

1. When conducting the work of Lab 11, we conducted the test that uses the Central Limit Theorem even though the sample size was “small” (i.e.,  $n < 30$ ). It turns out, that how “far off” the  $t$ -test is can be computed using a first-order Edgeworth approximation for the error. Below, we will do this for the the further observations.

(a) Boos and Hughes-Oliver (2000) note that

$$P(T \leq t) \approx F_Z(t) + \underbrace{\frac{\text{skew}}{\sqrt{n}} \frac{(2t^2 + 1)}{6}}_{\text{error}} f_Z(t),$$

where  $f_Z(\cdot)$  and  $F_Z(\cdot)$  are the Gaussian PDF and CDF and skew is the skewness of the data. What is the potential error in the computation of the  $p$ -value when testing  $H_0 : \mu_X = 0; H_a : \mu_X < 0$  using the zebra finch further data?

```
library(e1071)
finch_data = read_csv("zebrafinches.csv")
# View(finch_data)
further = finch_data$further
n = length(further)
t = mean(further) / (sd(further)/sqrt(n))

error = (skewness(further) / sqrt(n)) * ((2*t^2 + 1) / 6) * dnorm(t)
error

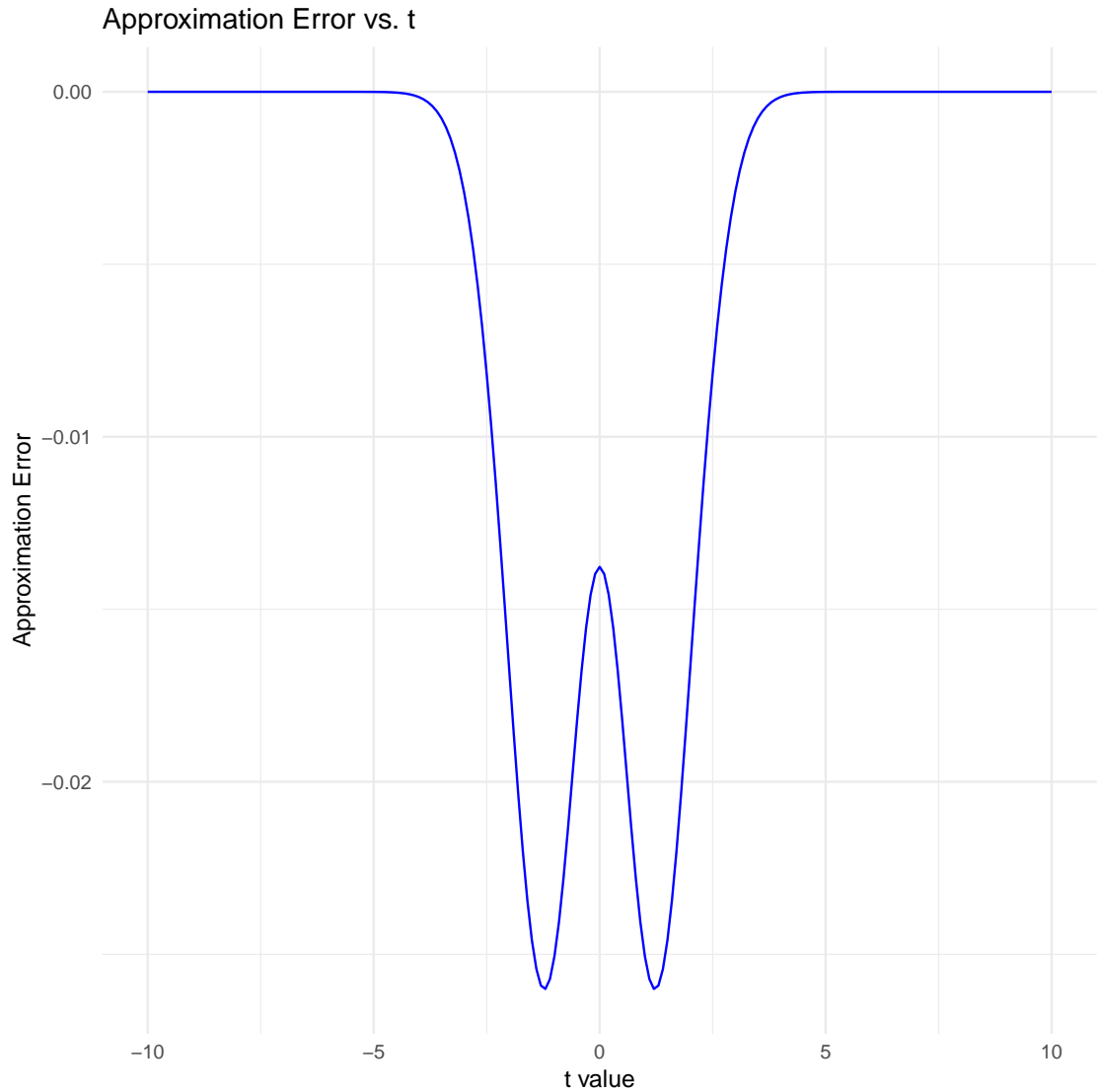
## [1] -1.226006e-13
```

