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?
   Solution: The potential error in the computation of the p-value is $-1.226006e - 13$

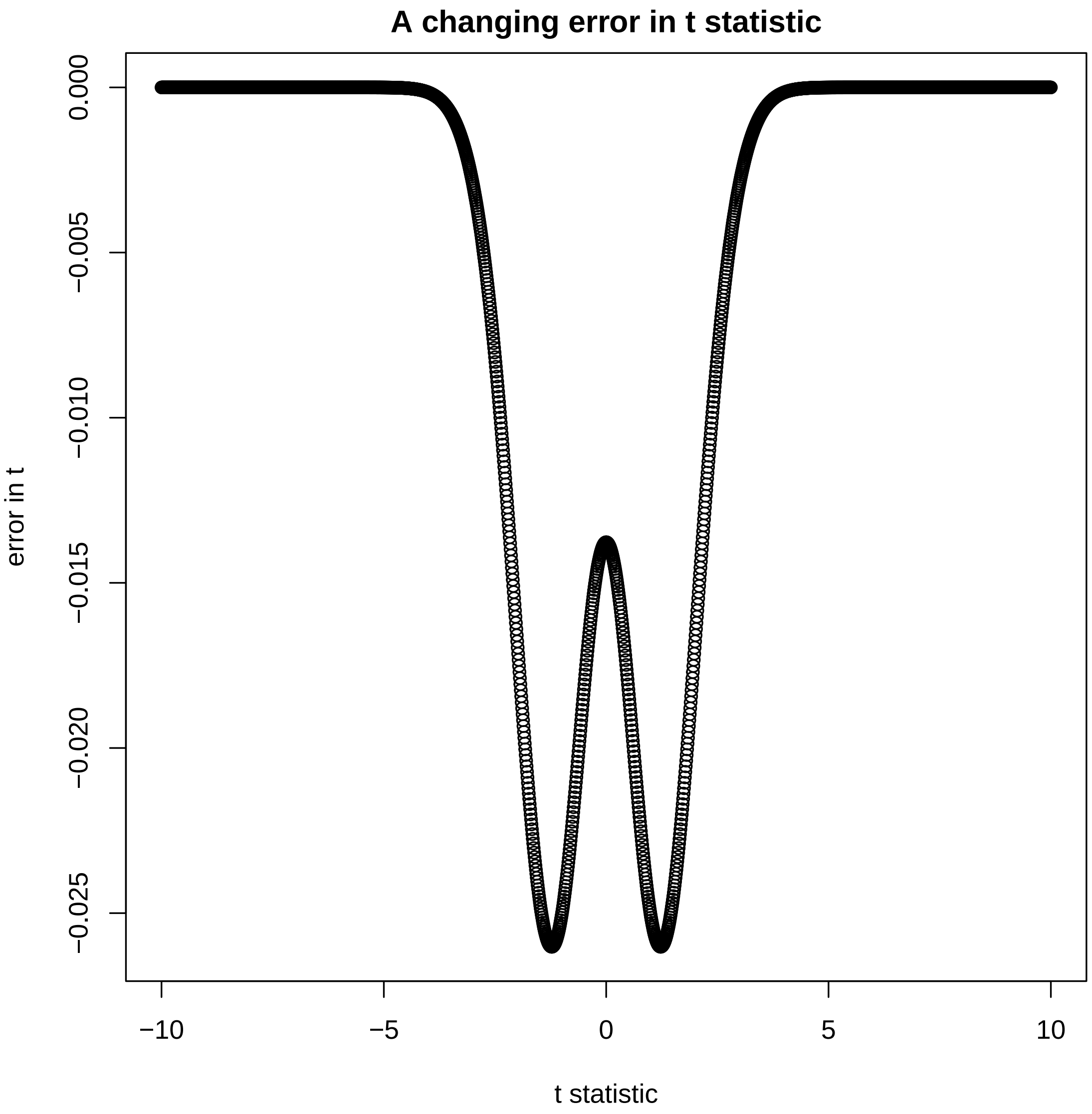
```
finch.dat = read.csv(’./zebrafinches.csv’)

#potential error in p-value
n = length(finch.dat$further)
t.further = t.test(finch.dat$further, alternative = ’less’,
                 mu = 0)$statistic[1]
gauss.pdf  = dnorm(t.further)
finch.skew = skewness(finch.dat$further)

error = gauss.pdf*finch.skew*(2*t.further^2 + 1)/(sqrt(n)*6)
```

   (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.
   Solution:

```
#t-statistic error from -10 to 10
t.errors = c()
for (i in seq(-10, 10, by = 0.01)){
t = i
gauss.pdf  = dnorm(t)
error = gauss.pdf*finch.skew*(2*t^2 + 1)/(sqrt(n)*6)
t.errors = append(t.errors, error)
}
par(mar = c(4, 4, 2, 1))
plot(seq(-10, 10, by = 0.01), t.errors,
   xlab = ’t statistic’,
   ylab = ’error in t’,
   main = ’A changing error in t statistic’)
```



(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:

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

which yields

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

Solution: Calculating n with the given formula, we get a value very close to zero (on the order of 10^-20. That means that our data is so far from the null hypothesis of 0, that any datapoint over here will be beyond that 0.05 threshold.

```
#Sample size needed
a= 0.05
gauss.pdf  = dnorm(t.further)
n = ((finch.skew/(6*0.10*a))*(2*t.further^2 + 1)*gauss.pdf)^2
```

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 rth 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).

