

Homework 6

(Parts 10-11; 40 pts)

Name: Ivan Wang

1. Suppose we plan to carry out the one-way ANOVA procedure on a data set with a continuous response variable and a single categorical predictor variable having $r = 6$ levels. Before fitting the model, the lead investigator wishes to use Hartley's F-max test, with $\alpha = 0.1$, to address the following hypotheses. (10 pts)

$$H_0: \sigma_1^2 = \sigma_2^2 = \dots = \sigma_6^2$$

H_a : not all equal

Assume that the response variables in each group are normally distributed with mean 0 and standard deviation 1, and that we have a balanced design with $n = 8$ observations per group.

- a) Write a simulation that generates n observations from $N(0,1)$ in each of the r groups, keeping track of which group each observation belongs to. You should have a total of $nr = 48$ observations. Obtain group-specific sample variances S_i^2 , and calculate the observed value of F_{\max} , where this variable is defined as:

$$F_{\max} = \frac{\max_i(S_i^2)}{\min_i(S_i^2)}$$

Print your value of F_{\max} . (2)

```
> ?rnorm
> n <- 8
> r <- 6
>
> observations <- rnorm(n * r, mean = 0, sd = 1)
> labels <- rep(1:r, each = n)
>
> data <- data.frame(value = observations, group = factor(labels))
>
> group_variances <- tapply(data$value, data$group, var)
>
> F_max <- max(group_variances) / min(group_variances)
> F_max
[1] 2.516873
```

- b) Perform the simulation from the previous part $B = 50,000$ times, storing 50,000 values of F_{\max} . Use this collection of values to estimate the 90th quantile of the distribution of F_{\max} . (2)

```

> n <- 8
> r <- 6
> B <- 50000
>
> F_values <- numeric(B)
>
> for (x in 1:B) {
+   data <- rnorm(n * r)
+   labels <- numeric(n * r)
+   for (i in 1:r) {
+     labels[((i-1)*n + 1):(i*n)] <- i
+   }
+
+   group_variances <- tapply(data, labels, var)
+   F_values[x] <- max(group_variances) / min(group_variances)
+ }
>
> quantile90 <- quantile(F_values, 0.9)
> quantile90
      90%
8.346422

```

- c) Rerun your previous simulation, but increase the number of groups to $r = 10$. What happens to the 90% critical value of F_{\max} when the number of groups increases? How can you explain this behavior? (3)

When the number of groups increases, the 90% critical value increases. When there are more groups, there will be more sample variances in that distribution.

```

> n <- 8
> r <- 10
> B <- 50000
>
> F_values <- numeric(B)
>
> for (x in 1:B) {
+   data <- rnorm(n * r)
+   labels <- numeric(n * r)
+   for (i in 1:r) {
+     labels[((i-1)*n + 1):(i*n)] <- i
+   }
+
+   group_variances <- tapply(data, labels, var)
+   F_values[x] <- max(group_variances) / min(group_variances)
+ }
>
> quantile90 <- quantile(F_values, 0.9)
> quantile90
    90%
11.2445

```

d) Run your simulation again with $r = 6$, but this time increase the sample size in each group to $n = 12$. How does increasing the sample size affect the 90% critical value of F_{\max} ? How can you explain this behavior? (3)

When we increase sample size, the 90% critical value of F_{\max} decreases from 8.39365 to 5.163849. Increasing n (the sample size) makes the sample variance more precise and stable because it reduces the extreme variances and shrinks the spread between the maximum and minimum variances. Also, 90% critical value of F_{\max} decreases as n increases.

```
> n <- 12
> r <- 6
> B <- 50000
>
> F_values <- numeric(B)
>
> for (x in 1:B) {
+   data <- rnorm(n * r)
+   labels <- numeric(n * r)
+   for (i in 1:r) {
+     labels[((i-1)*n + 1):(i*n)] <- i
+   }
+
+   group_variances <- tapply(data, labels, var)
+   F_values[x] <- max(group_variances) / min(group_variances)
+ }
>
> quantile90 <- quantile(F_values, 0.9)
> quantile90
  90%
5.208546
```

2. We will revisit the setting of Lab 11B, in which we wrote a program to estimate $Q_{0.95;3,12}$, the 95th quantile of the studentized range distribution with $r = 3$ ranges and $df = r(n - 1)$ degrees of freedom. In our example, take $n = 5$. (18 pts)