The small error value ( $\text{error} \approx -1.226e - 13$ ) indicates almost no additional error suggesting that the  $t$ -test is reliable in this case.

- (b) Compute the error for  $t$  statistics from -10 to 10 and plot a line that shows the error across  $t$ . Continue to use the skewness and the sample size for the zebra finch further data.

```
library(ggplot2)
library(e1071)
finch_data = read_csv("zebrafinches.csv")
further = finch_data$further
n = length(further)
t_vals = seq(-10, 10, by = 0.1)
error_vals = (skewness(further) / sqrt(n)) * ((2*t_vals^2 + 1) / 6) * dnorm(t_vals)
# error_vals
df = data.frame(t_vals, error_vals)

ggplot(df, aes(x = t_vals, y = error_vals)) +
  geom_line(color = "blue") +
  labs(title = "Approximation Error vs. t",
       x = "t value",
       y = "Approximation Error") +
  theme_minimal()
```



The graph shows how much skewness might affect the approximation error at various  $t$ -values. The two peaks of errors show where the  $t$ -test accuracy is the worst.

- (c) Suppose we wanted to have a tail probability within 10% of the desired  $\alpha = 0.05$ . Recall we did a left-tailed test using the further data. How large of a sample size would we need? That is, we need to solve the error formula equal to 10% of the desired left-tail probability:

$$0.10\alpha \stackrel{set}{=} \underbrace{\frac{\text{skew}}{\sqrt{n}} \frac{(2t^2 + 1)}{6} f_Z(t)}_{\text{error}},$$

which yields

$$n = \left( \frac{\text{skew}}{6(0.10\alpha)} (2t^2 + 1) f_Z(t) \right)^2.$$

```
library(e1071)

finch_data = read_csv("zebrafinches.csv")
```

```

# View(finch_data)
further = finch_data$further
alpha = 0.05
t = qnorm(alpha)
skewness(further)

## [1] -1.035507

n = ((skewness(further)/(6*0.1*0.05)) * (2*t^2 + 1) * dnorm(t))^2
n

## [1] 520.8876

```

For the t-test's approximation to be accurate within 10% of the p-value, we need 521 samples.

2. Complete the following steps to revisit the analyses from lab 11 using the bootstrap procedure.
  - (a) Now, consider the zebra finch data. We do not know the generating distributions for the closer, further, and difference data, so perform resampling to approximate the sampling distribution of the  $T$  statistic:

