# A P-Value Sampling Distribution

Austin Pesina

2021-11-30

```
set.seed(1)

# Function for P-value

pvalueSampling <- function(popA, popB, n=25, REPS=1000) {
  dist <- array(data=NA, dim=REPS)
  for (i in 1:REPS){
    A <- sample(popA, size=n)
    B <- sample(popB, size=n)
    H <- t.test(A, B, alternative="two.sided")
    dist[i] <- H$p.value
  }
  return(dist)
}

p.less <- function(A, limit=0.05/2) {
  return( length(A[A < limit])/length(A) )
}
```

### Exercise 1

**What was the estimated power of the test in the second example above? (Remember: It was a two-sided test, so use $\alpha/2$ rather than $\alpha$.)**

```
pop_a_0.5 <- rnorm(1000, mean = 10, sd = 2)
pop_b_0.5 <- rnorm(1000, mean = 11, sd = 2) # Cohen's D = 0.5

p_value_data_0.5 <- data.frame(pval = pvalueSampling(pop_a_0.5, pop_b_0.5, REPS = 4096))

p_reject_0.5 <- p.less(p_value_data_0.5$pval)
names(p_reject_0.5) <- "observed power"
p_reject_0.5
```

```
## observed power
##      0.2573242
```

The observed power of the two-sided test is 0.2573242.

### Exercise 2

**Modify the example to compare sample means when Cohen's D is 1.0 and 1.5. How does the observed power change?**

```
# For Cohen's D = 1.0

pop_a_1 <- rnorm(1000, mean = 10, sd = 2)
pop_b_1 <- rnorm(1000, mean = 12, sd = 2) # Cohen's D = (12-10)/2 = 1

p_value_data_1 <- data.frame(pval = pvalueSampling(pop_a_1, pop_b_1, REPS = 4096))

p_reject_1 <- p.less(p_value_data_1$pval)
names(p_reject_1) <- "observed power"
p_reject_1
```

```
## observed power
##      0.8771973
```

```
# For Cohen's D = 1.5

pop_a_1.5 <- rnorm(1000, mean = 9, sd = 2)
pop_b_1.5 <- rnorm(1000, mean = 12, sd = 2) # Cohen's D = (12-9)/2 = 1.5

p_value_data_1.5 <- data.frame(pval = pvalueSampling(pop_a_1.5, pop_b_1.5, REPS = 4096))

p_reject_1.5 <- p.less(p_value_data_1.5$pval)
names(p_reject_1.5) <- "observed power"
p_reject_1.5
```

```
## observed power
##      0.9985352
```

The greater our Cohen's D, the closer to 1 our observed power becomes.

### Exercise 3

**Reduce the sample sizes to $n = 9$, and compare to one of your previous cases. How does the observed power change?**

```
pvalueSampling <- function(popA, popB, n=9, REPS=1000) {
  dist <- array(data=NA, dim=REPS)
  for (i in 1:REPS){
    A <- sample(popA, size=n)
    B <- sample(popB, size=n)
    H <- t.test(A, B, alternative="two.sided")
    dist[i] <- H$p.value
  }
  return(dist)
}
```

```r
p.less <- function(A, limit=0.05/2) {
  return( length(A[A < limit])/length(A) )
}



pop_a_n_9 <- rnorm(1000, mean = 10, sd = 2)
pop_b_n_9 <- rnorm(1000, mean = 11, sd = 2) # Cohen's D = 0.5

p_value_data_n_9 <- data.frame(pval = pvalueSampling(pop_a_n_9, pop_b_n_9, REPS = 4096))

p_reject_n_9 <- p.less(p_value_data_n_9$pval)
names(p_reject_n_9) <- "observed power"
p_reject_n_9
```

```
## observed power
##     0.09228516
```

When our sample size was 25, our observed power was 0.2573242. As we decreased our sample size to 9, our observed power dropped down to 0.0922852.

## Exercise 4

Perform a similar comparison where the underlying populations are not normally distributed. You might try either a uniform distribution (`runif(n, min, max)`) or a gamma distribution (`rgamma(n,shape,rate)`). When adjusting your values for Cohen's D, recall that