- a) Turn your work from Lab 11B into a function that accepts four arguments: a probability (e.g. 0.95 to request the 95th quantile), r , n , and B , the number of simulated data sets. Demonstrate that your function works by reproducing the result obtained at the end of Lab 11B: an estimate of $Q_{0.95;3,12}$ based on 10,000 simulations. (3)

```

> simulate_quantile <- function(prob, r, n, B) {
+   mu <- 100
+   sigma <- 10
+
+   Q_values <- numeric(B)
+
+   for (x in 1:B) {
+     counter <- 0
+     data <- numeric(r * n)
+     groups <- numeric(r * n)
+
+     for (a in 1:r) {
+       for (b in 1:n) {
+         counter <- counter + 1
+         data[counter] <- rnorm(1, mu, sigma)
+         groups[counter] <- a
+       }
+     }
+
+     fitted <- aov(data ~ as.factor(groups))
+     mse <- summary(fitted)[[1]][2, 3]
+
+     group_means <- tapply(data, groups, mean)
+
+     Y_max <- max(group_means)
+     Y_min <- min(group_means)
+
+     Q_values[x] <- (Y_max - Y_min) / sqrt(mse / n)
+   }
+
+   quantile(Q_values, prob)
+ }
>
>
>
> prob <- 0.95
> r <- 3
> n <- 5
> B <- 10000
> estimated <- simulate_quantile(prob, r, n, B)
> estimated
      95%
3.810836

```

- b) Use your function with $n = 5$ and $B = 10,000$ to investigate what happens to the 95th percentile of the studentized range distribution as the number of groups (r) increases. Use the following table as a template, and round your answers to 4 decimal places. (4)

r	3	4	5	6	7
$Q_{.95;r,r(n-1)}$	3.8108	3.9850	4.0569	4.3821	4.5019

- c) Use your function with $r = 3$ and $B = 10,000$ to investigate what happens to the 95th percentile of the studentized range distribution as the number of observations in each group increases. Use the following table as a template, and round your answers to 4 decimal places. (4)

n	5	6	7	8	9
$Q_{.95;r,r(n-1)}$	3.7917	3.6260	3.6625	3.5546	3.5169

- d) Use your function with $r = 3$, $n = 5$, and $B = 10,000$ to investigate what happens to the percentiles of the studentized range distribution as the desired probability (p) increases. Use the following table as a template, and round your answers to 4 decimal places. (4)

p	0.8	0.85	0.9	0.95	0.99
$Q_{p;r,r(n-1)}$	2.5809	2.8470	3.1572	3.7549	5.0653

- e) Summarize the behavior of the simulation-based quantiles as r , n , and p change. (3)

Increasing r leads to higher quantiles, increase n results to decreased quantiles, and increased p results in larger quantiles.

3. We will revisit Lab 11C, in which we wrote a simulation investigating the type I error rate of ANOVA. We saw that when there were $r = 3$ groups and the design was balanced ($n_1 = n_2 = n_3 = 10$), we could violate certain assumptions of the ANOVA model (normality and constant variance) and still maintain a type I error rate that was close to the desired 5%. We will now investigate cases where the design is not balanced. (12 pts)

- a) First we will generate data when all ANOVA assumptions are satisfied. Update your simulation from Lab 11C to accommodate different group sample sizes. One suggestion is to declare a sample size vector $n_{vec} = [n_1 \ n_2 \ n_3]$ to be used when generating data using `rnorm(.)`. Set $r = 3$, and $n_{vec} = [5 \ 8 \ 10]$, and generate a data vector of length 23 (i.e. $5 + 8 + 10$) from $N(0, 1)$. The goal is to generate n_1 responses for group 1, n_2 responses for group 2, and n_3 observations for group 3. You will want to create a second vector of length 23 to store group labels. The labels themselves are not important, but let's label the groups with digits 5, 8, and 10. Create a 23×2 data frame composed of the data vector you generated and the vector of group labels. Print the data frame to convince me that you have done it correctly. (2)


```
> data_frame
```

	y	label
1	-0.6011400068	1
2	0.0840806155	1
3	-0.0008604606	1
4	1.4753368781	1
5	1.4384679437	1
6	1.7022775043	2
7	-0.0232442064	2
8	-0.1089317569	2
9	0.8640494064	2
10	0.0293703782	2
11	-0.1033936449	2
12	0.6849721584	2
13	-0.2641473152	2
14	-0.7912946733	3
15	-1.5107434823	3
16	-1.0173776838	3
17	0.3603341292	3
18	-0.9922075202	3
19	-0.1608082241	3
20	1.0812370482	3
21	0.3967455577	3
22	-0.2880478530	3
23	0.1979181016	3

