Assignment 04

Programming and Estimation - Answer Key

Randomization Test

In this set of exercises, you are going to write a function to carry out a randomization test as described in Statistics Without the Agonizing Pain.

1. Write a function that carries out a randomization test for the mean difference between two groups. This function should take inputs of (1) a vector containing the grouping variable (i.e., conditions), (2) a vector containing the outcome variable, and (3) the number of permutations/randomizations to perform. The output from this function should be a vector of randomized/permuted mean differences. Include the syntax for your function in your word-processed document. Be sure that your function includes comments.

Enter the data presented in Statistics Without the Agonizing Pain into a data frame so that it can be used in your function.

```
# Enter in the data
attract = data.frame(
  mosquitos = c(27, 20, 21, 26, 27, 31, 24, 21, 20, 19, 23, 24, 28, 19, 24, 29, 18, 20,
                17, 31, 20, 25, 28, 21, 27, 21, 22, 15, 12, 21, 16, 19, 15, 22, 24, 19,
                23, 13, 22, 20, 24, 18, 20),
  drink = c(rep("beer", 25), rep("water", 18))
)

head(attract)
```

```
##   mosquitos drink
## 1        27  beer
## 2        20  beer
## 3        21  beer
## 4        26  beer
## 5        27  beer
## 6        31  beer
```

```
# Mean difference (to check data entry)
tapply(attract$mosquitos, attract$drink, mean)
```

```
##     beer    water
## 23.60000 19.22222
```

```
diff(tapply(attract$mosquitos, attract$drink, mean))
```

```
##     water
## -4.377778
```

```r
# Write a function
randtest = function(outcome, group, k = 1000){
  n = length(outcome) # Sample size
  n1 = table(group)[[1]] # Size of group 1
  md = rep(NA, k) # Create vector with k NA values
  for(i in 1:k){
    rand_outcome = sample(outcome, replace = FALSE) # Randomize the outcome
    md[i] = mean(rand_outcome[1:n1]) - mean(rand_outcome[(n1+1):n]) # Compute mean difference
  }
  return(md)
}
```

2. Use your function to carry out a randomization test (using 1000 permutations) on the data. Assign the output into an object. Show the results of running the `head()` function on this object.

```r
myMeanDiffs = randtest(outcome = attract$mosquitos, group = attract$drink, k = 1000)
head(myMeanDiffs)
```

```
## [1] -0.68666667  0.46000000  2.56222222 -0.59111111 -0.01777778  0.36444444
```

3. Use the `t.test()` or the `lm()` function to carry out a parametric test to examine the mean difference in the number of mosquitos between the two conditions. Present all pertinent results (i.e., t-value, df, p-value) from this analysis.

```r
# To replicate the video's analysis set var.equal=FALSE in the t.test() function
t.test(mosquitos ~ drink, data = attract, var.equal = FALSE)
```

```
##
##  Welch Two Sample t-test
##
## data:  mosquitos by drink
## t = 3.6582, df = 39.113, p-value = 0.0007474
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  1.957472 6.798084
## sample estimates:
##  mean in group beer mean in group water
##            23.60000            19.22222
```

4. Compute the p-value for the observed mean difference using the object containing your function's assigned output. Show the syntax you used to compute this.

```
# Count how many have an absolute value greater than or equal to the observed difference of 4.377778
sum(abs(myMeanDiffs) >= 4.377778)
```

```
## [1] 0
```

```
# p-value is that count divided by the number of permutations
sum(abs(myMeanDiffs) >= 4.377778) / 1000
```

```
## [1] 0
```

5. How does that the p-value computed in the parametric analysis compare to that from the randomization analysis?

It is within rounding. If we wanted to be closer, we could carry out more permutations.

```
myMeanDiffs2 = randtest(outcome = attract$mosquitos, group = attract$drink, k = 1000000)
sum(abs(myMeanDiffs2) >= 4.377778) / 1000000
```

```
## [1] 0.000841
```

6. Compute the estimated standard error for the randomization analysis using the object containing your function's assigned output. Show the syntax you used to compute this.

```
sd(myMeanDiffs)
```

```
## [1] 1.383084
```

```
# SE for more permutations
sd(myMeanDiffs2)
```

```
## [1] 1.380867
```

7. How does that compare to the estimated standard error computed in the parametric analysis?

In general,

$$t = \frac{\text{Estimate}}{SE_{\text{Estimate}}}$$

For the two-sample test, the estimate is the observed mean difference (4.377778). We know the value of t, so we can solve for SE.

```
se = 4.377778 / 3.6582
se
```

```
## [1] 1.196703
```

The SE for both sets of analyses are similar.

