

1. A group of researchers is running an experiment over the course of 30 months, with a single observation collected at the end of each month. Let  $X_1, \dots, X_{30}$  denote the observations for each month. From prior studies, the researchers know that

$$X_i \sim f_X(x),$$

but the mean  $\mu_X$  is unknown, and they wish to conduct the following test

$$H_0 : \mu_X = 0$$

$$H_a : \mu_X > 0.$$

At month  $k$ , they have accumulated data  $X_1, \dots, X_k$  and they have the  $t$ -statistic

$$T_k = \frac{\bar{X} - 0}{S_k / \sqrt{n}}.$$

The initial plan was to test the hypotheses after all data was collected (at the end of month 30), at level  $\alpha = 0.05$ . However, conducting the experiment is expensive, so the researchers want to “peek” at the data at the end of month 20 to see if they can stop it early. That is, the researchers propose to check whether  $t_{20}$  provides statistically discernible support for the alternative. If it does, they will stop the experiment early and report support for the researcher’s alternative hypothesis. If it does not, they will continue to month 30 and test whether  $t_{30}$  provides statistically discernible support for the alternative.

- (a) What values of  $t_{20}$  provide statistically discernible support for the alternative hypothesis?

**Solution:** Below, we find the value of  $t_{20}$  that provides statistically discernible support for the alternative hypothesis?

```
n <- 20
qt(0.95, df = n - 1) #find the value for t for 20 observations

## [1] 1.729133
```

- (b) What values of  $t_{30}$  provide statistically discernible support for the alternative hypothesis?

**Solution:** Below, we find the value of  $t_{30}$  that provides statistically discernible support for the

```
n <- 30
qt(0.95, df = n - 1) #find the value for t for 30 observations

## [1] 1.699127
```

- (c) Suppose  $f_X(x)$  is a Laplace distribution with  $a = 0$  and  $b = 4.0$ . Conduct a simulation study to assess the Type I error rate of this approach.

**Note:** You can use the `rlaplace()` function from the `VGAM` package for R (Yee, 2010).

**Solution:** Below, we find the Type I error rate of this approach.

```
R <- 1000 #resample size
hypo.check <- tibble( #tibble to store data
  reject = rep(NA, R)
)

for (i in 1:R){
  sim.data.30 <- rlaplace(n = 30, location = 0, scale = 4) #simulate rlapace data for n = 30
  sim.data.20 <- sim.data.30[1:20] #take the first 20 observations

  p.value.20 <- t.test(sim.data.20, #grab the p-value for the first 20 observations
    mu = 0,
    alternative = "greater")$p.value
  p.value.30 <- t.test(sim.data.30, #grab the p-value for the first 30 observations
    mu = 0,
    alternative = "greater")$p.value

  #Perform a hyphesis check for n = 20 and n = 30
  hypo.check$reject[i] <- ifelse(p.value.20 <= 0.05, 1,
    ifelse(p.value.30 <= 0.05, 1, 0))
}

type1.error.rate <- sum(hypo.check$reject)/R #Caluicate the Type I error rate
type1.error.rate #display

## [1] 0.098
```

- (d) **Optional Challenge:** Can you find a value of  $\alpha < 0.05$  that yields a Type I error rate of 0.05?  
**Solution:** Below, we find the value of  $\alpha < 0.05$  that yields a Type I error rate of 0.05.

```
solve <- function(x) { #Create a function to solve for f(x) = 0
  hypo.check <- tibble( #tibble to store data
    reject = rep(NA, R)
  )

  for (i in 1:R){
    sim.data.30 <- rlaplace(n = 30, location = 0, scale = 4) #simulate rlapace data for n = 30
    sim.data.20 <- sim.data.30[1:20] #take the first 20 observations

    p.value.20 <- t.test(sim.data.20, #grab the p-value for the first 20 observations
      mu = 0,
      alternative = "greater")$p.value
    p.value.30 <- t.test(sim.data.30, #grab the p-value for the first 30 observations
      mu = 0,
      alternative = "greater")$p.value

    #Perform a hypothesis check for n = 20 and n = 30
    hypo.check$reject[i] <- ifelse(p.value.20 <= x, 1,
      ifelse(p.value.30 <= x, 1, 0))
  }

  return(50-sum(hypo.check)) #return the solution (Should be 0)
}

alpha <- uniroot(solve, lower = 0, upper = 0.05)$root #solve for alpha
alpha #display

## [1] 0.03606428
```

