

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

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

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(tidyverse)
finches.dat <- read_csv("zebrafinches.csv")

## Rows: 25 Columns: 3
## -- Column specification -----
## Delimiter: ","
## dbl (3): closer, further, diff
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.

alpha <- 0.05
t <- t.test(x = finches.dat$further,
            alternative = "two.sided",
            mu = 0,
            conf.level = 0.95)

t.score.further <- -7.778

n <- length(finches.dat$further)

library(e1071)
further.skew <- skewness(finches.dat$further)

f.of.z <- dnorm(t.score.further, 0, 1)

F.of.z <- pnorm(t.score.further, 0, 1)

numerator.a <- further.skew / sqrt(n)
numerator.b <- (2*(t.score.further*t.score.further) + 1)/6

ans <- F.of.z + numerator.a*numerator.b*f.of.z
print(ans)

## [1] -1.189085e-13
```

- (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.

```

t.vals <- seq(-10, 10, length.out = 1000)

calculate.error <- function(t.val){
  skew <- skewness(finches.dat$further)
  n <- length(finches.dat$further)

  f.of.z <- dnorm(t.val,0,1)
  F.of.z <- pnorm(t.val,0,1)

  numerator.a <- skew / sqrt(n)
  numerator.b <- (2*(t.val*t.val) + 1)/6

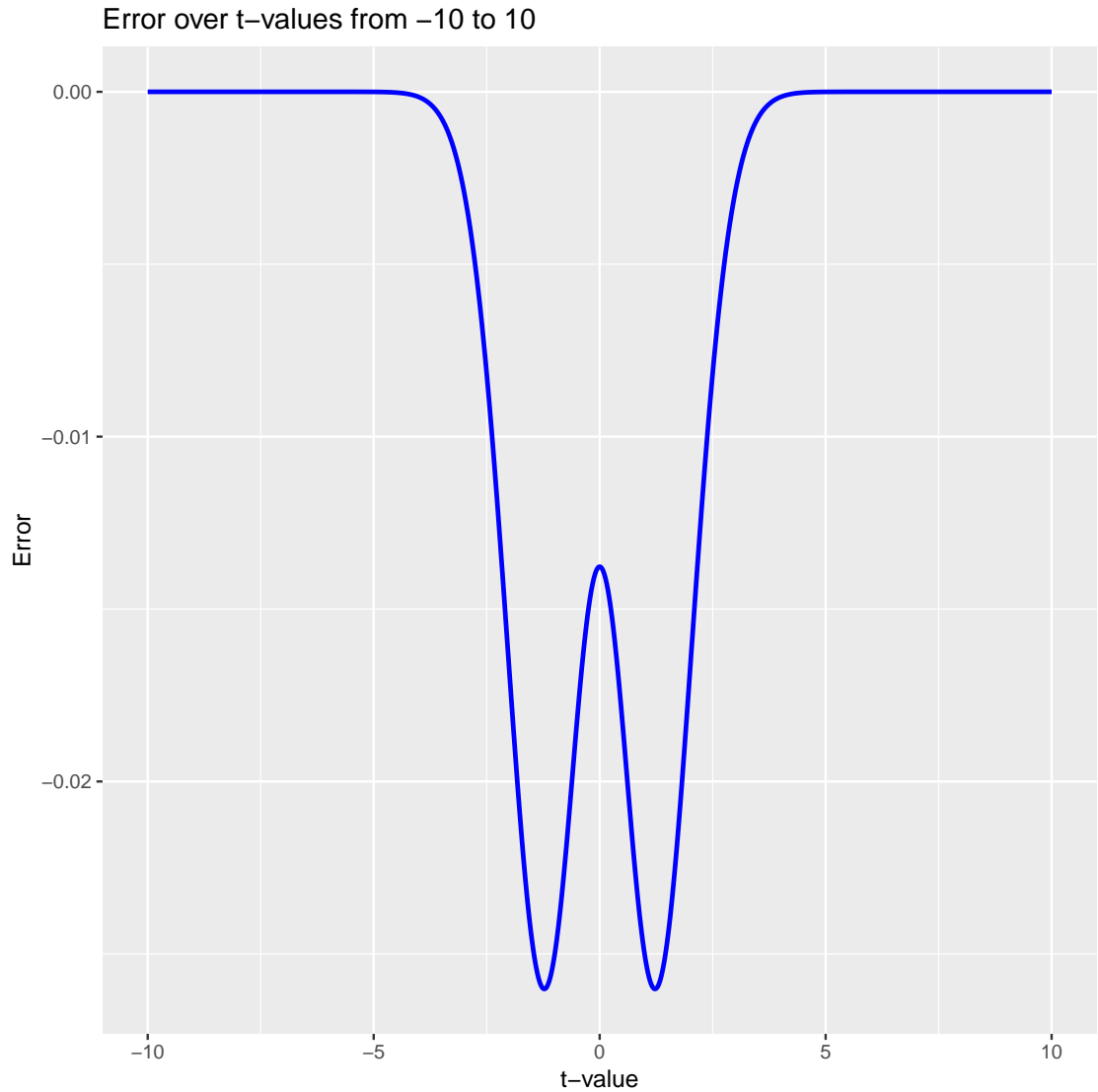
  ans <- numerator.a*numerator.b*f.of.z
  return(ans)
}

error.vals <- numeric(1000)
for (i in 0:length(t.vals)){
  t.val <- t.vals[i]
  error <- calculate.error(t.val)
  error.vals[i] = error
}