8. Plot the randomized/permuted mean differences collected in your function. Draw the appropriate t-distribution from the parameteric analysis as a line on this plot.

```
library(sm)
```

```
## Package 'sm', version 2.2-5.4: type help(sm) for summary information
```

```
sm.density(myMeanDiffs, xlab = "Mean Differences")
x = seq(from = -5, to = 5, by = 0.01)
y = dt(x = x, df = 39.113)
lines(x, y, type = "l", lty = "dashed")
```

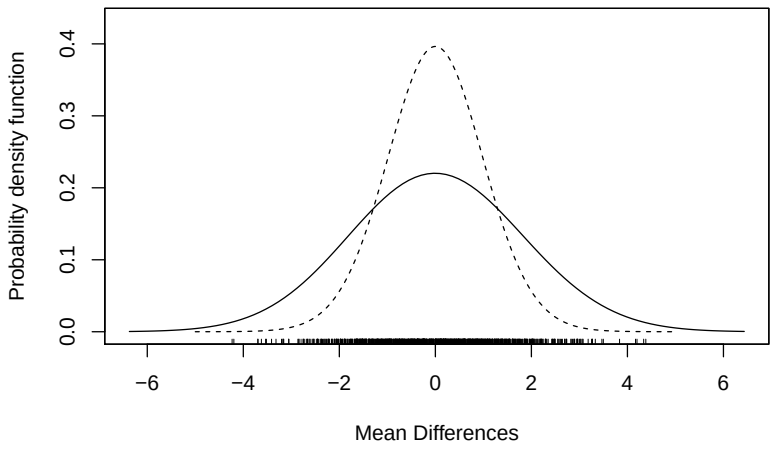

Figure 1. Density plots of the randomized mean differences (solid) and t-distribution with 39.113 *df* (dashed).

Technically, the t-distribution is a "standardized" distribution. So, to get them on the same scale, we should standardize the distribution of mean differences.

```
z = myMeanDiffs / sd(myMeanDiffs)
sm.density(z, xlab = "Mean Differences", col = "blue")
x = seq(from = -5, to = 5, by = 0.01)
y = dt(x = x, df = 39.113)
lines(x, y, type = "l", lty = "dashed", col = "red")
abline(v = 3.6582, col = "red", lty = "dashed")
abline(v = (4.377778 / sd(myMeanDiffs)), col = "blue")
```

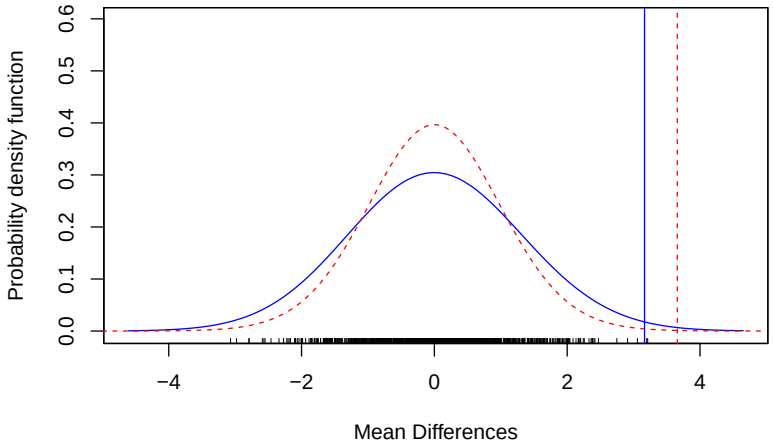

Figure 1. Density plots of the standardized randomized mean differences (solid blue) and t-distribution with 39.113 *df* (dashed red). The vertical dashed red line displays the observed t-value of 3.6582. The vertical solid blue line displays the observed standardized mean difference of 3.07.

