Permutation test example

Chris Hoover

July 11, 2018

See Dr. Stark's similar example in python

Rat example from Dr. Stark's lecture in which rats are assigned to enriched environment (treatment=1) or not (treatment=0; i.e. control)

```
#Cortical masses of rats following 2 months of treatment
treat <- c(689, 656, 668,660, 679, 663, 664, 647, 694,633,653)
ctrl <- c(657, 623, 652, 654, 568, 646, 600, 640, 605,635, 642)

#Two sample t-test
t.test(treat, ctrl)
```

```
##
##  Welch Two Sample t-test
##
## data:  treat and ctrl
## t = 3.4999, df = 17.098, p-value = 0.002725
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  13.87408 55.94411
## sample estimates:
## mean of x mean of y
##  664.1818  629.2727
```

```
#One sample t test on differences
t.test(treat-ctrl, mu = 0, alternative = "two.sided")
```

```
##
##  One Sample t-test
##
## data:  treat - ctrl
## t = 3.1166, df = 10, p-value = 0.01094
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##   9.951667 59.866515
## sample estimates:
## mean of x
##  34.90909
```

```
#Store the value of this t-statistic as our observed test statistic, t
t.obs <- t.test(treat-ctrl, mu = 0, alternative = "two.sided")$statistic
```

Code to permute data

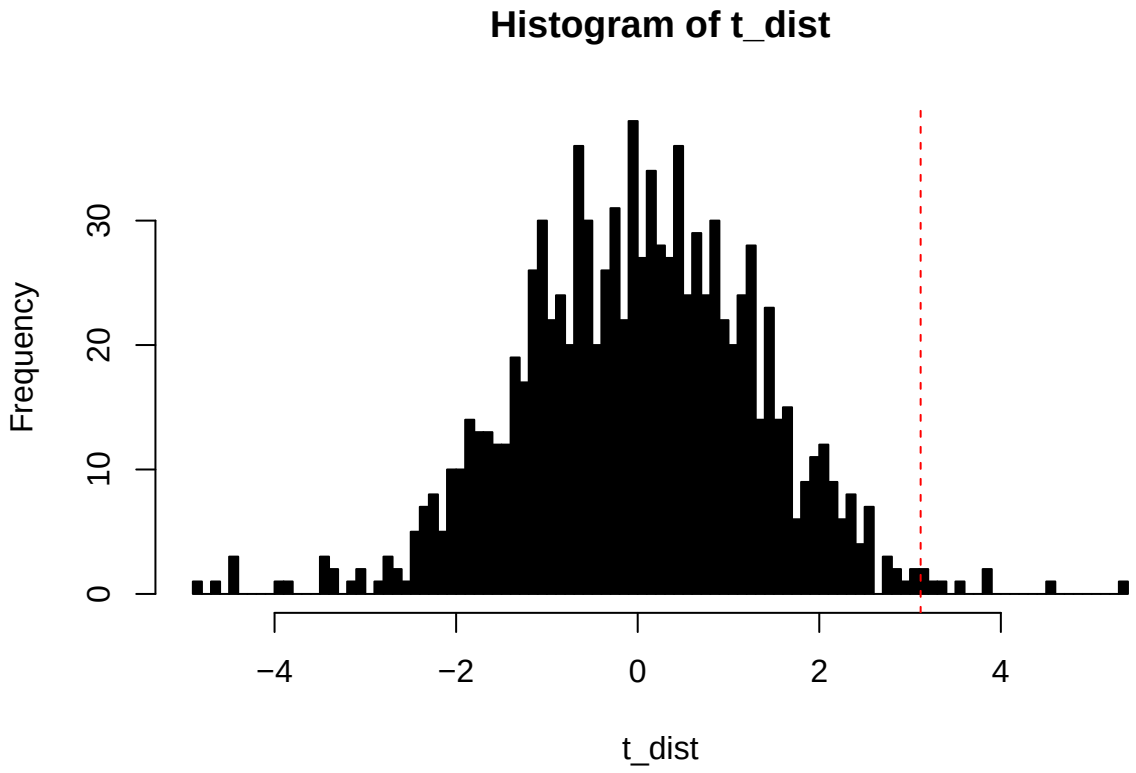
```r
#First we want all the data in the same data frame for convenience
rats <- data.frame(mass = c(treat, ctrl),
                   pair = rep(c(1:length(treat))),
                   treat = c(rep(1,length(treat)), rep(0, length(ctrl))))

rats
```

```
##    mass pair treat
## 1   689    1     1
## 2   656    2     1
## 3   668    3     1
## 4   660    4     1
## 5   679    5     1
## 6   663    6     1
## 7   664    7     1
## 8   647    8     1
## 9   694    9     1
## 10  633   10     1
## 11  653   11     1
## 12  657    1     0
## 13  623    2     0
## 14  652    3     0
## 15  654    4     0
## 16  568    5     0
## 17  646    6     0
## 18  600    7     0
## 19  640    8     0
## 20  605    9     0
## 21  635   10     0
## 22  642   11     0
```

