

PS1

Shuai Yan
2023-09-25

Problem Set 1: Understanding Type I Error Rates

To create reproducible results, set random seed to 1.

```
set.seed(1)
```

Part I

Type I error is the error of rejecting a null hypothesis even though it is true. Given that the compound really has no effect, the result values should come from the same normal distribution. Here, we assume that the results come from a standard normal distribution.

```
placebo_group <- rnorm(n=10, mean=0, sd=1)
compound_group <- rnorm(n=10, mean=0, sd=1)
```

To show the Type I error rate, we need to do this experiment `repeat_num` times and see how many times we would reject the null hypothesis, given in `error_num`. Type I error rate would then be `error_num / repeat_num`.

```
repeat_number <- 10000 # set the number of experiments
error_num <- 0 # initial the count for errors
alpha <- 0.05

for (i in 1:repeat_number) {
  placebo_group <- rnorm(n=10, mean=0, sd=1)
  compound_group <- rnorm(n=10, mean=0, sd=1)
  result <- t.test(placebo_group, compound_group)$p.value < alpha # Is result significant?
  if (result) {
    error_num <- error_num + 1 # when Type I error happens, error_num increase by one
  }
}

error_rate <- error_num / repeat_number
sprintf("Type I error rate is: %.3f%%", 100 * error_rate)
```

```
## [1] "Type I error rate is: 5.060%"
```

Part II

Based on Part I, add 10 mice when having a statistically insignificant result from an experiment with 10 mice per group.

```
repeat_number <- 10000 # set the number of experiments
error_num <- 0 # initial the count for errors
mod_error_num <- 0 # initial the count for errors after modify the experiment
alpha <- 0.05

for (i in 1:repeat_number) {
  placebo_group <- rnorm(n=10, mean=0, sd=1)
  compound_group <- rnorm(n=10, mean=0, sd=1)
  result <- t.test(placebo_group, compound_group)$p.value < alpha # Is result significant?
  if (!result) {
    # when result is not statistically significant, add 10 sample per group
    mod_placebo_group <- append(placebo_group, rnorm(n=10, mean=0, sd=1))
    mod_compound_group <- append(compound_group, rnorm(n=10, mean=0, sd=1))
    mod_result <- t.test(mod_placebo_group, mod_compound_group)$p.value < alpha # Is result significant after modification?
    if (mod_result) {
      mod_error_num <- mod_error_num + 1 # error occurs after modify the experiment
    }
  } else {
    # error happens before adding sample
    error_num <- error_num + 1
    mod_error_num <- mod_error_num + 1
  }
}

true_error_rate <- error_num / repeat_number
mod_error_rate <- mod_error_num / repeat_number
sprintf("The true Type I error rate is: %.3f", 100 * true_error_rate)
```

```
## [1] "The true Type I error rate is: 4.790"
```

```
sprintf("The Type I error rate after adding more samples is: %.3f%%", 100 * mod_error_rate)
```

```
## [1] "The Type I error rate after adding more samples is: 8.390%"
```

```
sprintf("This increased the Type I error rate by: %.3f%%", 100 * (mod_error_rate - true_error_rate) / true_error_rate)
```

```
## [1] "This increased the Type I error rate by: 75.157%"
```

Part III

We first wrap up the functionalities we have implemented to a function: `err_rate`. This function takes input of: `alpha` (level of significance), `sample_num` (the number at the start of the experiment), `add_num` (the number added after an insignificant result), and `repeat_num` (experiments carried out); and gives back the error rate as return values.

```
err_rate <- function(alpha, sample_num=10, add_num=10, repeat_num=10000) {
  error_num <- 0 # initial the count for errors

  for (i in 1:repeat_num) {
    sample_1 <- rnorm(n=2*sample_num, mean=0, sd=1) # only sample once for an experiment for efficiency improvement
    placebo_group <- sample_1[1 : sample_num]
    compound_group <- sample_1[(sample_num+1) : (2*sample_num)]
    result <- t.test(placebo_group, compound_group)$p.value < alpha # Is result significant?
    if (!result) {
      # when result is not statistically significant, add 10 sample per group
      sample_2 <- rnorm(n=2*add_num, mean=0, sd=1)
      mod_placebo_group <- append(placebo_group, sample_2[1 : add_num])
      mod_compound_group <- append(compound_group, sample_2[(add_num+1) : (2*add_num)])
      mod_result <- t.test(mod_placebo_group, mod_compound_group)$p.value < alpha # Is result significant after modification?
      if (mod_result) {
        error_num <- error_num + 1 # error occurs after modify the experiment
      }
    } else {
      # error happens before adding sample
      error_num <- error_num + 1
    }
  }

  return (error_num / repeat_num)
}
```