   Solution: In the below code, the mean of the shifted `resamples.null.closer` is approximately 0

```
R <- 10000
resamples <- tibble(xbars = rep(NA, R))

for(i in 1:R){
curr.resample <- sample(finch.dat$‘closer‘,
                        size = nrow(finch.dat),
                        replace = T)

resamples$xbars[i] <- mean(curr.resample)
}
#Shifted Resamples
delta = mean(resamples$xbars)
resamples.null.closer = data.frame(xbars.shifted = resamples$xbars - delta)
```

   (b) Compute the bootstrap p-value for each test using the shifted resamples.

   Solution: We calculated the bootstrap p-value with the below code. The output for the closer data is 1e-04, the further data is 1e-04, and the difference data is also 1e-04. This is probably due to an error in the code not working correctly, which I did not have time ot rectify before the submission deadline.

```
####Further####
R <- 10000
resamples <- tibble(xbars = rep(NA, R))

for(i in 1:R){
curr.resample <- sample(finch.dat$‘further‘,
                        size = nrow(finch.dat),
                        replace = T)

resamples$xbars[i] <- mean(curr.resample)
}


# Confidence Interval
quantile(resamples$xbars, c(0.025, 0.975))

library(boot)
boot.mean <- function(d, i){
mean(d[i])
}
boots <- boot(data = finch.dat$‘further‘,
            statistic = boot.mean,
            R = R)
boot.ci(boots, type="bca")
```

```
# shift so H0 is true
mean(resamples$xbars)
(delta <- mean(resamples$xbars) - 2.9)


resamples <- resamples |>
mutate(xbars.shifted = xbars - delta)

low <- 2.9 - delta
high <- 2.9 + delta

resamples |>
summarize(mean = mean(xbars.shifted),
          p.low = mean(xbars.shifted <= low),
          p.high = mean(xbars.shifted >= high))|>
mutate(p = p.low + p.high) |>
view()


boot.pval(boots, theta_null = mu0) # this does something slightly different
```

How do these compare to the t-test p-values?

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

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

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
Solution: We did the following randomization procedure for each of the closer, further, and difference zebra finch data.

```
mu0 = 0
#Closer
R <- 10000
rand <- tibble(xbars = rep(NA, R))

# PREPROCESSING: shift the data to be mean 0 under H0
x.shift <- finch.dat$‘closer‘ - mu0
# RANDOMIZE / SHUFFLE
for(i in 1:R){
  curr.rand <- x.shift *
    sample(x = c(-1, 1),
           size = length(x.shift),
           replace = T)

  rand$xbars[i] <- mean(curr.rand)
}
# Thinking is hard
rand <- rand |>
  mutate(xbars = xbars + mu0) # shifting back
```

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

Solution: To compute the randomization p-value for each text, we used the following code:

```
# p-value
(delta <- abs(mean(finch.dat$‘closer‘) - mu0))
(low <- mu0 - delta) # mirror
(high<- mu0 + delta)   # xbar
```

```
mean(rand$xbars <= low) +
mean(rand$xbars >= high)
```

The result is a p value that is effectively 0, which checks out with our other texts (ex, a plain t test) which have a p value on order of $10^{-}8$.

(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.
Solution: For the closer data, the confidence interval is: 0.1162231 to 0.1962231. For the further data, it is -0.2627244 to -0.1427244. Finally, for the difference data, the confidence interval is 0.2689475 to 0.4489475. These intervals were found using the below code for the randomization test.

```
#Confidence Interval
mu0.iterate <- 0.01
starting.point <- mean(finch.dat$‘closer‘)

mu.lower <- starting.point
repeat{
rand <- tibble(xbars = rep(NA, R))

# PREPROCESSING: shift the data to be mean 0 under H0
x.shift <- finch.dat$‘closer‘ - mu.lower
# RANDOMIZE / SHUFFLE
for(i in 1:R){
  curr.rand <- x.shift *
    sample(x = c(-1, 1),
           size = length(x.shift),
           replace = T)

  rand$xbars[i] <- mean(curr.rand)
}
# Thinking is hard
rand <- rand |>
  mutate(xbars = xbars + mu.lower) # shifting back

# p-value
(delta <- abs(mean(finch.dat$‘closer‘) - mu.lower))
(low <- mu.lower - delta) # mirror
(high<- mu.lower + delta)   # xbar
(p.val <- mean(rand$xbars <= low) +
    mean(rand$xbars >= high))

if(p.val < 0.05){
  break
}else{
  mu.lower <- mu.lower - mu0.iterate
}
}

mu.upper <- starting.point
repeat{
rand <- tibble(xbars = rep(NA, R))

# PREPROCESSING: shift the data to be mean 0 under H0
x.shift <- finch.dat$‘closer‘ - mu.upper
# RANDOMIZE / SHUFFLE
for(i in 1:R){
  curr.rand <- x.shift *
    sample(x = c(-1, 1),
           size = length(x.shift),
           replace = T)

  rand$xbars[i] <- mean(curr.rand)
}
# Thinking is hard
rand <- rand |>
  mutate(xbars = xbars + mu.upper) # shifting back

# p-value
(delta <- abs(mean(finch.dat$‘closer‘) - mu.upper))
(low <- mu.upper - delta) # mirror
(high<- mu.upper + delta)   # xbar
(p.val <- mean(rand$xbars <= low) +
    mean(rand$xbars >= high))

if(p.val < 0.05){
  break
}else{
  mu.upper <- mu.upper + mu0.iterate
}
}

c(mu.lower, mu.upper)
```

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.