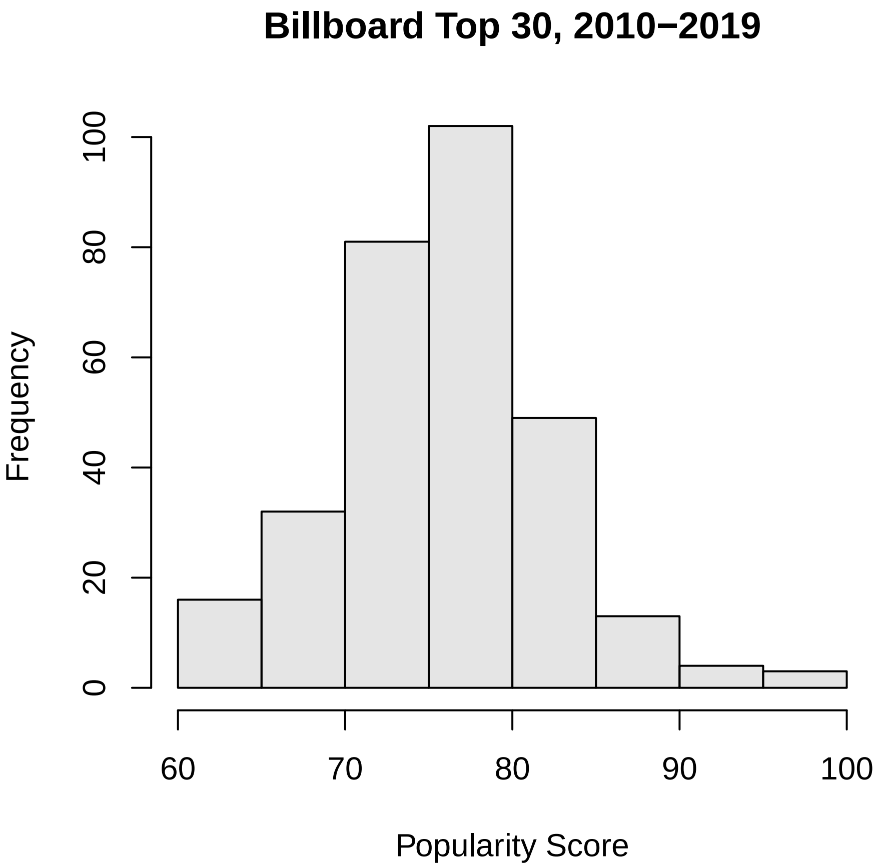
STAT 341: Tutorial 7 – More Horvitz-Thompson & Permutation Tests

Friday March 6, 2020

Part I: Horvitz-Thompson Estimation of the CDF

(a) Load the data and plot a histogram of the `popularity` score.

```r
spot <- read.csv("/Users/nstevens/Dropbox/Teaching/STAT_341/Assignments/Assignment1/spotify.csv")
hist(spot$popularity, xlab = "Popularity Score", main = "Billboard Top 30, 2010-2019",
    col = adjustcolor("grey", 0.4))
```



(b) Using the following code, take a *simple random sample without replacement* of size $n = 50$ from the population.

```r
N <- dim(spot)[1]
n <- 50
set.seed(341)
srsSampIndex <- sample(N, n)
srsSamp <- spot[srsSampIndex, ]
```

(c)(i) Calculate the Horvitz-Thompson estimate of the proportion of songs with popularity scores less than or equal to 50.

```
y_u <- (srsSamp$popularity <= 75)/N
incl.prob <- rep(n/N, N)
pi_u <- incl.prob[srsSampIndex]
propSub75HT <- sum(y_u/pi_u)
print(propSub75HT)
```

```
## [1] 0.36
```

