

Homework 2 - BIOST 544

John Schoof

11/3/2020

Question 1

Below is a function that does a re-randomization of treatment assignments to evaluate if the treatment effect that is observed in our dataset is consistent with the null hypothesis that the standard of care is at least as effective as the new-treatment.

$$H_o : \text{effect}_T \leq \text{effect}_C$$

The below function called “simulate.perm.test” takes in a dataset with two columns, called “tx” and “outcome,” and returns a p-value corresponding to the above hypothesis test and a histogram showing where the observed difference lies in the sampling distribution. The function first runs an internal function called “perm.one.trial.” The “perm.one.trial” function copies the dataset to create a permuted dataset with which I then randomly permute the “tx” column. This shuffles the treatment assignments to our observed outcomes. The “perm.one.trial” function ends by calculating the difference in the proportion of patients that respond positively between treatment and control groups in this single permuted simulation. Still within the “simulate.perm.test” function, the “perm.one.trial” is then simulated 10,000 times to create a sampling distribution of simulated treatment effects. Finally, the function returns a p-value that is telling us how consistent our observed data is with this null sampling distribution, and it returns a histogram showing where our observed treatment effect lies in that sampling distribution.

```
set.seed(1)
simulate.perm.test <- function(dat){

  perm.one.trial <- function(dat){
    ## create the permuted data
    perm <- sample(0:1, nrow(dat), replace=TRUE)
    perm.data <- dat
    perm.data$tx <- perm

    ## calculate the difference in proportion responding positively
    prop.treat <- perm.data %>%
      filter(tx == 1) %>%
      summarise(prop = mean(outcome)) %>%
      .$prop
    prop.control <- perm.data %>%
      filter(tx == 0) %>%
      summarise(prop = mean(outcome)) %>%
      .$prop

    prop.diff <- prop.treat - prop.control

    return(prop.diff)
  }
}
```

```

}

# Simulate perm.one.trial 10,000 times to create sampling distribution
nsim <- 1e4
permuted.stats <- data.frame(replicate(nsim, perm.one.trial(dat)))
colnames(permuted.stats) <- c("prop.diff")

# Calculate observed difference in proportions
resp.prop.treat <- dat %>%
  filter(tx == 1) %>%
  summarise(prop = mean(outcome)) %>%
  .$prop
resp.prop.control <- dat %>%
  filter(tx == 0) %>%
  summarise(prop = mean(outcome)) %>%
  .$prop
resp.prop.diff <- resp.prop.treat - resp.prop.control

# calculate a p value that null is consistent with observed difference
pval <- 1 - mean(permuted.stats$prop.diff <= resp.prop.diff)

# Histogram of sampling distribution
hist <- ggplot(data = permuted.stats, mapping = aes(x=prop.diff)) +
  geom_histogram(binwidth=0.08) +
  geom_vline(aes(xintercept = resp.prop.diff), color = "red") +
  labs(x = "Difference in Outcome between Treatment and Control",
       y = "Count",
       title = "Sampling Distribution of Treatment Effect")

# return p value and histogram
return(list(pval, hist))
}

```

Question 2a

Similar to question 1, below is a function that does a re-randomization of treatment assignments to evaluate if the treatment effect that is observed in our dataset is consistent with the null hypothesis that the standard of care is at least as effective as the new-treatment. However, this function uses the adaptive randomization scheme as described by question 2.

The below function called “adaptive.random” takes in a dataset with three columns, one called “tx,” “outcome,” and “order,” and returns a p-value corresponding to the above hypothesis test and a histogram showing where the observed difference lies in the sampling distribution. As in question 1, there is an internal function that runs one trial of the permuted simulation that is called “adaptive.one.trial.” The main difference here is that within the “adaptive.one.trial” function a for loop is run in order to randomize patients one at a time. This for loop recalculates p_{new} for each patient that is assigned based on the outcomes of the previous patients. As in question 1, the “adaptive.one.trial” function ends by calculating the difference in the proportion of patients that respond positively between treatment and control groups in this single permuted simulation. The “adaptive.random” function then replicates this 10,000 times and returns a p-value and a histogram just like in question 1.

```

adaptive.random <- function(dat){

  adaptive.one.trial <- function(dat){
    success.new <- 0
    failure.old <- 0
    perm.data <- dat
    # calculate p.new using defined randomization scheme
    for(i in 1:nrow(dat)){
      p.new <- (1 + 3*(success.new + failure.old))/(2+3*(i-1))

      perm.data$tx[i] <- rbinom(1, 1, p.new)

      if (perm.data$tx[i]==1 & perm.data$outcome[i]==1){
        success.new <- (success.new+1) }
      if (perm.data$tx[i]==0 & perm.data$outcome[i]==0) {
        failure.old=failure.old+1}
    }

    prop.treat <- perm.data %>%
      filter(tx == 1) %>%
      summarise(prop = mean(outcome)) %>%
      .$prop
    prop.control <- perm.data %>%
      filter(tx == 0) %>%
      summarise(prop = mean(outcome)) %>%
      .$prop

    prop.diff2 <- prop.treat - prop.control

    return(prop.diff2)
  }

  # Simulate perm.one.trial 10,000 times to create sampling distribution
  nsim <- 1e4
  set.seed(1)
  permuted.2b <- data.frame(replicate(nsim, adaptive.one.trial(dat)))
  colnames(permuted.2b) <- c("prop.diff2")

  # Calculate observed difference in proportions
  resp.prop.treat <- dat %>%
    filter(tx == 1) %>%
    summarise(prop = mean(outcome)) %>%
    .$prop
  resp.prop.control <- dat %>%
    filter(tx == 0) %>%
    summarise(prop = mean(outcome)) %>%
    .$prop
  resp.prop.diff <- resp.prop.treat - resp.prop.control

  # calculate a p value that null is consistent with observed difference
  pval2 <- 1-mean(permuted.2b$prop.diff2 <= resp.prop.diff)