2. Perform a simulation study to assess the robustness of the  $T$  test. Specifically, generate samples of size  $n = 15$  from the Beta(10,2), Beta(2,10), and Beta(10,10) distributions and conduct the following hypothesis tests against the actual mean for each case (e.g.,  $\frac{10}{10+2}$ ,  $\frac{2}{10+2}$ , and  $\frac{10}{10+10}$ ).

- (a) What proportion of the time do we make an error of Type I for a left-tailed test?

**Solution:** Below, we find the proportion of the times we make a Type I error for the Beta(10,2), Beta(2,10), and Beta(10,10) distributions with a left-tailed test.

```
hypo.check.less <- tibble( #Tibble to store our data
  reject.10.2 = rep(NA, R),
  reject.2.10 = rep(NA, R),
  reject.10.10 = rep(NA, R)
)

for (i in 1:R){
  sample.beta.10.2 <- rbeta(n = 15, shape1 = 10, shape2 = 2) #Generate some samples of beta (10,2)
  sample.beta.2.10 <- rbeta(n = 15, shape1 = 2, shape2 = 10) #Generate some samples of beta (2,10)
  sample.beta.10.10 <- rbeta(n = 15, shape1 = 10, shape2 = 10) #Generate some samples of beta (10,10)

  p.value.1 <- t.test(sample.beta.10.2, #Perform t-test and grab p-values for all samples
    mu = 10/12,
    alternative = "less")$p.value

  p.value.2 <- t.test(sample.beta.2.10,
    mu = 2/12,
    alternative = "less")$p.value

  p.value.3 <- t.test(sample.beta.10.10,
    mu = 10/20,
    alternative = "less")$p.value

  #Do hypothesis comparison and determine whether p-value < 0.05 (alpha)
  hypo.check.less$reject.10.2[i] <- ifelse(p.value.1 <= 0.05, 1, 0)
  hypo.check.less$reject.2.10[i] <- ifelse(p.value.2 <= 0.05, 1, 0)
  hypo.check.less$reject.10.10[i] <- ifelse(p.value.3 <= 0.05, 1, 0)
}

type1.error.rate.10.2 <- sum(hypo.check.less$reject.10.2)/R #Calculates type 1 error proportions
type1.error.rate.2.10 <- sum(hypo.check.less$reject.2.10)/R
type1.error.rate.10.10 <- sum(hypo.check.less$reject.10.10)/R

type1.error.rate.10.2

## [1] 0.022
```

```

type1.error.rate.2.10

## [1] 0.07

type1.error.rate.10.10

## [1] 0.052

```

- (b) What proportion of the time do we make an error of Type I for a right-tailed test?