#plot the results
plot.data <- tibble(
  t = t.vals,
  metric = error.vals
)

ggplot(plot.data, aes(x = t, y = metric)) +
  geom_line(color = "blue", linewidth = 1.0) +
  labs(title = "Error over t-values from -10 to 10", x = "t-value", y = "Error")

```



- (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:

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

which yields

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

```
skew <- skewness(finches.dat$further)

alpha <- .05

t<- qnorm(alpha,0,1)
```

```
f.of.z <- dnorm(t,0,1)

part.a <- skew/ (6*(.10*alpha))
part.b <- ((2*t*t) + 1) * f.of.z

n <- (part.a*part.b)*(part.a*part.b)
print(n)

## [1] 520.8876
```

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).

```
R <- 1000
mu0 <- 0

resamples.closer <- tibble(t = numeric(R))
for(i in 1:R){
  curr.sample <- sample(x= finches.dat$closer, size = length(finches.dat$closer), replace = T)
  resamples.closer$t[i] <- (mean(curr.sample) - mu0)/(sd(finches.dat$closer)/sqrt(length(finches
)})

resamples.further <- tibble(t = numeric(R))
for(i in 1:R){
  curr.sample <- sample(x= finches.dat$further, size = length(finches.dat$further), replace = T)
  resamples.further$t[i] <- (mean(curr.sample) - mu0)/(sd(finches.dat$further)/sqrt(length(finches
)})

resamples.difference <- tibble(t = numeric(R))
for(i in 1:R){
  curr.sample <- sample(x= finches.dat$diff, size = length(finches.dat$diff), replace = T)
  resamples.difference$t[i] <- (mean(curr.sample) - mu0)/(sd(finches.dat$diff)/sqrt(length(finches
)})

mean.t.closer <- mean(resamples.closer$t)
print(mean.t.closer)

## [1] 8.287307

mean.t.further <- mean(resamples.further$t)
print(mean.t.further)

## [1] -7.762185

mean.t.difference <- mean(resamples.difference$t)
print(mean.t.difference)
```

```
## [1] 8.533112
```

The plot shows how the error in p-value calculation changes with t-statistics, highlighting that larger t-values tend to result in more pronounced errors.

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

```
library(boot)

t.obs.closer <- (mean(finches.dat$closer) - 0) / (sd(finches.dat$closer)/sqrt(length(finches.d
t.obs.further <- (mean(finches.dat$further) - 0) / (sd(finches.dat$further)/sqrt(length(finches
t.obs.difference <- (mean(finches.dat$diff) - 0) / (sd(finches.dat$diff)/sqrt(length(finches.d

closer.shifted <- t.obs.closer - mean.t.closer
further.shifted <- t.obs.further - mean.t.further
diff.shifted <- t.obs.difference - mean.t.difference

closer.shifted.p <- mean(resamples.closer$t >= closer.shifted)
further.shifted.p <- mean(resamples.further$t >= further.shifted)
diff.shifted.p <- mean(resamples.difference$t >= diff.shifted)

print(closer.shifted.p)

## [1] 1

print(further.shifted.p)

## [1] 0

print(diff.shifted.p)

## [1] 1
```

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

```
percentile.closer <- quantile(resamples.closer$t, 0.05)
percentile.further <- quantile(resamples.further$t, 0.05)
percentile.difference <- quantile(resamples.difference$t, 0.05)

print(percentile.closer)

##          5%
## 6.782656

print(percentile.further)

##          5%
## -9.383775

print(percentile.difference)

##          5%
## 6.976885

critical.value <- qt(0.05, df = length(finches.dat$closer) - 1)
print(critical.value)

## [1] -1.710882
```

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

```
ci.closer <- quantile(resamples.closer$t, c(0.025, 0.975))
ci.further <- quantile(resamples.further$t, c(0.025, 0.975))
ci.difference <- quantile(resamples.difference$t, c(0.025, 0.975))

print(ci.closer)

##      2.5%      97.5%
## 6.508614 10.174648

print(ci.further)

##      2.5%      97.5%
## -9.691517 -6.022896

print(ci.difference)

