-2, lower.tail=FALSE)

     Far
0.179346

> m[1]-m[2] + qt(c(0.025,0.975), sum(n)-2)*SE

[1] -1.627295  8.614465
```

This agrees with the results we obtained in class. Note the `var.equal=TRUE` argument, which specifies that we want to proceed under the assumption of equal variances. By default, this is false; i.e., the Welch t-test is performed:

```
> t.test(IQ~Smelter, data=lead)


Welch Two Sample t-test

data:  IQ by Smelter
t = 1.38, df = 120.62, p-value = 0.1702
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -1.518663  8.505833
sample estimates:
 mean in group Far mean in group Near
          92.68657           89.19298

> ## By hand:
> SE <- sqrt(sum(v/n))
> df <- sum(v/n)^2/sum((v/n)^2/(n-1))
> 2*pt((m[1]-m[2])/SE, df, lower.tail=FALSE)

      Far
0.1701521

> m[1]-m[2] + qt(c(0.025,0.975), df)*SE

[1] -1.518663  8.505833
```

3 Student vs. Welch t-test: Comparison

In this section, we'll be comparing the Student (equal variance) and Welch (unequal variance) approaches to the two-sample t-test in terms of power and preservation of type I error rate. We could, of course, approach this simulation from a confidence interval perspective instead, but given the connection between the hypothesis tests and confidence intervals, the conclusions we reach would be similar.

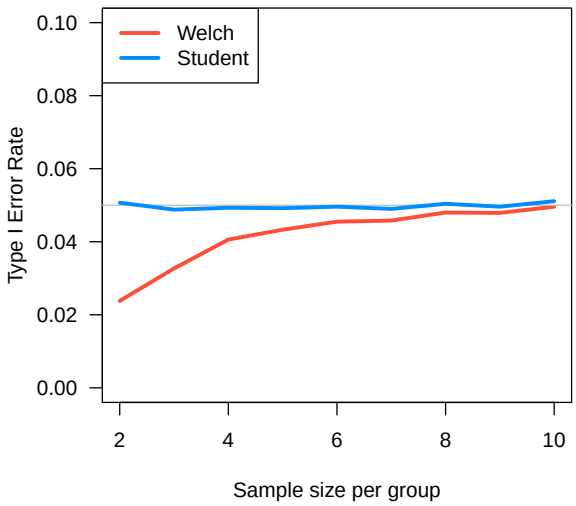
3.1 Equal variance case

What if the assumption made by the Student approach – equal variance – actually holds? For the sake of time in lab, you can run these with $N = 1000$ replications, but the lines will be a little bit "wiggly".

```
> col <- c("#FF4E37FF", "#008DFFFF")
> N <- 10000
> n <- 2:10
> pW <- pS <- matrix(NA, N, length(n))
> pb <- txtProgressBar(1, N, style=3)
> for (i in 1:N) {
+   for (j in 1:length(n)) {
+     s1 <- rnorm(n[j], mean=0, sd=1)
+     s2 <- rnorm(n[j], mean=0, sd=1)
+     pW[i,j] <- t.test(s1, s2, var.equal=FALSE)$p.value
+     pS[i,j] <- t.test(s1, s2, var.equal=TRUE)$p.value
+   }
```

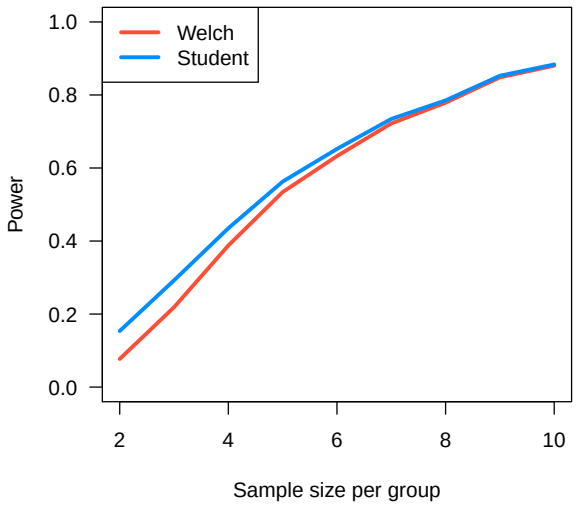
```
+   setTxtProgressBar(pb, i)
+ }
```