(c)(ii) Calculate the standard error for this estimate. It will be helpful to use the `estVarHT` function defined below.

```
estVarHT <- function(y_u, pi_u, pi_uv) {
    ## y_u = an n element array containing the variate values for the sample

    ## pi_u = an n element array containing the (marginal) inclusion
    ## probabilities for the sample

    ## pi_uv = an nxn matrix containing the joint inclusion probabilities
    ## for the sample

    delta <- pi_uv - outer(pi_u, pi_u)
    estimateVar <- sum((delta/pi_uv) * outer(y_u/pi_u, y_u/pi_u))
    return(abs(estimateVar))
}
```

```
joint.incl.prob <- matrix((n * (n - 1))/(N * (N - 1)), nrow = N, ncol = N)
diag(joint.incl.prob) <- incl.prob
pi_uv = joint.incl.prob[srsSampIndex, srsSampIndex]
SEpropSub75HT <- sqrt(estVarHT(y_u, pi_u, pi_uv))
print(SEpropSub75HT)
```

```
## [1] 0.06259686
```

(c)(iii) Calculate an approximate 95% confidence interval for the proportion of songs with popularity scores less than or equal to 75 minutes.

```
propSub75HT + 2 * c(-1, 1) * SEpropSub75HT
```

```
## [1] 0.2348063 0.4851937
```

(c)(iv) Is the true proportion included in this confidence interval?

```
mean(spot$popularity <= 75)
```

```