- b) Fit a one-way ANOVA model to the data contained in your data frame. Extract the p-value for the ANOVA F -test and create a binary “reject” variable to note whether the test rejects H_0 . (2)

```
> res_aov
```

```
Call:
```

```
aov(formula = y ~ as.factor(label), data = data_frame)
```

```
Terms:
```

	as.factor(label)	Residuals
Sum of Squares	1.06483	33.33697
Deg. of Freedom	2	20

```
Residual standard error: 1.291065
```

```
Estimated effects may be unbalanced
```

```
> |
      Df Sum Sq Mean Sq F value Pr(>F)
as.factor(label)  2    1.06   0.5324    0.319    0.73
Residuals       20   33.34   1.6668
```

```
> TOE <- mean(reject)
```

```
> TOE
```

```
[1] 0.0514
```

```
> |
```

Since $P\text{-value} = 0.0514 > 0.05$, we fail to reject H_0 . There is not enough statistical evidence that 5% significance level to conclude that group means are significantly different.

- c) Update your program to run in a loop. Generate data $B = 10,000$ times and store 10,000 realizations of the binary rejection variable. Present the mean of your “reject” vector. Note: because we have not violated any ANOVA assumptions yet, `mean(reject)` should return a value close to 0.05. (2)

```
> TOE <- mean(reject)
```

```
> TOE
```

```
[1] 0.0499
```

```
> |
```

- d) We will now violate the equal variance assumption, running an updated simulation with $B = 10,000$. This time, rather than calling `rnorm(.)` with $mean=0$ and $sd=1$, we will set $sd=1/n_i$ where n_i is the group sample size (5, 8, or 10). What is the simulation-based type I error rate? (2)

```

> reject <- ifelse(pvals <= 0.05, 1, 0)
> TOE <- mean(reject)
> TOE
[1] 0.1101

```

- e) Run another $B = 10,000$ simulations, this time with the standard deviation of the normal random variable set to $1/(n_i^2)$. Give the simulation-based type I error rate. (2)

```

> reject <- ifelse(pvals <= 0.05, 1, 0)
> TOE <- mean(reject)
> TOE
[1] 0.1787

```

- f) Describe the impact of violating the equal variance assumption when the group sample sizes are not the same, as opposed to when the group sample sizes were balanced (as in Lab 11C). (2)

Unbalanced groups or unequal variances result in increased bias and unpredictable Type 1 error rates. Balanced groups also result from unequal variance but Type 1 error rate is more closer to 0.05.