```

```

# Histogram
plot2 <- ggplot(data = permuted.2b, mapping = aes(x=prop.diff2)) +
  geom_histogram(binwidth=0.08) +
  geom_vline(aes(xintercept = resp.prop.diff), color = "red") +
  labs(x = "Difference in Outcome between Treatment and Control",
       y = "Count",
       title = "Sampling Distribution of Treatment Effect for Adaptive Randomization")

# Return p-value and histogram
return(list(pval2, plot2))
}

```

Question 2b

Below, the “adapt.random” function from question 2a is run on the “HW2-adaptive-trial.txt” dataset. This tests the null hypothesis that the standard of care is at least as effective as the new treatment. Returned is a p-value of 0.205. This means that we do not have evidence to reject the null hypothesis at the $\alpha = 0.05$ significance level. In other words, if the null hypothesis were true and the standard of care is in fact at least as effective as the new treatment, then seeing a our sample wouldn’t be that unusual.

```

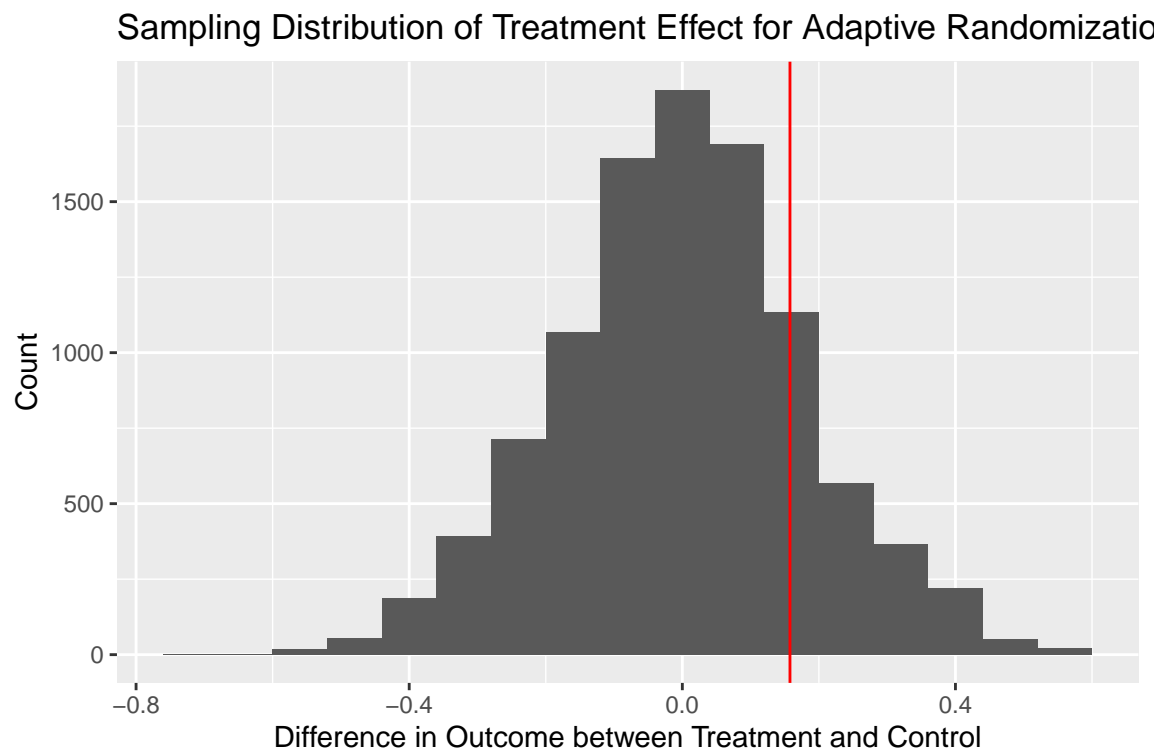
set.seed(1)
dat.q2 <- hw2
adaptive.random(dat.q2)

```

```

## [[1]]
## [1] 0.205
##
## [[2]]

```



Question 3

After running the function from question 1 on the “HW2-adaptive-trial.txt” dataset, a p-value of 0.154 is returned. This means that we do not have evidence to reject the null hypothesis at the $\alpha = 0.05$ significance level. In other words, if the null hypothesis were true and the standard of care is in fact at least as effective as the new treatment, then seeing a our sample wouldn’t be that unusual.

The sampling distributions for the 10,000 permutation simulations for both randomization schemes look identical. They are both normally distributed and centered on zero. This is to be expected given that the null probability of selecting the treatment for any observation is 50%. When these are randomized, we see what happens when there is no association. If there is no association, the difference in treatment effects would be zero.

```
dat.q1 <- hw2[ , 1:2]

set.seed(1)
simulate.perm.test(dat.q1)
```

```
## [[1]]
## [1] 0.154
##
## [[2]]
```