## [1] 0.43
```

Yes!

(d) This question concerns the cumulative distribution function (CDF) of popularity scores:

$$F_{\mathcal{P}}(y) = \sum_{u \in \mathcal{P}} \frac{I_{(-\infty,\ y]}(y_u)}{N}$$

where y represents `popularity`. For the questions below, use the same sample from part (b).

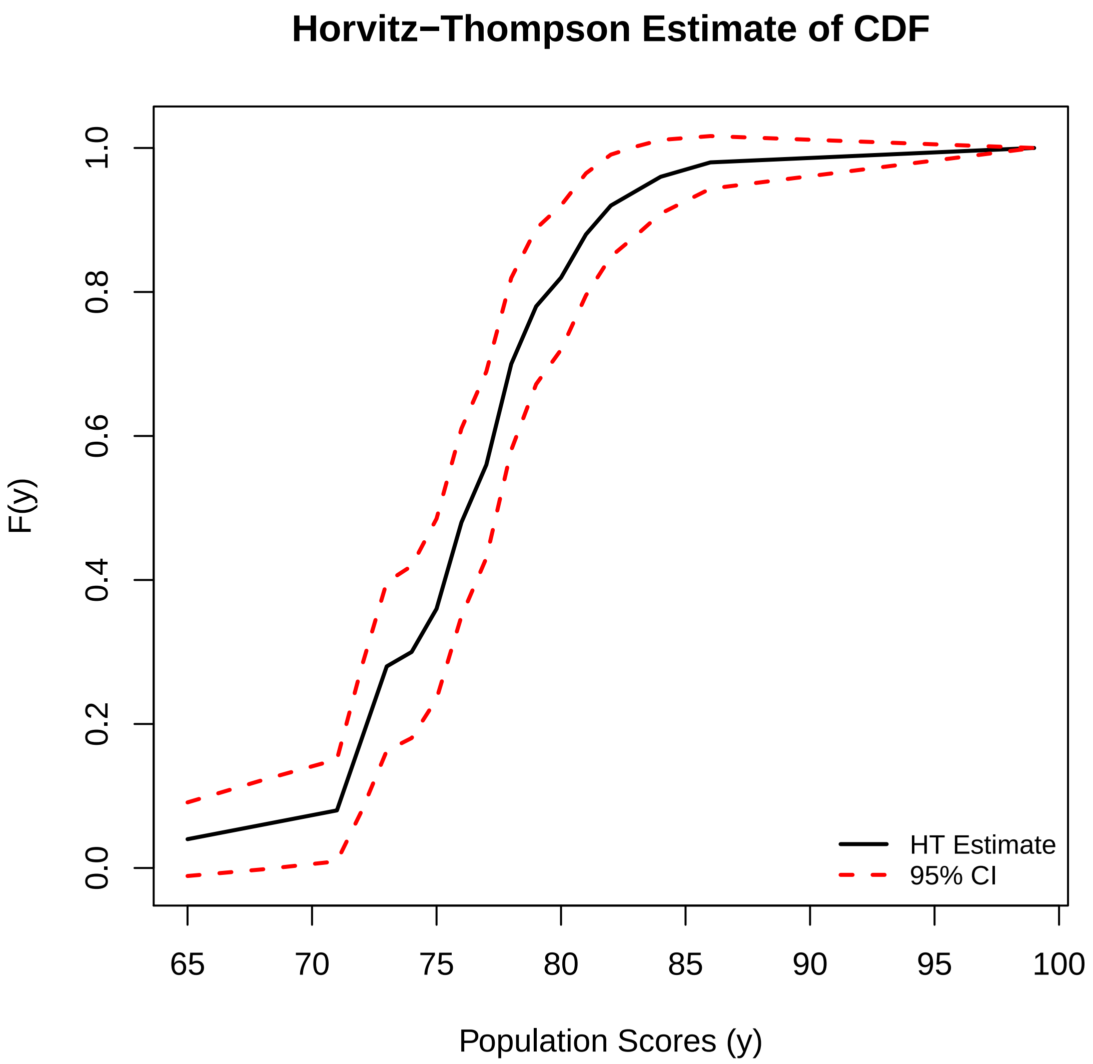
i. Calculate the Horvitz-Thompson estimate of the (CDF) of popularity scores and the standard error of the estimate. Note that this requires an estimate and standard error for every value of `popularity` observed in the sample. Print out the first 10 CDF estimates and the first 10 corresponding standard errors. It will again be helpful to use the `estVarHT` function.

```r
cdfHT <- rep(0, n)
SEcdfHT <- rep(0, n)
for (i in 1:n) {
    y_u <- (srsSamp$popularity <= srsSamp$popularity[i])/N
    cdfHT[i] <- sum(y_u/pi_u)
    SEcdfHT[i] <- sqrt(estVarHT(y_u, pi_u, pi_uv))
}
head(data.frame(est = cdfHT, se = SEcdfHT), n = 10)
```

```
##     est         se
## 1  0.28 0.05855400
## 2  0.36 0.06259686
## 3  0.18 0.05010194
## 4  0.18 0.05010194
## 5  0.18 0.05010194
## 6  0.70 0.05976143
## 7  0.18 0.05010194
## 8  0.28 0.05855400
## 9  0.36 0.06259686
## 10 0.04 0.02555506
```

