False Positive Error Bounds

Samantha Bothwell

7/11/2019

False Positive

Let's start with a definiton of what a false positive is:

False Positive: Make the decision to reject the null hypothesis (H_o) when the null is true.

This is the same as a Type I error. We know that the probability that you make a Type I error is α so,

$$P(\text{False Positive}) = P(\text{Reject } H_o | \text{True } H_o) = \alpha$$

Since you can either have a false positive (we'll denote as FP) or not, we can say the following:

$$FP \sim Binomial(N, \alpha),$$

where N is the number of trials. We expect the number of false positives to be $\mu = N\alpha$. The standard deviation, based on the binomial distribution, is $\sigma = \sqrt{N\alpha(1-\alpha)}$. The confidence interval around this mean will be

$$\left[N\alpha - t_{\alpha/2}\sqrt{\frac{N\alpha(1-\alpha)}{N}}, N\alpha + t_{\alpha/2}\sqrt{\frac{N\alpha(1-\alpha)}{N}}\right].$$

Theoretical confidence interval

Let's use $\alpha = 0.05$ and $n = 1000$.

The expected number of false positives is $1000 \times 0.05 =$ **50**.

The standard deviation is $\sqrt{1000 \times 0.05(1-0.05)} =$ **6.89**.

To determine the confidence interval we will use, let's use R:

```
# Since we have a large sample size, we will use the normal approximation for the binomial distribution
50 - 1.96*6.89/sqrt(1000)
```

```
## [1] 49.57295
```

```
50 + 1.96*6.89/sqrt(1000)
```

```
## [1] 50.42705
```

So, the 95% confidence interval for the number of false positives is **[49.6, 50.4]** - or we can say the 95% confidence interval for the proportion of false positives is [0.0496, 0.0504].

Unfortunately, we cannot always know exactly where our false positive lies. But we are 95% confident that the true number of false positives present in a sample of 1000 is within the interval [49.6, 50.4].

Simulation

Let's start with an easy example to determine the number of false positives. We will be using white noise - so we'll test the following hypothesis:

$$H_o : \mu = 0$$
$$H_A : \mu \neq 0$$

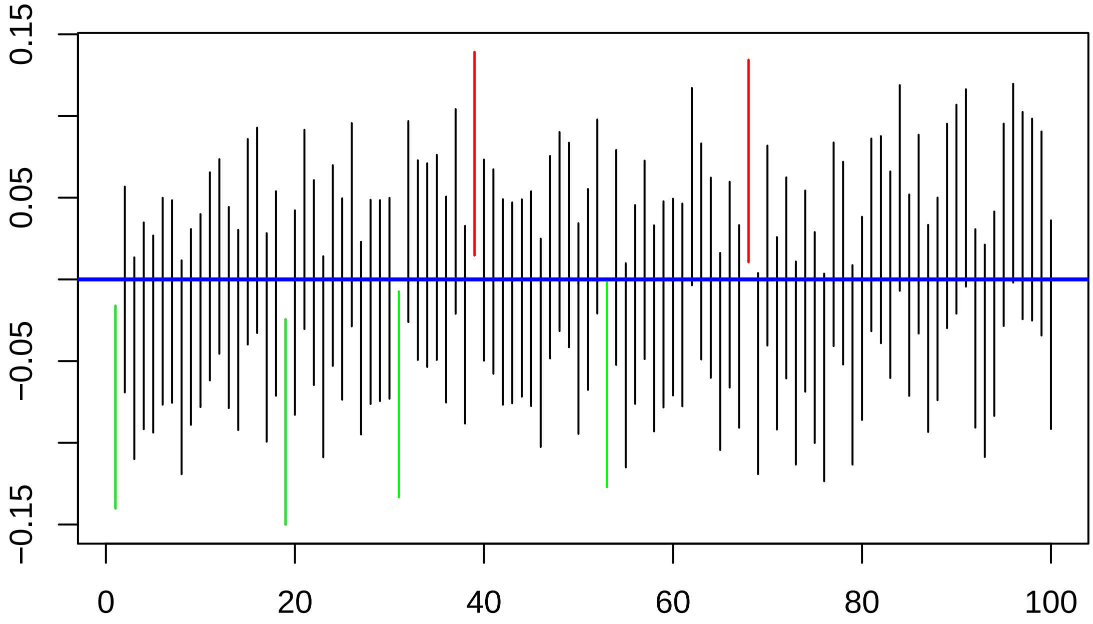
Since the samples are taken from a standard normal distribution, we know our null is true. Let's see when we'll reject the null by creating confidence intervals and seeing how many contain 0:

```r
set.seed(2019) # set seed for reproducible results

# create matrix to store confidence intervals
n = 100
dat = matrix(data = NA, nrow = n, ncol = 3)
dat = as.data.frame(dat)

# Simulation to create confidence intervals from a standard normal
for (i in 1:n){
  white_noise = rnorm(1000, mean = 0, sd = 1)
  mn = mean(white_noise)
  sderr = sd(white_noise)/sqrt(1000)
  lower = mn - 1.96*sderr
  upper = mn + 1.96*sderr

  dat[i,1] = mn; dat[i,2] = lower; dat[i,3] = upper
}
colnames(dat) = c("Mean", "Lower", "Upper")

# Check if CI is too high or too low
too_high = (0 < dat$Lower)
too_low = (0 > dat$Upper)

# Proportion of misses - this should be approximately alpha
paste("Proportion of hits:", mean(too_low + too_high))
```

```
## [1] "Proportion of hits: 0.06"
```