##      2.5%      97.5%
## 6.752102 10.541812

alpha <- 0.05
t.critical <- qt(1 - alpha/2, df = length(finches.dat$closer) - 1)

ci.t.closer <- mean(finches.dat$closer) + c(-1, 1) * t.critical * (sd(finches.dat$closer) / sqrt(n))
ci.t.further <- mean(finches.dat$further) + c(-1, 1) * t.critical * (sd(finches.dat$further) / sqrt(n))
ci.t.difference <- mean(finches.dat$diff) + c(-1, 1) * t.critical * (sd(finches.dat$diff) / sqrt(n))

print(ci.t.closer)

## [1] 0.1173875 0.1950586

print(ci.t.further)

## [1] -0.2565176 -0.1489313

print(ci.t.difference)

## [1] 0.2719028 0.4459921
```

The bootstrap confidence intervals are wider than the t -test intervals, suggesting higher uncertainty when resampling the data.

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

```
R <- 1000

randomization.closer <- numeric(R)
randomization.further <- numeric(R)

randomize_test_statistic <- function(data1, data2) {
  combined_data <- c(data1, data2)

  shuffled_data <- sample(combined_data)
```

```

group1 <- shuffled_data[1:length(data1)]
group2 <- shuffled_data[(length(data1)+1):length(shuffled_data)]

return(abs(mean(group1) - mean(group2)))
}

for(i in 1:R) {
  randomization.closer[i] <- randomize_test_statistic(finches.dat$closer, finches.dat$further)
  randomization.further[i] <- randomize_test_statistic(finches.dat$further, finches.dat$closer)
}

observed.closer <- abs(mean(finches.dat$closer) - mean(finches.dat$further))
observed.further <- abs(mean(finches.dat$further) - mean(finches.dat$closer))

p.value.closer <- mean(randomization.closer >= observed.closer)
p.value.further <- mean(randomization.further >= observed.further)

print(p.value.closer)

## [1] 0

print(p.value.further)

## [1] 0

```

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

```

set.seed(123)
R <- 10000
obs.stat <- mean(finches.dat$closer) - mean(finches.dat$further)
combined <- c(finches.dat$closer, finches.dat$further)
n1 <- length(finches.dat$closer)
rand.diff <- numeric(R)

for (i in 1:R) {
  shuffled <- sample(combined)
  group1 <- shuffled[1:n1]
  group2 <- shuffled[(n1 + 1):length(shuffled)]
  rand.diff[i] <- mean(group1) - mean(group2)
}

p.val.diff <- mean(abs(rand.diff) >= abs(obs.stat))
print(p.val.diff)

## [1] 0

obs.closer <- mean(finches.dat$closer)
rand.closer <- numeric(R)

for (i in 1:R) {
  signs <- sample(c(-1, 1), length(finches.dat$closer), replace = TRUE)
  rand.closer[i] <- mean(signs * finches.dat$closer)
}

p.val.closer <- mean(abs(rand.closer) >= abs(obs.closer))
print(p.val.closer)

```

```
## [1] 0

obs.further <- mean(finches.dat$further)
rand.further <- numeric(R)

for (i in 1:R) {
  signs <- sample(c(-1, 1), length(finches.dat$further), replace = TRUE)
  rand.further[i] <- mean(signs * finches.dat$further)
}

p.val.further <- mean(abs(rand.further) >= abs(obs.further))
print(p.val.further)

## [1] 0
```

Both tests give a p-value of 0, which suggests very strong evidence against the null hypothesis in both cases.

- (c) Compute the randomization confidence interval by iterating over values of μ_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.

```
set.seed(123)
R <- 1000
x <- finches.dat$closer
n <- length(x)

compute_pval <- function(mu0) {
  obs <- mean(x) - mu0
  null_dist <- numeric(R)

  for (i in 1:R) {
    signs <- sample(c(-1, 1), n, replace = TRUE)
    shifted_sample <- (x - mu0) * signs + mu0
    null_dist[i] <- mean(shifted_sample) - mu0
  }

  mean(abs(null_dist) >= abs(obs))
}

q1 <- mean(x) - sd(x) / sqrt(n)
q3 <- mean(x) + sd(x) / sqrt(n)

mu.lower <- q1
while (compute_pval(mu.lower) < 0.05) {
  mu.lower <- mu.lower - 0.0001
}

mu.upper <- q3
while (compute_pval(mu.upper) < 0.05) {
  mu.upper <- mu.upper + 0.0001
}

print(mu.lower)
```



```
## [1] 0.1374065
print(mu.upper)
## [1] 0.1750397
```

This confidence interval represents the range within which the true mean difference is likely to fall with 95

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)

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

- (a) Perform a simulation study to evaluate the Type I error for conducting this hypothesis test both ways.
- (b) 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.