$$T = \frac{\bar{x}_r - 0}{s/\sqrt{n}},$$

where  $\bar{x}_r$  is the mean computed on the  $r^{th}$  resample and  $s$  is the sample standard deviation from the original samples. At the end, create an object called `resamples.null.closer`, for example, and store the resamples shifted to ensure they are consistent with the null hypotheses at the average (i.e., here ensure the shifted resamples are 0 on average, corresponding to  $t = 0$ , for each case).

```

library(e1071)

finch_data = read_csv("zebrafinches.csv")
further = finch_data$further
closer = finch_data$closer
diff = finch_data$diff
n = nrow(finch_data)
m0 = 0

set.seed(123)
R <- 10000
further_resamples <- tibble(t.stats=rep(NA, R))
closer_resamples = tibble(t.stats=rep(NA, R))
diff_resamples = tibble(t.stats=rep(NA, R))
for(i in 1:R){
  new_further_resample = sample(further,
                                size = n,
                                replace = T)

  new_closer_resample = sample(closer,
                                size = n,
                                replace = T)

  new_diff_resample = sample(diff,
                              size = n,
                              replace = T)
}

```

```
further_resamples$t.stats[i] <- mean(new_further_resample)/(sd(further)/sqrt(n))
closer_resamples$t.stats[i] = mean(new_closer_resample)/(sd(closer)/sqrt(n))
diff_resamples$t.stats[i] = mean(new_diff_resample)/(sd(diff)/sqrt(n))
}
# View(further_resamples)

resamples.null.further = further_resamples$t.stats - mean(further_resamples$t.stats)
resamples.null.closer = closer_resamples$t.stats - mean(closer_resamples$t.stats)
resamples.null.diff = diff_resamples$t.stats - mean(diff_resamples$t.stats)
```

- (b) Compute the bootstrap  $p$ -value for each test using the shifted resamples. How do these compare to the  $t$ -test  $p$ -values?

```
## [1] 0
## [1] 0
## [1] 0
```

- (c) What is the 5<sup>th</sup> percentile of the shifted resamples under the null hypothesis? Note this value approximates  $t_{0.05, n-1}$ . Compare these values in each case.

```
## [1] "-1.68480212921038 is the 5th percentile of the shifted further data resamples"
## [1] "-1.59314617071328 is the 5th percentile of the shifted closer data resamples"
## [1] "-1.56409157711738 is the 5th percentile of the shifted diff data resamples"
```

- (d) Compute the bootstrap confidence intervals using the resamples. How do these compare to the  $t$ -test confidence intervals?

```
library(e1071)

finch_data = read_csv("zebrafinches.csv")
further = finch_data$further
closer = finch_data$closer
diff = finch_data$diff
n = nrow(finch_data)
m0 = 0

set.seed(123)
R <- 10000
further_resamples <- tibble(t.stats=rep(NA, R))
closer_resamples = tibble(t.stats=rep(NA, R))
diff_resamples = tibble(t.stats=rep(NA, R))
for(i in 1:R){
  new_further_resample = sample(further,
                                size = n,
                                replace = T)

  new_closer_resample = sample(closer,
                                size = n,
                                replace = T)

  new_diff_resample = sample(diff,
                              size = n,
                              replace = T)
```

```

further_resamples$t.stats[i] <- mean(new_further_resample)/(sd(further)/sqrt(n))
closer_resamples$t.stats[i] = mean(new_closer_resample)/(sd(closer)/sqrt(n))
diff_resamples$t.stats[i] = mean(new_diff_resample)/(sd(diff)/sqrt(n))
}
# View(further_resamples)

resamples.null.further = further_resamples$t.stats - mean(further_resamples$t.stats)
resamples.null.closer = closer_resamples$t.stats - mean(closer_resamples$t.stats)
resamples.null.diff = diff_resamples$t.stats - mean(diff_resamples$t.stats)

CI_further = quantile(resamples.null.further, c(0.025, 0.975))
CI_closer = quantile(resamples.null.closer, c(0.025, 0.975))
CI_diff = quantile(resamples.null.diff, c(0.025, 0.975))

CI_further_t = mean(further) + c(-1, 1) * qt(0.975, df = n-1) * sd(further)/sqrt(n)
CI_closer_t = mean(closer) + c(-1, 1) * qt(0.975, df = n-1) * sd(closer)/sqrt(n)
CI_diff_t = mean(diff) + c(-1, 1) * qt(0.975, df = n-1) * sd(diff)/sqrt(n)
#Should I use resamples.null.data or just the data
CI_further

##      2.5%      97.5%
## -2.033830  1.826771

CI_further_t

## [1] -0.2565176 -0.1489313

CI_closer

##      2.5%      97.5%
## -1.878078  1.926030

CI_closer_t

## [1] 0.1173875 0.1950586

CI_diff

##      2.5%      97.5%
## -1.857155  2.032365

CI_diff_t

## [1] 0.2719028 0.4459921