So, in this case the proportion of false positives in a sample of 100 replicates of a normal distribution is 0.06 - meaning we have 6 total false positives. We can visualize our 100 confidence intervals and see the 6 replicates that missed $\mu = 0$.

```r
# Plot the CIs
plot(c(1,100), c(min(dat$Lower), max(dat$Upper)), type = "n", xlim = c(1,100),
     xlab = paste(n, "replicates of confidence intervals"), ylab = "")
x = 1:n #index for each of the n CIs
segments(x, dat$Lower, x, dat$Upper)
segments(x[too_high], dat$Lower[too_high], x[too_high], dat$Upper[too_high], col = "red")
segments(x[too_low], dat$Lower[too_low], x[too_low], dat$Upper[too_low], col = "green")
abline(h = 0, col = "blue", lwd = 2)
```

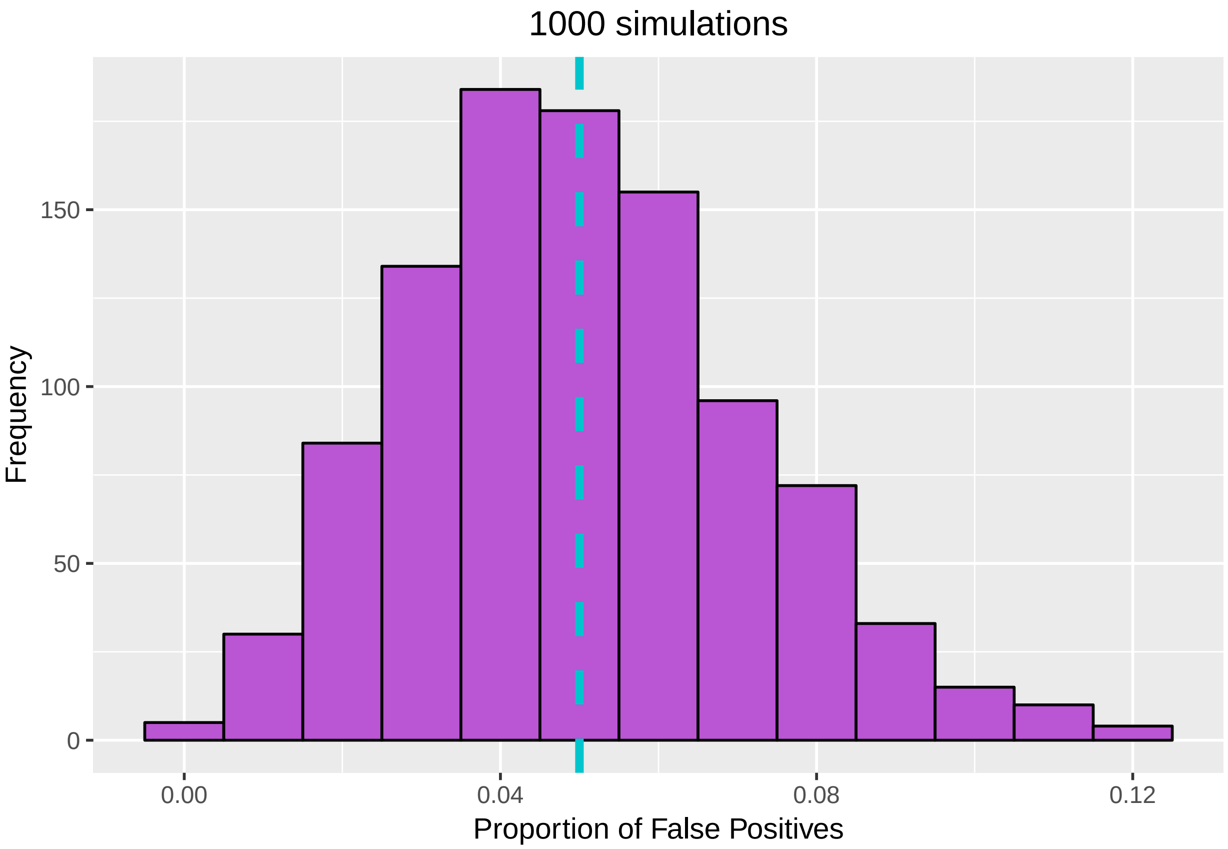


100 replicates of confidence intervals

Based on these confidence intervals, we would reject the null hypothesis for the samples highlighted in green or red. These are all false positives - we reject the null, but the null is true.

Now let's run some simulations to find the distribution for false positives and the confidence interval. We will use the simulation above but run it 1000 times to get a list of the number of false positives.

```
set.seed(2019)

N = 1000
n = 100

misses = matrix(data = NA, nrow = N, ncol = 1)
dat = matrix(data = NA, nrow = n, ncol = 3)
dat = as.data.frame(dat)
