Exercise 1

Twan J. Kerkhof - 1287753 - t.j.kerkhof@tilburguniversity.edu
* Apologies for the terrible layout but I have been struggling with markdown for several hours and I am not able to fix it in a reasonable amount of time *

Part 1:
1:
population standard error of the mean difference, you can estimate this with SE of student t and SE of welch: First calculate SE_{Welch} using the given formula which yields: $SE_{Welch} = \sqrt{\frac{1}{10} + \frac{1}{200}} = \sqrt{\frac{21}{200}}$. For the SE_p we first need the pooled standard deviation which we calculate using the given formula which yields: $\sqrt{\frac{208}{208}} = 1$. We use this together with the given formula for SE_p to find:

$$SE_p = 1 * \sqrt{\frac{1}{10} + \frac{1}{200}} = \sqrt{\frac{21}{200}}$$

. So $SE_{Welch} = SE_p$ with 0.324.

b. See below:
Input:
-S: integer definin the number of independent data sets to generate in the MC simulation
-n1 and n2: sample size for dataset 1 and dataset 2 respecitvely.
-μ_1 and μ_2: means for dataset 1 and dataset 2 respecitvely.
-σ_1^2 and σ_2^2: variances for dataset 1 and dataset 2 respectively.
Output:
-S estimates of the population standard error of the mean difference for the entered values using SE for Welch t-test and Student t-test.

MonteCarlo_SE(S, n1, n2, μ_1, μ_2, σ_1^2, σ_2^2)
1. Initialize variables SE_student and SE_welch as vectors of length S
2. For i from 1 to S do:
A. Generate and store two seperate datasets with their respecitve values from N(μ, σ^2).
B. Obtain pooled standard deviation and save as a variable.
C. Obtain and add results of the Standard Errors for welch and students t-tests to the vectors SE_student and SE_welch.
3. Obtain bias, variance and MSE from the vectors SE_student and SE_welch and store in variables for output.
4. Divide the two obtained MSE's to optain the Relative Efficiency and store in variable for output.
5. Return two the two biasses, two variances, two MSE's and the Relative Efficiency.

Note: the A. B. and C. in the psuedo code above should be indented but due to some problems with markdown I was not able to do this.

c. See code and results below:

```
set.seed(1000)
n1 <- c(10, 100, 200)
n2 <- 200
var2 <- c(1, 2 , 10)
var1 <- 1
mean1 <- 0
mean2 <- 1
conditions <- expand.grid(n1, var2)

MonteCarlo_SE <- function(n1, var2, S) {
```

```r
  SE_student <- rep(NA, S)
  SE_welch <- rep(NA, S)
  for (i in 1:S) {
    data_1 <- rnorm(n1, mean1, sqrt(var1))
    data_2 <- rnorm(n2, mean2, sqrt(var2))
    pooled_sd <- sqrt(((n1-1)*var(data_1)+(n2-1)*var(data_2))/(n1+n2-2))
    SE_student[i] <- pooled_sd*sqrt(((1/n1)+(1/n2)))
    SE_welch[i] <- sqrt((var(data_1)/n1)+(var(data_2)/n2))
  }
  bias_Student <- sqrt((var1/n1)+(var2/n2)) - mean(SE_student)
  bias_Welch <- sqrt((var1/n1)+(var2/n2)) - mean(SE_welch)
  Out_variance_student <- var(SE_student)
  Out_variance_welch <- var(SE_welch)
  Out_bias_student <- bias_Student
  Out_bias_welch <- bias_Welch
  Out_MSE_student <- (bias_Student)^2 + Out_variance_student
  Out_MSE_welch <- (bias_Welch)^2 + Out_variance_welch
  RE <- ((Out_MSE_student)/(Out_MSE_welch))
  return(c(Out_variance_student, Out_variance_welch, Out_bias_student,
           Out_bias_welch, Out_MSE_student, Out_MSE_welch, RE))
}

results <- matrix(NA, 7, 9)
for (i in 1:nrow(conditions)) {
  results[, i] <- MonteCarlo_SE(n1 = conditions[i, 1],
                                var2 = conditions[i,2], S = 10000)
}

out_results <- cbind(conditions, t(round(results, digits = 3)))
colnames(out_results) <- c("n1", "var2", "Variance_S", "variance_W",
                           "Bias_S", "Bias_W", "MSE_S", "MSE_W", "RE")
out_results
```

```
##    n1 var2 Variance_S variance_W Bias_S Bias_W MSE_S MSE_W      RE
## 1  10    1      0.000      0.005  0.000  0.006 0.000 0.005   0.049
## 2 100    1      0.000      0.000  0.000  0.000 0.000 0.000   0.659
## 3 200    1      0.000      0.000  0.000  0.000 0.000 0.000   1.000
## 4  10    2      0.000      0.005 -0.121  0.009 0.015 0.005   3.218
## 5 100    2      0.000      0.000 -0.017  0.000 0.000 0.000   8.639
## 6 200    2      0.000      0.000  0.000  0.000 0.000 0.000   1.000
## 7  10   10      0.002      0.003 -0.617  0.004 0.383 0.004 108.810
## 8 100   10      0.000      0.000 -0.079  0.000 0.006 0.000  57.766
## 9 200   10      0.000      0.000  0.000  0.000 0.000 0.000   1.000
```

d. The assumption violated in the majority of the conditions is the assumption that both variances are equal.