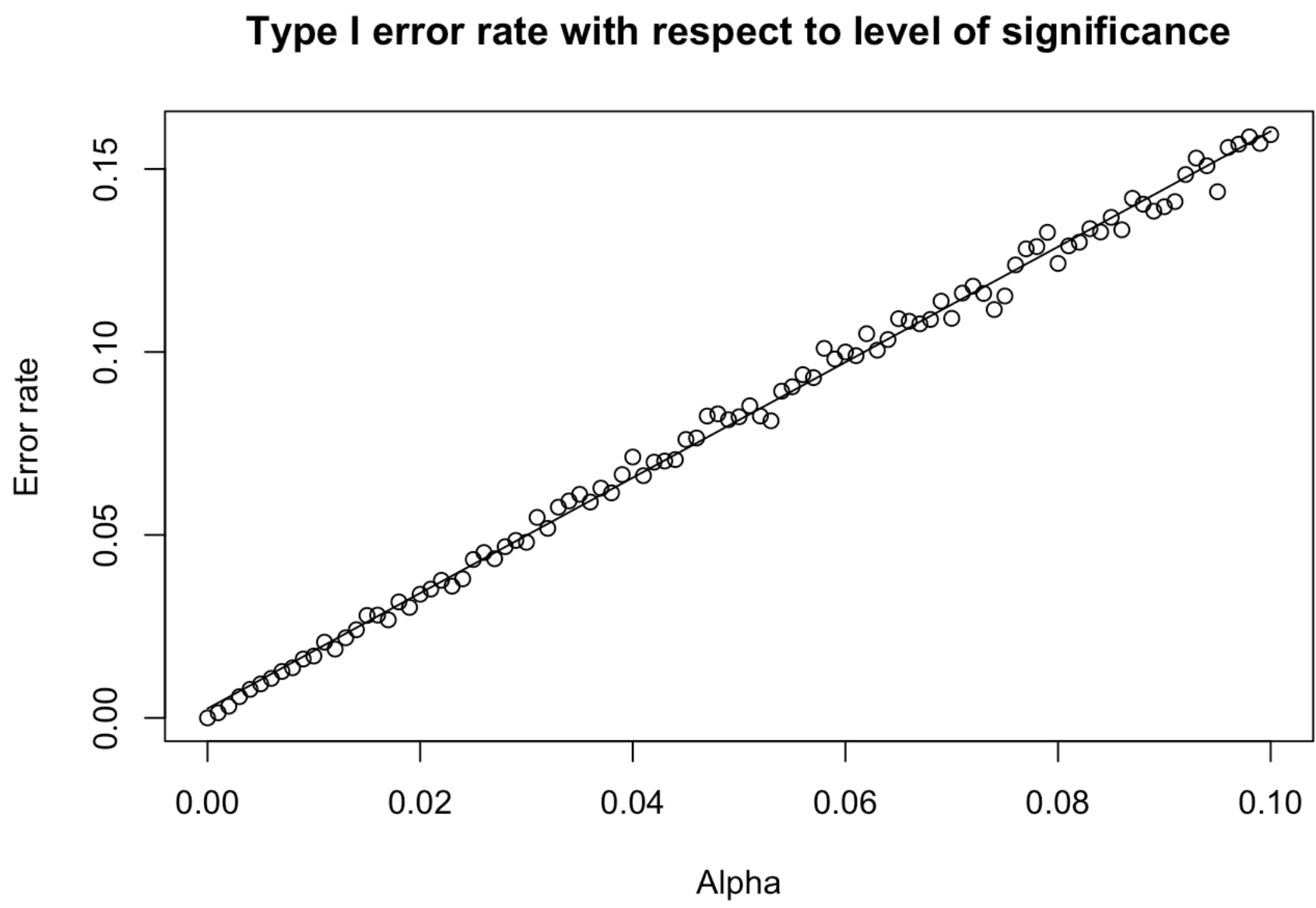
With this function, we can plot a figure of error rate according to level of significance.

```
alpha <- seq(from=0.0, to=0.1, by=0.001) # a sequence of alpha to test on the wrong protocol
error_rates <- alpha # initialize a vector of error rates with the same size of the alpha sequence

for (i in 1:length(alpha)) {
  error_rates[i] <- err_rate(alpha[i])
}

fit <- lm(error_rates~alpha) # curve fitting for error rate v.s. alpha

plot(alpha, error_rates,
     main="Type I error rate with respect to level of significance",
     xlab="Alpha",
     ylab="Error rate")
lines(alpha, predict(fit, data=data.frame(x=alpha))) # add fitting curve to figure
```



```
summary(fit)

##
## Call:
## lm(formula = error_rates ~ alpha)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.0086273 -0.0014787  0.0000504  0.0017889  0.0069477
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.0025455   0.0005621    4.529 1.66e-05 ***
## alpha       1.5777030   0.0097113   162.460 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.002845 on 99 degrees of freedom
## Multiple R-squared:  0.9963, Adjusted R-squared:  0.9962
## F-statistic: 2.639e+04 on 1 and 99 DF,  p-value: < 2.2e-16
```

From the previous simulation and fitting results, the relationship of Type I error rate and the level of significance following the wrong protocol is $P(\text{Type I error}) = 1.6 \cdot \alpha$. Therefore, to maintain an overall Type I error rate of 5%, α should be lower than 0.03.