HW 6 code:

```
"""  
1. Suppose we plan to carry out the one-way ANOVA procedure on a data set with a continuous  
response variable and a single categorical predictor variable having  $r=6$  levels.  
Before fitting the model, the lead investigator wishes to use Hartley's F-max test,  
with  $\alpha=0.1$ , to address the following hypotheses. (10 pts)
```

```
H_0:  $\sigma_1^2 = \sigma_2^2 = \dots = \sigma_6^2$   
H_a: not all equal
```

```
Assume that the response variables in each group are normally distributed with mean 0  
and standard deviation 1, and that we have a balanced design with  $n=8$  observations per group.  
"""
```

```
"""  
a. Write a simulation that generates  $n$  observations from  $N(0,1)$  in each of the  $r$  groups,  
keeping track of which group each observation belongs to. You should have a total of  
 $nr=48$  observations. Obtain group-specific sample variances  $S_i^2$ , and calculate the  
observed value of  $F_{\max}$ , where this variable is defined as:
```

```
Print your value of  $F_{\max}$ . (2)
```

```
"""
```

```
n <- 8  
r <- 6
```

```
observations <- mnorm(n * r, mean = 0, sd = 1)  
labels <- rep(1:r, each = n)
```

```
data <- data.frame(value = observations, group = factor(labels))
```

```
group_variances <- tapply(data$value, data$group, var)
```

```
F_max <- max(group_variances) / min(group_variances)  
F_max
```

```
"""  
b. Perform the simulation from the previous part  $B=50,000$  times, storing 50,000 values  
of  $F_{\max}$ . Use this collection of values to estimate the 90th quantile of the distribution  
of  $F_{\max}$ . (2)
```

```
"""
```

```
n <- 8  
r <- 6  
B <- 50000
```

```
F_values <- numeric(B)
```

```
for (x in 1:B) {  
  data <- rnorm(n * r)  
  labels <- numeric(n * r)  
  for (i in 1:r) {  
    labels[((i-1)*n + 1):(i*n)] <- i  
  }  
}
```

```
group_variances <- tapply(data, labels, var)  
F_values[x] <- max(group_variances) / min(group_variances)  
}
```

```
quantile90 <- quantile(F_values, 0.9)  
quantile90
```

```
"""  
c. Rerun your previous simulation, but increase the number of groups to  $r=10$ .  
What happens to the 90% critical value of  $F_{\max}$  when the number of groups increases?  
How can you explain this behavior? (3)
```

```
"""
```

```
n <- 8  
r <- 10  
B <- 50000
```

```
F_values <- numeric(B)
```

```
for (x in 1:B) {  
  data <- rnorm(n * r)  
  labels <- numeric(n * r)
```

```

for (i in 1:r) {
  labels[((i-1)*n + 1):(i*n)] <- i
}

group_variances <- tapply(data, labels, var)
F_values[x] <- max(group_variances) / min(group_variances)
}

quantile90 <- quantile(F_values, 0.9)
quantile90

```

```

"""
d. Run your simulation again with r=6, but this time increase the sample size in each
group to n=12. How does increasing the sample size affect the 90% critical value of
F_max? How can you explain this behavior? (3)
"""

```

```

n <- 12
r <- 6
B <- 50000

```

```

F_values <- numeric(B)

```

```

for (x in 1:B) {
  data <- rnorm(n * r)
  labels <- numeric(n * r)
  for (i in 1:r) {
    labels[((i-1)*n + 1):(i*n)] <- i
  }
}

```

```

group_variances <- tapply(data, labels, var)
F_values[x] <- max(group_variances) / min(group_variances)
}

```

```

quantile90 <- quantile(F_values, 0.9)
quantile90

```

```

"""
We will revisit the setting of Lab 11B, in which we wrote a program to estimate
Q (0.95;3,12), the 95th quantile of the studentized range distribution with r=3
ranges and df=r(n-1) degrees of freedom. In our example, take n=5. (18 pts)
"""

```

```

"""
a. Turn your work from Lab 11B into a function that accepts four arguments:
a probability (e.g. 0.95 to request the 95th quantile), r, n, and B, the
number of simulated data sets. Demonstrate that your function works by
reproducing the result obtained at the end of Lab 11B: an estimate of
Q (0.95;3,12) based on 10,000 simulations. (3)
"""

```

```

simulate_quantile <- function(prob, r, n, B) {
  mu <- 100
  sigma <- 10
}

```

```

Q_values <- numeric(B)

```

```

for (x in 1:B) {
  counter <- 0
  data <- numeric(r * n)
  groups <- numeric(r * n)
}

```

```

for (a in 1:r) {
  for (b in 1:n) {
    counter <- counter + 1
    data[counter] <- rnorm(1, mu, sigma)
    groups[counter] <- a
  }
}

```

```

fitted <- aov(data ~ as.factor(groups))
mse <- summary(fitted)[[1]][2, 3]

```

```

group_means <- tapply(data, groups, mean)

```

```

Y_max <- max(group_means)
Y_min <- min(group_means)

```

```

    Q_values[x] <- (Y_max - Y_min) / sqrt(mse / n)
  }

```

```

quantile(Q_values, prob)
}

```

```

prob <- 0.95
r <- 3
n <- 5
B <- 10000
estimated <- simulate_quantile(prob, r, n, B)
estimated

```

```

####
b. Use your function with n=5 and B=10,000 to investigate what happens to the
95th percentile of the studentized range distribution as the number of groups (r)
increases. Use the following table as a template, and round your answers to 4
decimal places. (4)
####

```

```

prob <- 0.95
r <- 4
n <- 5
B <- 10000
estimated <- simulate_quantile(prob, r, n, B)
estimated