3
- With small sample sizes and unequal variances the student t-test is more biased than the welch t-test.
With small sample sizes and an equal variances Welch t-test will be more biased.
With large sample sizes compared to the variance the difference in bias between Welch and Student-tests.
- The results show that when the differences in variances between dataset 1 and dataset 2 are larger, with dataset 2 having a higher variance, the variance increases in both tests.
- The relative efficiency shows that Welch t-test is more efficient when the sample size is low and the variances are unequal. Student t-test is more efficient when samples siezes are low and variances are equal.

Question 2:
2.1
Null hypothesis: The means of the two t-tests are the same Alternative Hypothesis: The means of the two t-tests are not the same

2.2
MonteCarlo_ST(S, n1, n2, μ_1, μ_2, σ_1^2, σ_2^2)
1. intialize variables p_values_student, p_values_welch, TP_student and TP_welchs
2. For i from 1 to S do:
A. Generate and store three datasets, 2 with the their respective values N(μ, σ^2) as data1 and data2 and a data set with the all values equal to the second dataset but with μ_1 and store as data3
B. Obtain the amount P values below the provided significance level for the welch and student t-tests on data1 and data2 and store them in their respective variables p_values_welch and p_values_student
C. Obtain the amount of p values below the provided significance level for the welch and student t-tests on data3 and store them in their respective variables TP_student and TP_welch
3. Devide all devide the values stored in the variables p_values_student, p_values_welch, TP_student and TP_welch all by S
4. Return p_values_student, p_values_welch, TP_student and TP_welchs

Note: the A. B. and C. in the psuedo code above should be indented but due to some problems with markdown I was not able to do this.

2.3
```r
set.seed(10000)
S <- 10000
n1 <- c(10, 100, 200)
n2 <- 200
var1 <- 1
var2 <- c(1, 2, 10)
mean1 <- 0
mean2 <- 1
conditions <- expand.grid(n1, var2)

MonteCarlo_ST <- function(n1, var2) {
  p_values_welch <- rep(0, S)
  p_values_student <- rep(0, S)
  TP_student <- rep(0, S)
  TP_welch <- rep(0, S)
  for (i in 1:S) {
    data_1 <- rnorm(n1, mean1, sqrt(var1))
    data_2 <- rnorm(n2, mean2, sqrt(var2))
    data_3 <- rnorm(n2, mean1, sqrt(var2))
    res_welch <- t.test(data_1, data_3, var.equal = FALSE)
    res_student <- t.test(data_1, data_3, var.equal = TRUE)
    pwr_calc_welch <- t.test(data_1, data_2, var.equal = FALSE)
    pwr_calc_student <- t.test(data_1, data_2, var.equal = TRUE)
    if (res_welch$p.value <= 0.05) {
      p_values_welch[i] <- 1
    }
    if (res_student$p.value <= 0.05) {
      p_values_student[i] <- 1
    }
    if (pwr_calc_welch$p.value <= 0.05) {
      TP_welch[i] <- 1
    }
```

```r
    if (pwr_calc_student$p.value <= 0.05) {
      TP_student[i] <- 1
    }
  }
  return(c(sum(p_values_student)/S, sum(p_values_welch)/S,
           sum(TP_student)/S, sum(TP_welch)/S))
}

results <- matrix(NA, 4, 9)
for (i in 1:nrow(conditions)) {
  results[, i] <- MonteCarlo_ST(n1 = conditions[i, 1],
                                var2 = conditions[i,2])
}
out_results <- cbind(conditions, t(round(results, digits = 4)))
colnames(out_results) <- c("n1", "var2", "Level Student", "Level Welch",
                           "Power Student", "Power Welch")
out_results
```

```
##    n1 var2 Level Student Level Welch Power Student Power Welch
## 1  10    1        0.0462      0.0489        0.8673      0.7960
## 2 100    1        0.0502      0.0502        1.0000      1.0000
## 3 200    1        0.0543      0.0543        1.0000      1.0000
## 4  10    2        0.0080      0.0505        0.6404      0.7865
## 5 100    2        0.0296      0.0527        1.0000      1.0000
## 6 200    2        0.0535      0.0534        1.0000      1.0000
## 7  10   10        0.0000      0.0513        0.0071      0.6923
## 8 100   10        0.0107      0.0516        0.9261      0.9827
## 9 200   10        0.0470      0.0468        0.9899      0.9898
```

2.4
- The method more consistently performing close to the provided significance level of 0.05 is the Welch t-test. It gets closes with n1 = 100 and n2 = 200 and equal variances for both datasets with variance being 1. - There is not a clear winner considering power. it is very clear however that the power is greater with with the larger sample sizes (100 and 200) with power = 1 for both methods for all combinations values of variance except 10.

Part 1 and 2:
I support this practice as my results and what I have found in documentation and research is that the welch t-test pefroms better in a wider range of cases compared to the student t-test. The Welch t-test only shows bad performance with small sample sizes. It will yield a better result in most of the cases. But comparing these two methods and others when performing comparissons of means is always important and worth thinking about.