PSYC 7710 Lab

Lab 8 Activity

Andrew Graves, Department of Psychology, University of Virginia

Directions:

A. Answer the following questions and save the code you used in an R script.
B. You have until the end of lab to complete.
C. Set the seed at 42 before **EACH** random draw.

Questions:

1. Simulate 2 standardized vectors of data with size 50 from a multivariate normal distribution. The two vectors should have a correlation value of .300. Name the data *my_cor_data*.

```
library(tidyverse)
library(mvtnorm)

# Modify ggplot themes
theme_set(theme_bw())
theme_update(text = element_text(family = "serif"),
axis.title.y = element_text(margin = margin(r = 20)))

# Set up correlation structure
cor_mat <- matrix(c(1, .3,
                   .3, 1),
          nrow = 2, ncol = 2, byrow = TRUE)

set.seed(42)
# Run multivariate simulation
my_cor_data <- rmvnorm(n = 50, mean = c(0, 0), sigma = cor_mat) %>%
  data.frame()
```

2. Run a permutation test on the correlation between the two vectors in *my_cor_data*. Report the *p*-value, plot the null distribution as a histogram, and overlay the histogram with a vertical line indicating the observed correlation value.

```
# P-value function
get_p_value <- function(null, obs){

  p_less <- (sum(obs <= null) + 1)/(length(null) + 1)
  p_greater <- (sum(obs >= null) + 1)/(length(null) + 1)

  if (p_less < p_greater) {

    p_value <- p_less * 2

  } else {

    p_value <- p_greater * 2
```

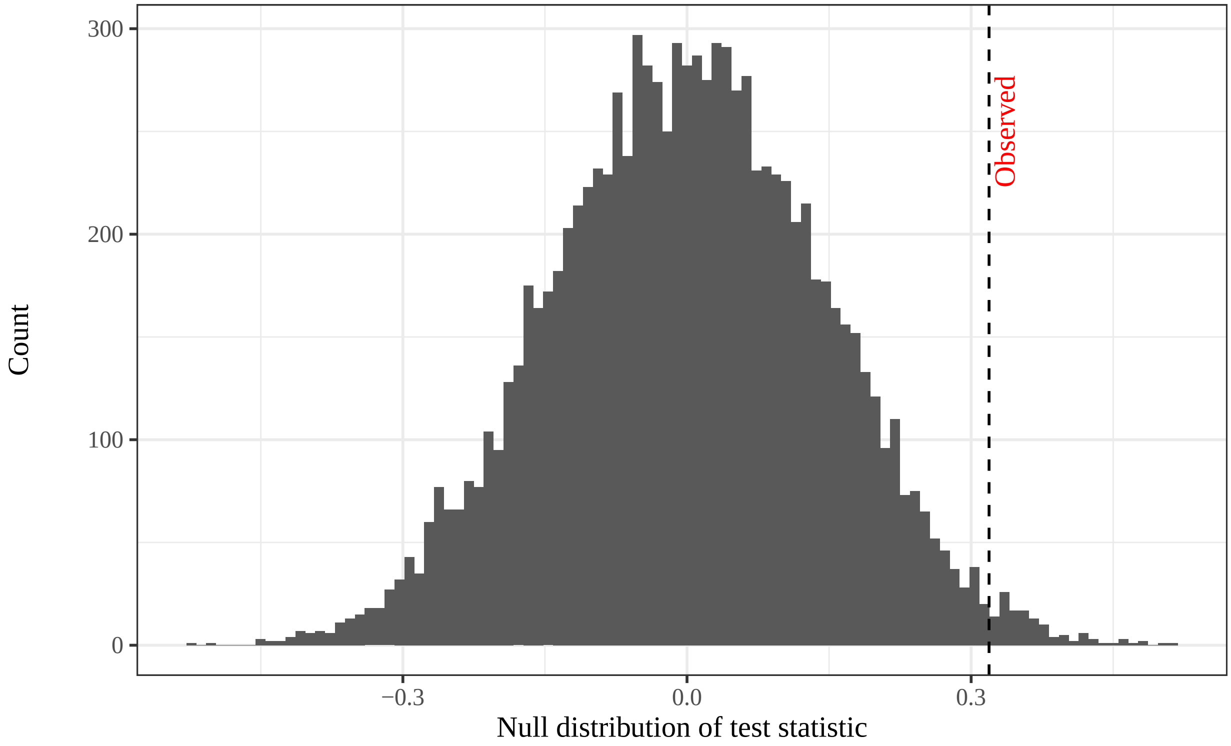
```r
  }
}

# Checking symmetry function
check_symmetry <- function(null_dist){
  set.seed(42)
  symm_test <- lawstat::symmetry.test(null_dist)

  if (symm_test$p.value > .05) {
    print(paste0("Null distribution is symmetric, p-value = ",
      symm_test$p.value))

  } else {
    print(paste0("Null distribution is asymmetric, p-value = ",
      symm_test$p.value, ". Consider changing the seed."))

 }
}

# Correlation permutation test function
cor_perm_test <- function(data, iter = 10^4, method = "pearson"){

  set.seed(42)
  obs_cor <- cor(data[, 1], data[, 2], method = method)
  null_cor <- rep(NA, iter)

  for (i in 1:iter){

    index <- sample(nrow(data), replace = FALSE)
    permuted_x <- data[index, 1]
    null_cor[i] <- cor(permuted_x, data[, 2], method = method)

  }

  p_value <- get_p_value(null_cor, obs_cor)

  return(list(null = null_cor, obs = obs_cor, p = p_value))

}

# Plot permutation test results function
plot_test_results <- function(data){

  df <- data.frame(data$null)
  names(df) <- "X1"

  df %>%
    ggplot(aes(x = X1)) +
    geom_histogram(bins = sqrt(nrow(df))) +
    geom_vline(xintercept = data$obs, linetype = 2) +
    labs(x = "Null distribution of test statistic", y = "Count") +
    annotate("text", family = "serif", x = data$obs +
               diff(range(-(abs(data$obs)), max(abs(df$X1)))) * .02,
             y = sqrt(nrow(df)) * 2.5, label = "Observed", angle = 90,
             color = "red")

}


# Run correlation permutation test
my_cor_test <- cor_perm_test(my_cor_data)
# Check symmetry of correlation null distribution
check_symmetry(my_cor_test$null)
```

```
## [1] "Null distribution is symmetric, p-value = 0.842"
```

```
# Report p-value
paste("P-value:", my_cor_test$p)
```

```
## [1] "P-value: 0.0257974202579742"
```

```
# Plot correlation permutation test results
plot_test_results(my_cor_test)
```