ii. Construct a line-plot of the Horvitz-Thompson estimated CDF values versus the sample population scores. Add to this plot approximate 95% confidence intervals for each estimate (these are also a function of `Time`). Be sure to include a relevant title and axis labels as well as a legend that distinguishes the lines.

```r
ci_lo <- cdfHT - 2 * SEcdfHT
ci_hi <- cdfHT + 2 * SEcdfHT
plot(sort(srsSamp$popularity), cdfHT[order(srsSamp$popularity)], type = "l",
    main = "Horvitz-Thompson Estimate of CDF", ylab = "F(y)", xlab = "Population Scores (y)",
    ylim = c(min(ci_lo), max(ci_hi)), col = "black", lty = 1, lwd = 2)
lines(sort(srsSamp$popularity), ci_lo[order(srsSamp$popularity)], col = "red",
    lty = 2, lwd = 2)
lines(sort(srsSamp$popularity), ci_hi[order(srsSamp$popularity)], col = "red",
    lty = 2, lwd = 2)
legend("bottomright", c("HT Estimate", "95% CI"), col = c("black", "red"),
    lty = c(1, 2), lwd = 2, bty = "n", cex = 0.8)
```

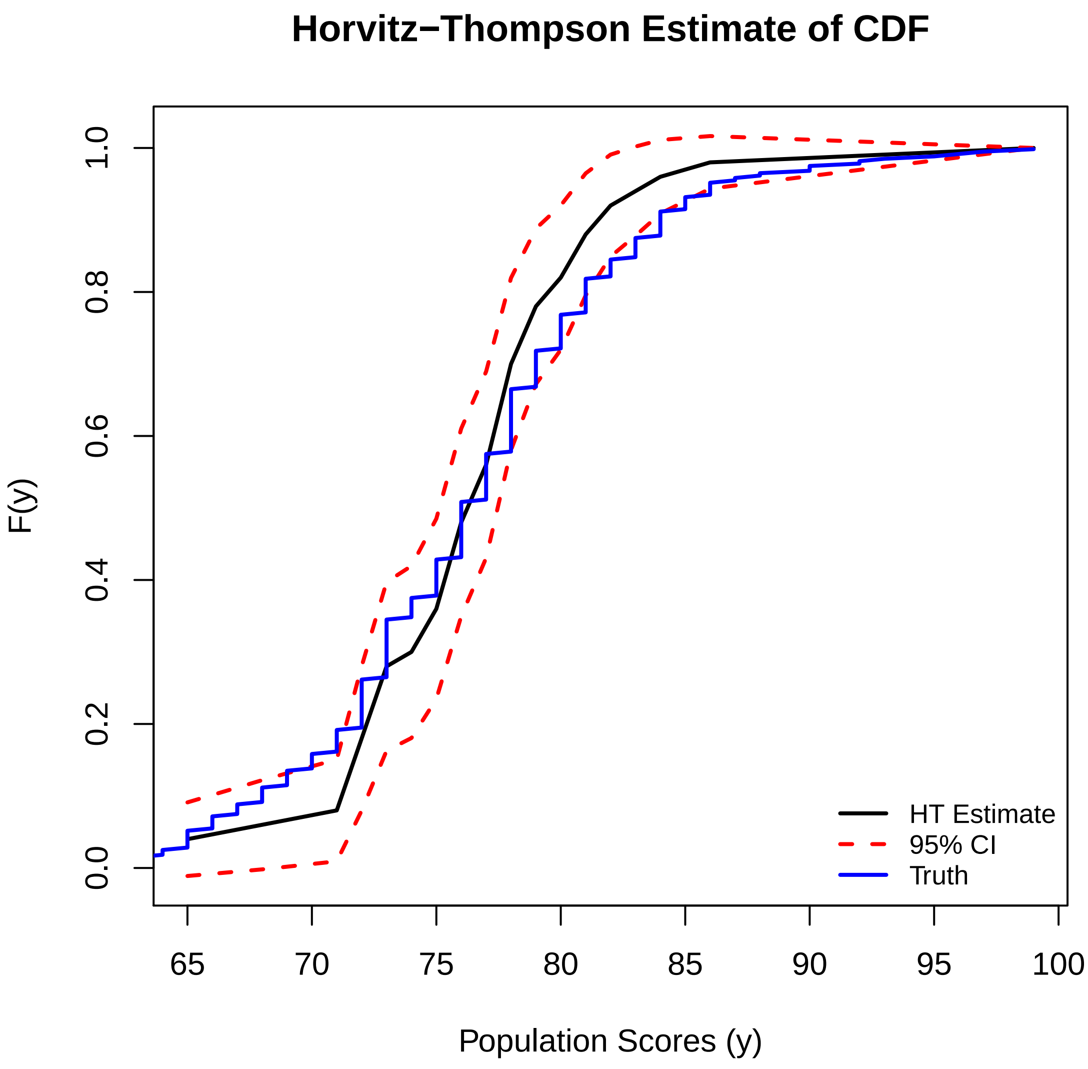


iii. Reconstruct the plot with the *true* CDF overlaid. Comment on the coverage of the confidence interval calculated from our sample.

```
ci_lo <- cdfHT - 2 * SEcdfHT
ci_hi <- cdfHT + 2 * SEcdfHT
plot(sort(srsSamp$popularity), cdfHT[order(srsSamp$popularity)], type = "l",
    main = "Horvitz-Thompson Estimate of CDF", ylab = "F(y)", xlab = "Population Scores (y)",
    ylim = c(min(ci_lo), max(ci_hi)), col = "black", lty = 1, lwd = 2)
lines(sort(srsSamp$popularity), ci_lo[order(srsSamp$popularity)], col = "red",
    lty = 2, lwd = 2)
lines(sort(srsSamp$popularity), ci_hi[order(srsSamp$popularity)], col = "red",
    lty = 2, lwd = 2)

qvals <- sort(spot$popularity)
pvals <- ppoints(length(qvals))
lines(qvals, pvals, col = "blue", lwd = 2)

legend("bottomright", c("HT Estimate", "95% CI", "Truth"), col = c("black",
    "red", "blue"), lty = c(1, 2, 1), lwd = 2, bty = "n", cex = 0.8)
```



Part II: Permutations Tests

Consider just the 2018 and 2019 songs. We may refer to them as $\mathcal{P}_{2018}$ and $\mathcal{P}_{2019}$. The null hypothesis being tested by a permutation test is the following:

H_0: $\mathcal{P}_{2018}$ and $\mathcal{P}_{2019}$ are drawn from the same population of popularity scores.