```

3. Complete the following steps to revisit the analyses from lab 11 using the randomization procedure.

- (a) Now, consider the zebra finch data. We do not know the generating distributions for the closer, further, and difference data, so perform the randomization procedure

```

randomization_test = function(data, mu0 = 0, R = 10000){
  x_shift = data-m0
  rand = tibble(xbars = rep(NA, R))

  for (i in 1:R){
    curr.rand = x_shift * sample(c(-1,1),
                                length(x_shift),
                                replace = T)

```

```

    rand$xbars[i] = mean(curr.rand)
  }

  rand = rand |>
    mutate(
      xbars = xbars + mu0
    )
  return(rand$xbars)
}

finch_data = read_csv("zebrafinches.csv")

rand_further = randomization_test(finch_data$further)
rand_closer = randomization_test(finch_data$closer)
rand_diff = randomization_test(finch_data$diff)

```

(b) Compute the randomization test  $p$ -value for each test.

```

randomization_test = function(data, mu0 = 0, R = 10000){
  x_shift = data-mu0
  rand = tibble(xbars = rep(NA, R))

  for (i in 1:R){
    curr.rand = x_shift * sample(c(-1,1),
                                size = length(x_shift),
                                replace = T)
    rand$xbars[i] = mean(curr.rand)
  }

  rand = rand |>
    mutate(
      xbars = xbars + mu0
    )
  return(rand$xbars)
}

finch_data = read_csv("zebrafinches.csv")

rand_further = randomization_test(finch_data$further)
rand_closer = randomization_test(finch_data$closer)
rand_diff = randomization_test(finch_data$diff)

pval_randomization = function(data, rand_samples, mu0 = 0){
  delta = abs(mean(data) - mu0)
  low = mu0 - delta
  high = mu0 + delta
  mean(rand_samples <= low) + mean(rand_samples >= high)
}

pval_further = pval_randomization(finch_data$further, rand_further)
pval_closer = pval_randomization(finch_data$closer, rand_closer)
pval_diff = pval_randomization(finch_data$diff, rand_diff)

pval_further

```

```
## [1] 0
pval_closer
## [1] 0
pval_diff
## [1] 0
```

- (c) Compute the randomization confidence interval by iterating over values of  $\mu_0$ .

**Hint:** You can “search” for the lower bound from  $Q_1$  and subtracting by 0.0001, and the upper bound using  $Q_3$  and increasing by 0.0001. You will continue until you find the first value for which the two-sided  $p$ -value is greater than or equal to 0.05.

```
## [1] -0.2627244 -0.1427244
## [1] 0.1162231 0.1962231
## [1] 0.2689475 0.4589475
```

4. **Optional Challenge:** In this lab, you performed resampling to approximate the sampling distribution of the  $T$  statistic using

$$T = \frac{\bar{x}_r - 0}{s/\sqrt{n}}.$$

I’m curious whether it is better/worse/similar if we computed the statistics using the sample standard deviation of the resamples ( $s_r$ ), instead of the original sample ( $s$ )

$$T = \frac{\bar{x}_r - 0}{s_r/\sqrt{n}}.$$

- Perform a simulation study to evaluate the Type I error for conducting this hypothesis test both ways.
- Using the same test case(s) as part (a), compute bootstrap confidence intervals and assess their coverage – how often do we ‘capture’ the parameter of interest?

## References

Boos, D. D. and Hughes-Oliver, J. M. (2000). How large does  $n$  have to be for  $z$  and  $t$  intervals? *The American Statistician*, 54(2):121–128.