colnames(dat) = c("Mean", "Lower", "Upper")

# Simulation to determine a CI for false positives
for (j in 1:N){
  for (i in 1:n){
    white_noise = rnorm(1000, mean = 0, sd = 1)
    mn = mean(white_noise)
    sderr = sd(white_noise)/sqrt(1000)
    lower = mn - 1.96*sderr
    upper = mn + 1.96*sderr

    dat[i,1] = mn; dat[i,2] = lower; dat[i,3] = upper
  }

  too_high = (0 < dat$Lower)
  too_low = (0 > dat$Upper)

  # Update matrix by noting the number of false positives (misses) in the sample
  misses[j] = mean(too_low + too_high)
}

library(ggplot2)
qplot(misses, geom = "histogram", binwidth = 0.01, main = paste(N, "simulations"),
      xlab = "Proportion of False Positives", ylab = "Frequency", fill = I("mediumorchid"),
      col = I("black")) +
  geom_vline(xintercept = 0.05, linetype = "dashed", color = "turquoise3", size = 1.5) +
  theme(plot.title = element_text(hjust = 0.5))
```



Now we can use this simulation to determine the confidence interval for the proportion of false positives - before we found the theoretical confidence interval to be [0.0496, 0.0504]. Let's see how close our simulation came to this:

```
test = t.test(misses, mu = 0.05)
test$conf.int[c(1,2)]
```

```
## [1] 0.04874016 0.05143984
```

Based on the simulations, the 95% confidence interval for the number of false positives is [0.048, 0.051] - this is pretty close to our theoretical calculation.