$$\begin{aligned} U \sim unif(a,b) \quad & E[U] = \frac{a+b}{2} \quad Var[U] = \frac{(b-a)^2}{12} \\ X \sim gamma(\alpha,\lambda) \quad & E[X] = \frac{\alpha}{\lambda} \qquad Var[X] = \frac{\alpha}{\lambda^2} \end{aligned}$$

```r
### Functions

# t-test function

pvalueSampling <- function(popA, popB, n=25, REPS=1000) {
  dist <- array(data=NA, dim=REPS)
  for (i in 1:REPS){
    A <- sample(popA, size=n)
    B <- sample(popB, size=n)
    H <- t.test(A, B, alternative="two.sided")
    dist[i] <- H$p.value
  }
  return(dist)
}

# power function

p.less <- function(A, limit=0.05/2) {
  return( length(A[A < limit])/length(A) )
}
```

```r
# Pooled SD

se <- function(var_1, var_2){
  sqrt(((var_1)^2 + (var_2)^2)/2)
}


# Cohen's D

cohen_d <- function(mean_a, mean_b, se){
  (mean_a - mean_b)/se
}
```

```r
set.seed(1)

# Data

sh_1 <- 5.1
r_1 <- 1
sh_2 <- 4
r_2 <- 1

gam_pop_a_0.5 <- rgamma(1000, shape = sh_1, rate = r_1)
gam_pop_b_0.5 <- rgamma(1000, shape = sh_2, rate = r_2)

# Mean = a*s where a = shape and s = scale

mean_gam_pop_a_0.5 <- sh_1 * r_1
mean_gam_pop_b_0.5 <- sh_2 * r_2

# Variance = a*s^2 where a = shape and s = scale

var_gam_pop_a <- sqrt(sh_1 / (r_1)^2)
var_gam_pop_b <- sqrt(sh_2 / (r_2)^2)

# Pooled SE

pool_se_0.5 <- se(var_gam_pop_a, var_gam_pop_b)

# Cohen's D

cohen_d(mean_gam_pop_a_0.5, mean_gam_pop_b_0.5, pool_se_0.5)
```

```
## [1] 0.515688
```

```r
p_value_gam_0.5 <- data.frame(pval = pvalueSampling(gam_pop_a_0.5, gam_pop_b_0.5, REPS = 1000))

gam_p_reject_0.5 <- p.less(p_value_gam_0.5$pval)
names(gam_p_reject_0.5) <- "observed power"
gam_p_reject_0.5
```

```
## observed power
##          0.236
```

When we get our Cohen's D near a 0.5, our observed power is 0.236.

```r
# Data

sh_1 <- 5.2
r_1 <- 1
sh_2 <- 3
r_2 <- 1

gam_pop_a_1 <- rgamma(1000, shape = sh_1, rate = r_1)
gam_pop_b_1 <- rgamma(1000, shape = sh_2, rate = r_2)

# Mean = a*s where a = shape and s = scale

mean_gam_pop_a_1 <- sh_1 * r_1
mean_gam_pop_b_1 <- sh_2 * r_2

# Variance = a*s^2 where a = shape and s = scale

var_gam_pop_a_1 <- sqrt(sh_1 / (r_1)^2)
var_gam_pop_b_1 <- sqrt(sh_2 / (r_2)^2)

# Pooled SE

pool_se_1 <- se(var_gam_pop_a_1, var_gam_pop_b_1)

# Cohen's D

cohen_d(mean_gam_pop_a_1, mean_gam_pop_b_1, pool_se_1)
```

```
## [1] 1.086503
```

```r
p_value_gam_1 <- data.frame(pval = pvalueSampling(gam_pop_a_1, gam_pop_b_1, REPS = 1000))

gam_p_reject_1 <- p.less(p_value_gam_1$pval)
names(gam_p_reject_1) <- "observed power"
gam_p_reject_1
```

```
## observed power
##          0.924
```

When we get our Cohen's D near 1, our observed power became much closer to 1, at 0.939.

```r
set.seed(1)

# Data

sh_1 <- 7.7
r_1 <- 1
```

```r
sh_2 <- 4
r_2 <- 1

gam_pop_a_1.5 <- rgamma(1000, shape = sh_1, rate = r_1)
gam_pop_b_1.5 <- rgamma(1000, shape = sh_2, rate = r_2)

# Mean = a*s where a = shape and s = scale

mean_gam_pop_a_1.5 <- sh_1 * r_1
mean_gam_pop_b_1.5 <- sh_2 * r_2

# Variance = a*s^2 where a = shape and s = scale

var_gam_pop_a_1.5 <- sqrt(sh_1 / (r_1)^2)
var_gam_pop_b_1.5 <- sqrt(sh_2 / (r_2)^2)

# Pooled SE

pool_se_1.5 <- se(var_gam_pop_a_1.5, var_gam_pop_b_1.5)

# Cohen's D

cohen_d(mean_gam_pop_a_1.5, mean_gam_pop_b_1.5, pool_se_1.5)
```

```
## [1] 1.529762
```

```r
p_value_gam_1.5 <- data.frame(pval = pvalueSampling(gam_pop_a_1.5, gam_pop_b_1.5, REPS = 1000))

gam_p_reject_1.5 <- p.less(p_value_gam_1.5$pval)
names(gam_p_reject_1.5) <- "observed power"
gam_p_reject_1.5
```

```
## observed power
##          0.999
```

Our Cohen's D of 1.5 has an observed power of 0.999. We see that for the gamma distribution, the larger the Cohen's D value, the more quickly our observed power approaches 1.