```
> plot(n, apply(pW < 0.05, 2, mean), type="l", lwd=3, col=col[1], las=1,
+      ylab="Type I Error Rate", xlab="Sample size per group", ylim=c(0,0.1))
> abline(h=0.05, col="gray")
> lines(n, apply(pS < 0.05, 2, mean), lwd=3, col=col[2])
> legend("topleft", lwd=3, col=col, legend=c("Welch","Student"))
```




```
> pW <- pS <- matrix(NA, N, length(n))
> pb <- txtProgressBar(1, N, style=3)
> for (i in 1:N) {
+   for (j in 1:length(n)) {
+     s1 <- rnorm(n[j], mean=1.5, sd=1)
+     s2 <- rnorm(n[j], mean=0, sd=1)
+     pW[i,j] <- t.test(s1, s2, var.equal=FALSE)$p.value
+     pS[i,j] <- t.test(s1, s2, var.equal=TRUE)$p.value
+   }
+   setTxtProgressBar(pb, i)
+ }
```

```
> plot(n, apply(pW < 0.05, 2, mean), type="l", lwd=3, col=col[1], ylab="Power",
+      xlab="Sample size per group", ylim=c(0,1), las=1)
> lines(n, apply(pS < 0.05, 2, mean), lwd=3, col=col[2])
> legend("topleft", lwd=3, col=col, legend=c("Welch","Student"))
```



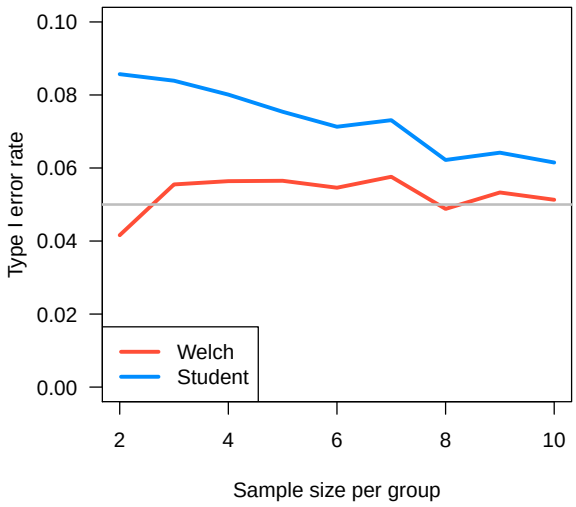
What conclusions should we draw from this simulation?

3.2 Unequal variance case

OK, now let's look at the question: What if the equal variance assumption is wrong? Does the Student approach dramatically fail, or is it robust to this assumption?