**Solution:** Below, we find the proportion of the times we make a Type I error for the Beta(10,2), Beta(2,10), and Beta(10,10) distributions with a right-tailed test.

```

hypo.check.greater <- tibble( #Tibble to store our data
  reject.10.2 = rep(NA, R),
  reject.2.10 = rep(NA, R),
  reject.10.10 = rep(NA, R)
)

for (i in 1:R){
  sample.beta.10.2 <- rbeta(n = 15, shape1 = 10, shape2 = 2) #Generate some samples of beta (10,2)
  sample.beta.2.10 <- rbeta(n = 15, shape1 = 2, shape2 = 10) #Generate some samples of beta (2,10)
  sample.beta.10.10 <- rbeta(n = 15, shape1 = 10, shape2 = 10) #Generate some samples of beta (10,10)

  p.value.1 <- t.test(sample.beta.10.2,          #Perform t-test and grab p-values for all samples
    mu = 10/12,
    alternative = "greater")$p.value

  p.value.2 <- t.test(sample.beta.2.10,
    mu = 2/12,
    alternative = "greater")$p.value

  p.value.3 <- t.test(sample.beta.10.10,
    mu = 10/20,
    alternative = "greater")$p.value

  #Do hypothesis comparison and determine whether p-value < 0.05 (alpha)
  hypo.check.greater$reject.10.2[i] <- ifelse(p.value.1 <= 0.05, 1, 0)
  hypo.check.greater$reject.2.10[i] <- ifelse(p.value.2 <= 0.05, 1, 0)
  hypo.check.greater$reject.10.10[i] <- ifelse(p.value.3 <= 0.05, 1, 0)
}

type1.error.rate.10.2 <- sum(hypo.check.greater$reject.10.2)/R #Calculates type 1 error proportions
type1.error.rate.2.10 <- sum(hypo.check.greater$reject.2.10)/R
type1.error.rate.10.10 <- sum(hypo.check.greater$reject.10.10)/R

type1.error.rate.10.2

## [1] 0.075

type1.error.rate.2.10

## [1] 0.021

type1.error.rate.10.10

## [1] 0.052

```

- (c) What proportion of the time do we make an error of Type I for a two-tailed test?

**Solution:** Below, we find the proportion of the times we make a Type I error for the Beta(10,2), Beta(2,10), and Beta(10,10) distributions with a two-tailed test.

```

hypo.check.two.sided <- tibble( #Tibble to store our data
  reject.10.2 = rep(NA, R),
  reject.2.10 = rep(NA, R),
  reject.10.10 = rep(NA, R)
)

for (i in 1:R){
  sample.beta.10.2 <- rbeta(n = 15, shape1 = 10, shape2 = 2) #Generate some samples of beta (10,2)
  sample.beta.2.10 <- rbeta(n = 15, shape1 = 2, shape2 = 10) #Generate some samples of beta (2,10)
  sample.beta.10.10 <- rbeta(n = 15, shape1 = 10, shape2 = 10) #Generate some samples of beta (10,10)

  p.value.1 <- t.test(sample.beta.10.2,          #Perform t-test and grab p-values for all samples
    mu = 10/12,
    alternative = "two.sided")$p.value

  p.value.2 <- t.test(sample.beta.2.10,
    mu = 2/12,

```

```

alternative = "two.sided")$p.value

p.value.3 <- t.test(sample.beta.10.10,
mu = 10/20,
alternative = "two.sided")$p.value

#Do hypothesis comparison and determine whether p-value < 0.05 (alpha)
hypo.check.two.sided$reject.10.2[i] <- ifelse(p.value.1 <= 0.05, 1, 0)
hypo.check.two.sided$reject.2.10[i] <- ifelse(p.value.2 <= 0.05, 1, 0)
hypo.check.two.sided$reject.10.10[i] <- ifelse(p.value.3 <= 0.05, 1, 0)
}

#Calculates type 1 error proportions
type1.error.rate.10.2 <- sum(hypo.check.two.sided$reject.10.2)/R
type1.error.rate.2.10 <- sum(hypo.check.two.sided$reject.2.10)/R
type1.error.rate.10.10 <- sum(hypo.check.two.sided$reject.10.10)/R

type1.error.rate.10.2

## [1] 0.062

type1.error.rate.2.10

## [1] 0.061

type1.error.rate.10.10

## [1] 0.059

```

- (d) How does skewness of the underlying population distribution effect Type I error across the test types?

**Solution:** As seen by our Type I error proportions, Type I error can be inflated for deflated depending on the skewness of the data and the type of test we use. For the right skewed distributions (Beta(2,10)), Type I error is inflated with right-tailed tests and deflated with left-tailed tests. For the left-skewed distributions (Beta(10,2)), Type I error is inflated with left-tailed tests and deflated with right-tailed tests. With symmetric distributions (10,10), we get balanced Type I errors with both left and right tailed tests. However, we also get balanced Type I errors when we use a two-sided test for all three distributoon types.

## References

Yee, T. W. (2010). The VGAM package for categorical data analysis. *Journal of Statistical Software*, 32(10):1–34.