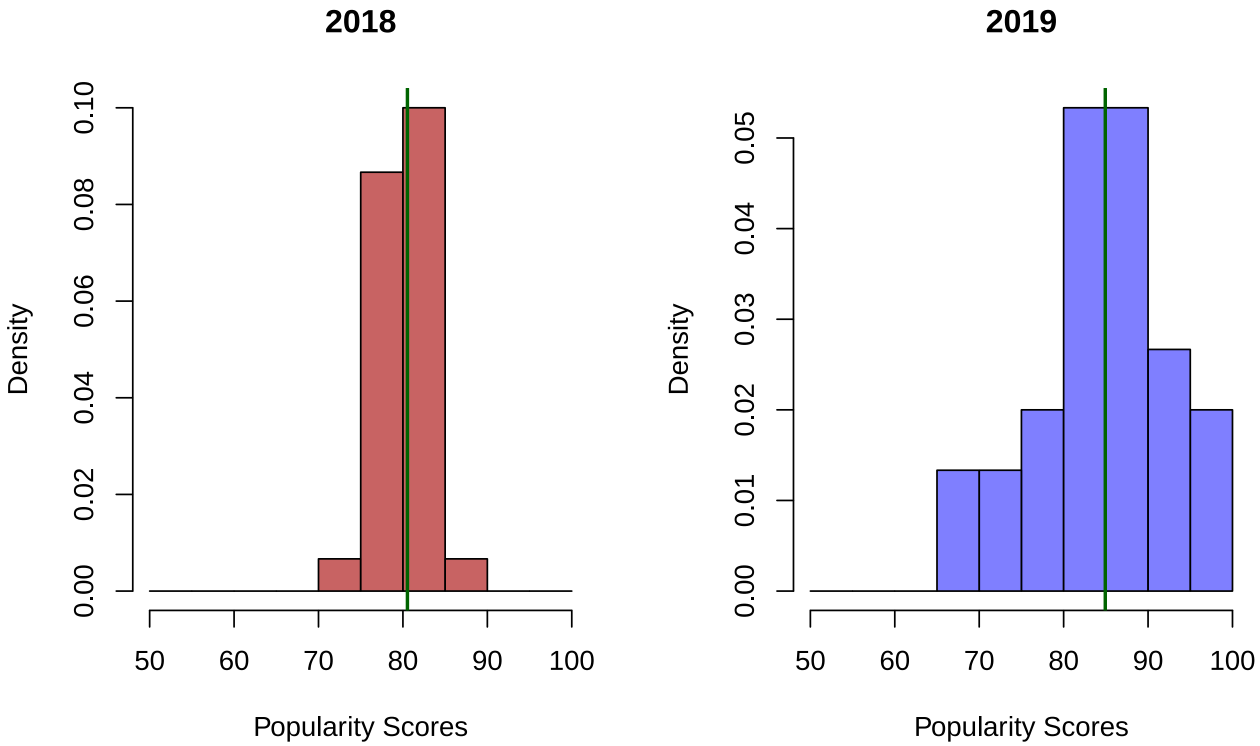
In other words, 2018 and 2019 popularity scores are indistinguishable.