```
> pW <- pS <- matrix(NA, N, length(n))
> pb <- txtProgressBar(1, N, style=3)
> for (i in 1:N) {
+   for (j in 1:length(n)) {
+     s1 <- rnorm(n[j], mean=0, sd=1)
+     s2 <- rnorm(n[j], mean=0, sd=4)
+     pW[i,j] <- t.test(s1, s2, var.equal=FALSE)$p.value
+     pS[i,j] <- t.test(s1, s2, var.equal=TRUE)$p.value
+   }
+   setTxtProgressBar(pb, i)
+ }
```

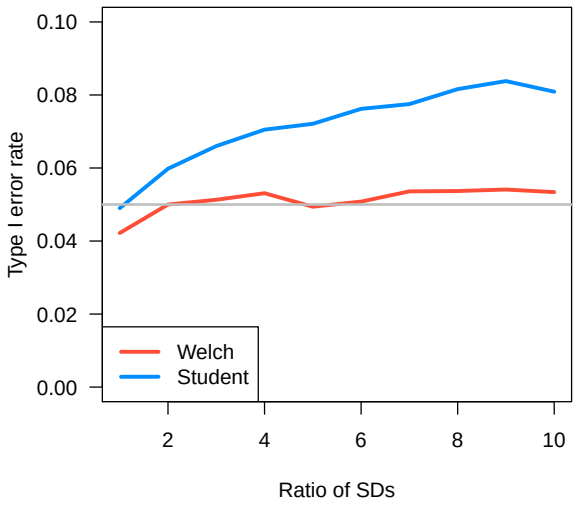
```
> plot(n, apply(pW < 0.05, 2, mean), type="l", lwd=3, col=col[1], las=1,
+      ylab="Type I error rate", xlab="Sample size per group", ylim=c(0,0.1))
> lines(n, apply(pS < 0.05, 2, mean), lwd=3, col=col[2])
> abline(h=0.05, col="gray", lwd=2)
> legend("bottomleft", lwd=3, col=col, legend=c("Welch","Student"))
```



Now let's look at what happens as we vary the ratio of standard deviations (as opposed to the sample size):

```
> n <- 5
> SD <- 1:10
> pW <- pS <- matrix(NA, N, length(SD))
> pb <- txtProgressBar(1, N, style=3)
> for (i in 1:N) {
+   for (j in 1:length(SD)) {
+     s1 <- rnorm(n, mean=0, sd=1)
+     s2 <- rnorm(n, mean=0, sd=SD[j])
+     pW[i,j] <- t.test(s1, s2, var.equal=FALSE)$p.value
+     pS[i,j] <- t.test(s1, s2, var.equal=TRUE)$p.value
+   }
+   setTxtProgressBar(pb, i)
+ }
```

```
> plot(SD, apply(pW < 0.05, 2, mean), type="l", lwd=3, col=col[1],
+      ylab="Type I error rate", xlab="Ratio of SDs", ylim=c(0,0.1), las=1)
> lines(SD, apply(pS < 0.05, 2, mean), lwd=3, col=col[2])
> abline(h=0.05, col="gray", lwd=2)
> legend("bottomleft", lwd=3, col=col, legend=c("Welch","Student"))
```



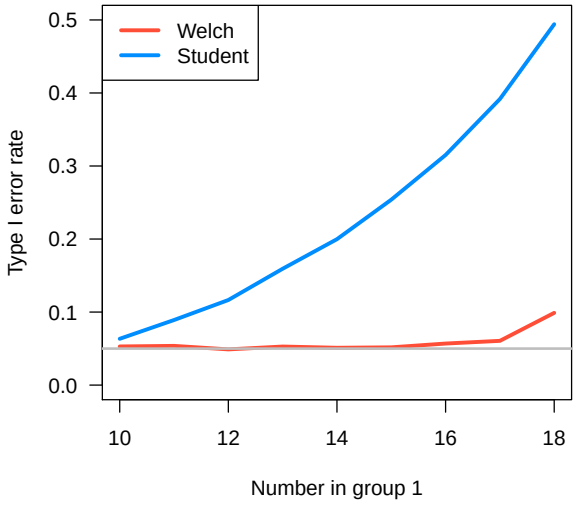
So far, it seems that the Student approach is surprisingly robust to the assumption of equal variances. Even when $n = 5$ and one group has a standard deviation 10 times that of the other (i.e., its variance is 100 times larger), the Student approach is still acceptable: its type I error rate is only 8%, or to put it another way, its confidence intervals still have 92% coverage.

It is tempting at this point to conclude that Student's approach is incredibly robust to the equal variance assumption, and that we have little to worry about when making this assumption. However, this lab illustrates an important shortcoming of simulations: although they are wonderful tools for investigating the empirical performance of statistical methods in settings where assumptions are not perfectly met, they can only investigate a finite number of configurations. In other words, a simulation study is only as good as its design – if the design does not thoroughly investigate all the possible cases, its results may be misleading.

This is indeed the case here: Student's t-test *can* fail spectacularly, but only when the sample sizes of the two groups are unequal.

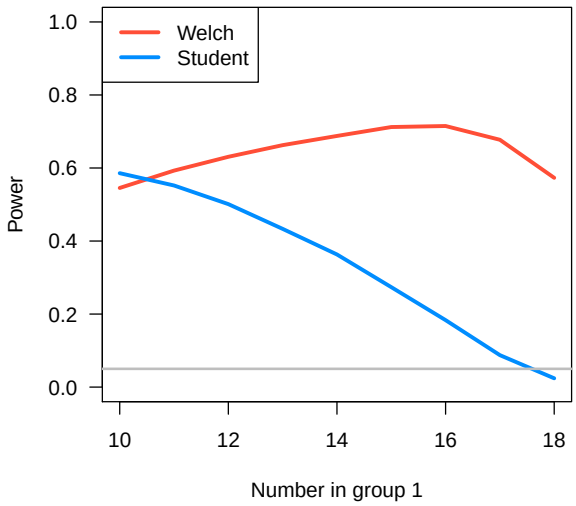
```
> n1 <- 10:18
> pW <- pS <- matrix(NA, nrow=N, ncol=length(n1))
> pb <- txtProgressBar(1, N, style=3)
> for (i in 1:N) {
+   for (j in 1:length(n1)) {
+     s1 <- rnorm(n1[j],    mean=0, sd=1)
+     s2 <- rnorm(20-n1[j], mean=0, sd=4)
+     pW[i,j] <- t.test(s1, s2, var.equal=FALSE)$p.value
+     pS[i,j] <- t.test(s1, s2, var.equal=TRUE)$p.value
+   }
+   setTxtProgressBar(pb, i)
+ }
```