3. Simulate 2 vectors of data with size 50 from a univariate normal distribution. The first vector should have a mean value of 5, the second vector should have a mean value of 4, and both vectors should have a standard deviation of 2. Concatenate the two vectors and assign them to group 1 and group 2. Name the data *my_group_mean_data*.

```
my_group_mean_data <- data.frame(dv = c(rnorm(50, 5, 2),
                                 rnorm(50, 4, 2)),
                                 group = rep(1:2, each = 50))
```

4. Run a permutation test on the mean difference between the two groups in *my_group_mean_data*.

Report the *p*-value, plot the null distribution as a histogram, and overlay the histogram with a vertical line indicating the observed mean difference value.

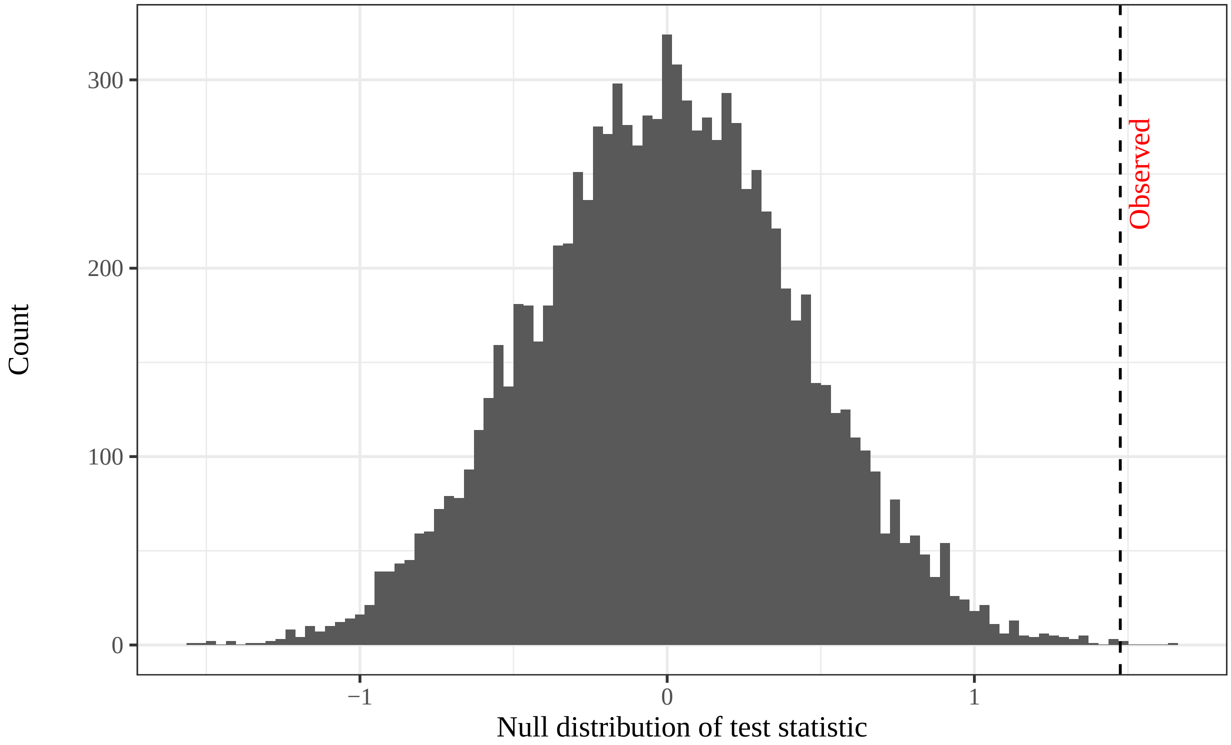
```
# Mean difference permutation test function
mean_diff_perm_test <- function(data, dv, group, iter = 10^4){

  set.seed(42)
  group <- enquo(group)
  dv <- enquo(dv)

  group_means <- data %>%
    group_by(!!group) %>%
    summarize(means = mean(!!dv)) %>%
    pull()

  obs_mean_diff <- group_means[1] - group_means[2]
  null_mean_diff <- rep(NA, iter)

  dv_data <- data %>%
    select(!!dv) %>%
    pull()

  for (i in 1:iter){

    index <- sample(nrow(data), size = nrow(data)/2, replace = FALSE)
    null_mean_diff[i] <- mean(dv_data[index]) - mean(dv_data[-index])

  }

  p_value <- get_p_value(null_mean_diff, obs_mean_diff)

  return(list(null = null_mean_diff, obs = obs_mean_diff, p = p_value))

}

# Run mean difference permutation test
my_mean_diff_test <- mean_diff_perm_test(my_group_mean_data, dv = dv, group = group)
# Check symmetry of mean difference null distribution
check_symmetry(my_mean_diff_test$null)
```

```
## [1] "Null distribution is symmetric, p-value = 0.628"
```

```
# Report p-value
paste("P-value:", my_mean_diff_test$p)
```

```
## [1] "P-value: 0.00059994000599994"
```