(a) Construct two histograms: one of the 2018 popularity scores and the other of the 2019 popularity scores, and plot them next to each in a 1×2 grid. Indicate (with a vertical line) the average popularity score in each year.

```
pop <- list(pop1 = spot[spot$year == 2018, ], pop2 = spot[spot$year ==
    2019, ])

par(mfrow = c(1, 2))
hist(pop[[1]]$popularity, col = adjustcolor("firebrick", 0.7), freq = FALSE,
    breaks = seq(50, 100, 5), xlab = "Popularity Scores", main = "2018")
abline(v = mean(pop[[1]]$popularity), col = "darkgreen", lwd = 2)
hist(pop[[2]]$popularity, col = adjustcolor("blue", 0.5), freq = FALSE,
    breaks = seq(50, 100, 5), xlab = "Popularity Scores", main = "2019")
```

```r
abline(v = mean(pop[[2]]$popularity), col = "darkgreen", lwd = 2)
```



(b) Test the null hypothesis stated above using the discrepancy measure

$$D(\mathcal{P}_{2018}, \mathcal{P}_{2019}) = |\overline{y}_{2018} - \overline{y}_{2019}|$$

First, let's calculate the observed discrepancy:

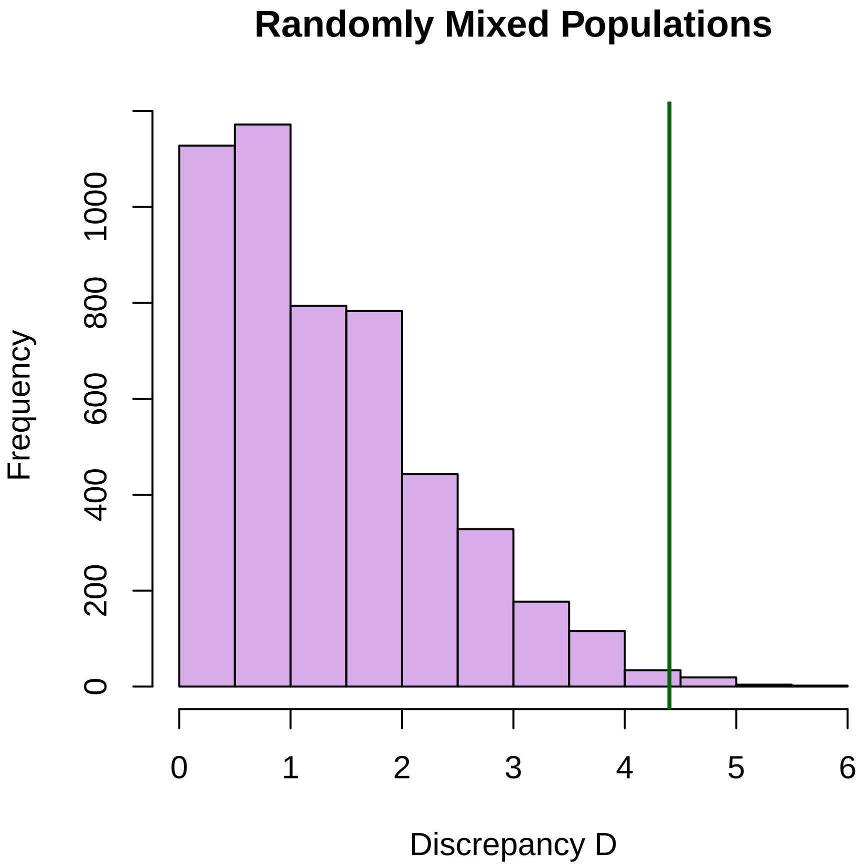
```r
D <- function(pop) {
    abs(mean(pop[[1]]$popularity) - mean(pop[[2]]$popularity))
}
d_obs <- D(pop)
print(d_obs)
```

```
## [1] 4.4
```

Now let's mix the sub-populations $M = 5,000$ times and plot a histogram of the 5,000 values of $D(\mathcal{P}^\star_{2018}, \mathcal{P}^\star_{2019})$.

```r
diffPops <- sapply(1:5000, FUN = function(...) {
    D(mixRandomly(pop))
})

hist(diffPops, breaks = 20, main = "Randomly Mixed Populations", xlab = "Discrepancy D",
    col = adjustcolor("darkorchid", 0.4))
abline(v = D(pop), col = "darkgreen", lwd = 2)
```



Now let's calculate the p-value:

```
mean(diffPops >= D(pop))
```

```
## [1] 0.0068
```

With a p-value of 0.0068 we have **strong evidence** against the null hypothesis that 2018 and 2019 popularity scores are indistinguishable, as evaluated by a comparison of averages.