```
> plot(n1, apply(pW < 0.05, 2, mean), type="l", lwd=3, col=col[1],
+      ylab="Type I error rate", xlab="Number in group 1", ylim=c(0,0.5), las=1)
> lines(n1, apply(pS < 0.05, 2, mean), lwd=3, col=col[2])
> abline(h=0.05, col="gray", lwd=2)
> legend("topleft", lwd=3, col=col, legend=c("Welch","Student"))
```




```
> n1 <- 10:18
> pW <- pS <- matrix(NA, nrow=N, ncol=length(n1))
> pb <- txtProgressBar(1, N, style=3)
> for (i in 1:N) {
+   for (j in 1:length(n1)) {
+     s1 <- rnorm(n1[j],    mean=0, sd=4)
+     s2 <- rnorm(20-n1[j], mean=3, sd=1)
+     pW[i,j] <- t.test(s1, s2, var.equal=FALSE)$p.value
+     pS[i,j] <- t.test(s1, s2, var.equal=TRUE)$p.value
+   }
+   setTxtProgressBar(pb, i)
+ }
```

```
> plot(n1, apply(pW < 0.05, 2, mean), type="l", lwd=3, col=col[1], ylab="Power",
+      xlab="Number in group 1", ylim=c(0, 1), las=1)
> lines(n1, apply(pS < 0.05, 2, mean), lwd=3, col=col[2])
> abline(h=0.05, col="gray", lwd=2)
> legend("topleft", lwd=3, col=col, legend=c("Welch","Student"))
```



Discussion: Why does the equal variance t-test fail in this way? Why is it sometimes far too conservative, and other times far too liberal?

Some concluding points:

- As long as you have $n > 10$ in each group, there is little reason to prefer Student's test: any possible gain in power is negligible at this point
- Student's t-test is, in general, very robust to the assumptions it makes – however, it can fail dramatically if *both* the standard deviations *and* the sample sizes are unequal
- Simulations are powerful tools, but require careful consideration and thoughtful design