```
# Plot mean difference permutation test results
plot_test_results(my_mean_diff_test)
```



5. Use the *lm* base R function to estimate the test statistics from questions 2 and 4. Which *p*-value is lower for both statistics, the *lm* function or the permutation tests? How close are the *p*-values across both methods for each statistic?

```
(cor_lm_p <- summary(lm(X2 ~ X1, my_cor_data))$coefficients[2,4])
```

```
## [1] 0.02398319
```

```
(mean_diff_lm_p <- summary(lm(dv ~ group, my_group_mean_data))$coefficients[2,4])
```

```
## [1] 0.0004894896
```

```
if ((cor_lm_p < my_cor_test$p) & (mean_diff_lm_p < my_mean_diff_test$p)){
  print("The lm function produces lower p-values")
} else if ((cor_lm_p > my_cor_test$p) & (mean_diff_lm_p > my_mean_diff_test$p)){
  print("The permutation test produces lower p-values")
} else {
  print("Neither method produces lower p-values for both the correlation
        and mean difference analysis")
}
```

```
## [1] "The lm function produces lower p-values"
```

```
paste("The difference between the two correlation p-values is",
      diff(range(cor_lm_p, my_cor_test$p)))
```

```
## [1] "The difference between the two correlation p-values is 0.0018142286618095"
```

```
paste("The difference between the two mean difference p-values is",
      diff(range(mean_diff_lm_p, my_mean_diff_test$p)))
```

```
## [1] "The difference between the two mean difference p-values is 0.000110450388608219"
```

```
# These are simulated data with known parameters. The assumptions of the
# linear model hold in these two cases, meaning that we can rely on the
# asymptotic properties of the linear model to estimate frequentist
# probabilities. With real data that is less cleanly distributed containing
# hetereogenous variances across populations, the permutation method can
# provide better estimates of the true probability and uncertainty
# surrounding the inference.
```