Let's make a function that scrambles the treatment assignment and re-estimates the t-statistic. If we ignore the fact that the experiment matched rat siblings to control for genetic factors, then our permutation test is quite simple

```r
#First we get the binary vector of treatment status
treat_vec <- rats$treat

permute_bad <- function(...){
  #resample the treatment vector
    new_treats <- sample(treat_vec, size = length(treat_vec), replace = FALSE)

  #Assign this new treatment vector to the data frame
    rats$treat_perm <- new_treats

  #Estimate the t-statistic on this new data
    t_perm <- t.test(rats$mass[rats$treat_perm == 1]-rats$mass[rats$treat_perm == 0],
                     mu = 0, alternative = "two.sided")$statistic

  return(t_perm)
}

#No we do this 1000 times
```

```r
set.seed(1) #Set random seed for reproducibility
nsims = 1000

t_dist <- replicate(nsims, permute_bad())

#Plot the distribution of these t-statistics
  hist(t_dist, breaks = 100, col = 1)

#add the location of our t.obs
  abline(v = t.obs, lty = 2, col = 2)
```



Now we calculate the p-value as $\frac{\text{length}(t > t_{obs})}{n}$

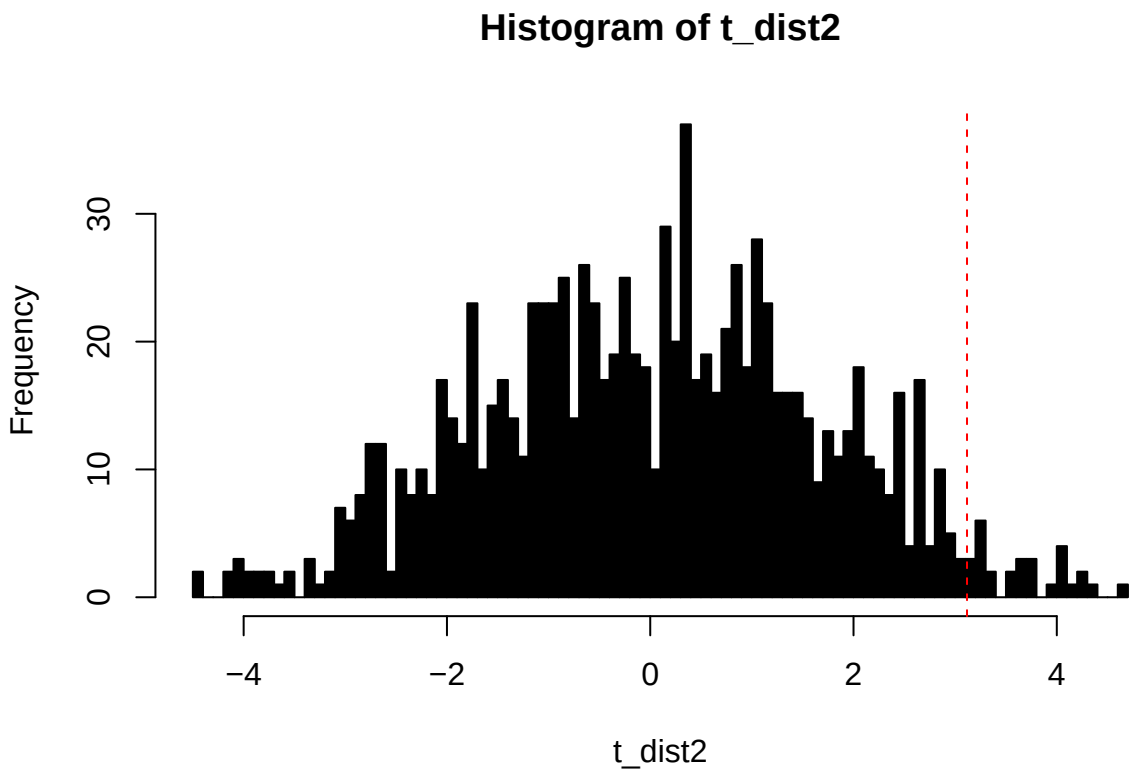
```r
p_val <- sum(t_dist > t.obs) / nsims
p_val
```

```
## [1] 0.009
```

Since the experiment was conducted on rat siblings, we want to make sure that our permutations respect this aspect of the experimentation. The code above removes the pairing of siblings, so it's not ideal. Permuting while maintaining sibling matches is a bit trickeier, but we can still do it.

```r
permute_good <- function(...){
  #resample the treatment vector for the first 11 rats
```

```
    new_treats_p1 <- rbinom(11, size = 1, prob = 0.5)
  #set their siblings to the opposite of whatever they got
    new_treats_p2 <- ifelse(new_treats_p1 == 1, 0, 1)
  #Combine for the full permuted treatment vector
    new_treats <- c(new_treats_p1, new_treats_p2)

  #Assign this new treatment vector to the data frame
    rats$treat_perm <- new_treats

  #Estimate the t-statistic on this new data
    t_perm <- t.test(rats$mass[rats$treat_perm == 1]-rats$mass[rats$treat_perm == 0],
                     mu = 0, alternative = "two.sided")$statistic

  return(t_perm)
}

#No we do this 1000 times
set.seed(1) #Set random seed for reproducibility
nsims = 1000

t_dist2 <- replicate(nsims, permute_good())

#Plot the distribution of these t-statistics
  hist(t_dist2, breaks = 100, col = 1)

#add the location of our t.obs
  abline(v = t.obs, lty = 2, col = 2)
```



Get our p-value from the matched permutation

```
p_val2 <- sum(t_dist2 > t.obs) / nsims
  p_val2
```

```
## [1] 0.028
```