9. How do these two distributions compare? Make reference to the shape, center, and variation.

The two shapes have the same center (0) and the variation is similar (based on the SEs computed earlier). The t-distribution with 39.113 *df* is more "scrunched up" near the center (more leptokurtic). This would suggest that perhaps the t-distribution is not a good estimate for the distribution of mean differences, and it would be better to use the randomization procedure.

Regression Estimation

Least squares estimation optimizes the criterion of the sum of squared error in computing the estimates for a given set of regression parameters. In this set of exercises, you are going to examine the estimates we obtain for the set of parameters for the model $\text{Wage}_i = \beta_0 + \beta_1(\text{Age}_i) + \epsilon_i$ when we change the criterion for model fit (mis-fit). In this set of exercises, you will be computing the sum of the absolute errors, $\left|Y_i - \hat{Y}_i\right|$, as a measure of the model fit (mis-fit). Use the following data set to help you answer the following questions,

```
myData = data.frame(
    wage = c(12, 8, 16.26, 13.65, 8.5),
    age = c(32, 33, 32, 33, 26)
    )
myData
```

```
##    wage age
## 1 12.00  32
## 2  8.00  33
## 3 16.26  32
## 4 13.65  33
## 5  8.50  26
```

10. Write a function that computes the sum of the absolute errors given the estimated the regression coefficients $\hat{\beta}_0$ and $\hat{\beta}_1$. Include the syntax for your function in your word-processed document. Be sure that your function includes comments.

```
absErr = function(b0, b1, x, y){
  yhat = b0 + b1*x
  err = abs(y - yhat)
  sum(err)
}
```

11. Provide the output of your function (i.e., the sum of the absolute errors) for the parameter estimates of $\hat{\beta}_0 = 3$ and $\hat{\beta}_1 = 2$.

```
absErr(b0 = 3, b1 = 2, x = myData$age, y = myData$wage)
```

```
## [1] 268.59
```

12. Carry out a grid search to estimate the coefficients (to the nearest hundredth) when we minimize the sum of the absolute errors. Report the parameter estimates.

```
## Generate candidate values for b0
candidates = expand.grid(
    b0 = seq(from = -7, to = -6, by = 0.001),
    b1 = seq(from = 0.5, to = 0.6, by = 0.001)
    )


## Generate the sum of the absolute error values and store the results
library(dplyr)
gridSearch = candidates %>%
  rowwise() %>%
  mutate(Error = absErr(b0, b1, x = myData$age, y = myData$wage))

## Get the coefficients with the smallest sum of the absolute errors
gridSearch %>% arrange(Error) %>% slice(1)
```

```
## Source: local data frame [1 x 3]
##
##       b0    b1 Error
##    (dbl) (dbl) (dbl)
## 1 -6.684 0.584  9.91
```

```
## Compare to built-in function
quantreg::rq(wage ~ age, 0.5, data = myData)
```

```
## Warning in rq.fit.br(x, y, tau = tau, ...): Solution may be nonunique

## Call:
## quantreg::rq(formula = wage ~ age, tau = 0.5, data = myData)
##
## Coefficients:
## (Intercept)         age
##  -6.6666667   0.5833333
##
## Degrees of freedom: 5 total; 3 residual
```

13. Compare the estimated effect of age from this estimation to that from the OLS estimate. How different are the interpretations for the effect of age on wage when we use a different criterion for measuring the model fit (or misfit)?

```
summary(lm(wage ~ age, data = myData))
```

```
##
## Call:
## lm(formula = wage ~ age, data = myData)
##
## Residuals:
##        1        2        3        4        5
## -0.08149 -4.58086  4.17851  1.06914 -0.58529
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -3.8983    19.3658  -0.201    0.853
## age           0.4994     0.6185   0.807    0.479
##
## Residual standard error: 3.649 on 3 degrees of freedom
## Multiple R-squared:  0.1785, Adjusted R-squared:  -0.09532
## F-statistic: 0.6519 on 1 and 3 DF,  p-value: 0.4785
```

The coefficients are not very similar. The statistic that minimizes the sum of the absolute error is the median. When the mean and median values are similar for the outcome, the intercepts will be reasonablly close in the two models. If conditional normality is met, then the slopes will also be close to one another. In this case, the assumptions are probably not well satisifed, so the two models give very different results.