(c) Test the null hypothesis stated above using the discrepancy measure

$$D(\mathcal{P}_{2018}, \mathcal{P}_{2019}) = \left| \frac{SD(\mathcal{P}_{2019})}{SD(\mathcal{P}_{2019})} - 1 \right|$$

First, let's calculate the observed discrepancy:

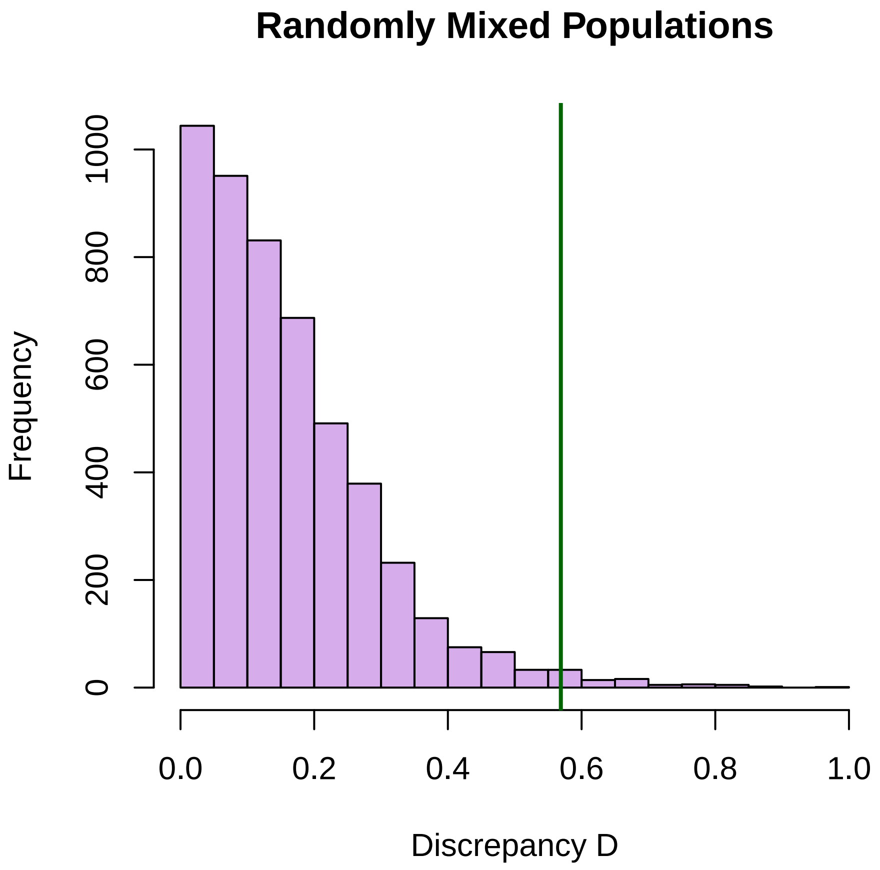
```
D <- function(pop) {
    abs(sd(pop[[1]]$popularity)/sd(pop[[2]]$popularity) - 1)
}
d_obs <- D(pop)
print(d_obs)
```

```
## [1] 0.5689747
```

Now let's mix the sub-populations $M = 5,000$ times and plot a histogram of the 5,000 values of $D(\mathcal{P}^{\star}_{2018}, \mathcal{P}^{\star}_{2019})$.

```
ratioPops <- sapply(1:5000, FUN = function(...) {
    D(mixRandomly(pop))
})
```

```
hist(ratioPops, breaks = 20, main = "Randomly Mixed Populations", xlab = "Discrepancy D",
    col = adjustcolor("darkorchid", 0.4))
abline(v = D(pop), col = "darkgreen", lwd = 2)
```



Now let's calculate the p-value:

```
mean(ratioPops >= D(pop))
```

```
## [1] 0.0134
```

With a p-value of 0.0134 we have **evidence** against the null hypothesis that 2018 and 2019 popularity scores are indistinguishable, as evaluated by a comparison of standard deviations.