```

```

prob <- 0.95
r <- 4
n <- 5
B <- 10000
estimated <- simulate_quantile(prob, r, n, B)
estimated

```

```

prob <- 0.95
r <- 6
n <- 5
B <- 10000
estimated <- simulate_quantile(prob, r, n, B)
estimated

```

```

prob <- 0.95
r <- 7
n <- 5
B <- 10000
estimated <- simulate_quantile(prob, r, n, B)
estimated

```

```

####
c. Use your function with r=3 and B=10,000 to investigate what happens to the 95th
percentile of the studentized range distribution as the number of observations in
each group increases. Use the following table as a template, and round your answers
to 4 decimal places. (4)
####

```

```

prob <- 0.95
r <- 3
n <- 5
B <- 10000
estimated <- simulate_quantile(prob, r, n, B)
estimated

```

```

prob <- 0.95
r <- 3
n <- 6
B <- 10000
estimated <- simulate_quantile(prob, r, n, B)
estimated

```

```

prob <- 0.95

```

```

r <- 3
n <- 7
B <- 10000
estimated <- simulate_quantile(prob, r, n, B)
estimated

```

```

prob <- 0.95
r <- 3
n <- 8
B <- 10000
estimated <- simulate_quantile(prob, r, n, B)
estimated

```

```

prob <- 0.95
r <- 3
n <- 9
B <- 10000
estimated <- simulate_quantile(prob, r, n, B)
estimated

```

```

"""
d. Use your function with r=3, n=5, and B=10,000 to investigate what happens to
the percentiles of the studentized range distribution as the desired probability
(p) increases. Use the following table as a template, and round your answers to
4 decimal places. (4)
"""

```

```

prob <- 0.80
r <- 3
n <- 5
B <- 10000
estimated <- simulate_quantile(prob, r, n, B)
estimated

```

```

prob <- 0.85
r <- 3
n <- 5
B <- 10000
estimated <- simulate_quantile(prob, r, n, B)
estimated

```

```

prob <- 0.90
r <- 3
n <- 5
B <- 10000
estimated <- simulate_quantile(prob, r, n, B)
estimated

```

```

prob <- 0.95
r <- 3
n <- 5
B <- 10000
estimated <- simulate_quantile(prob, r, n, B)
estimated

```

```

prob <- 0.99
r <- 3
n <- 5
B <- 10000
estimated <- simulate_quantile(prob, r, n, B)
estimated

```

```

"""
3. We will revisit Lab 11C, in which we wrote a simulation investigating the type
I error rate of ANOVA. We saw that when there were r=3 groups and the design was
balanced (n1=n2=n3=10), we could violate certain assumptions of the ANOVA model
(normality and constant variance) and still maintain a type I error rate that
was close to the desired 5%. We will now investigate cases where the design
is not balanced. (12 pts)
"""

```

```

"""
a. First we will generate data when all ANOVA assumptions are satisfied.
Update your simulation from Lab 11C to accommodate different group sample sizes.
One suggestion is to declare a sample size vector n_vec=[n_1&n_2&n_3] to be
used when generating data using rnorm(.). Set r=3, and n_vec=[5&8&10], and
generate a data vector of length 23 (i.e. 5+8+10) from N(0,1). The goal is to
generate n_1 responses for group 1, n_2 responses for group 2, and n_3 observations
for group 3. You will want to create a second vector of length 23 to store group
labels. The labels themselves are not important, but let's label the groups with
digits 5, 8, and 10. Create a 23x2 data frame composed of the data vector you
generated and the vector of group labels. Print the data frame to convince me
that you have done it correctly. (2)
"""

```

```

B <- 10000
r <- 3
n_vec <- c(5, 8, 10)
n_total <- sum(n_vec)
pvals <- numeric(B)

for (x in 1:B) {
  y <- numeric(n_total)
  label <- integer(n_total)
  index <- 1

  for (i in 1:r) {
    for (j in 1:n_vec[i]) {
      y[index] <- rnorm(1, mean = 0, sd = 1)
      label[index] <- i
      index <- index + 1
    }
  }

  data_frame <- data.frame(y, label)
  res_aov <- aov(y ~ as.factor(label), data = data_frame)
  pvals[x] <- summary(res_aov)[[1]]$`Pr(>F)`[1]
}
res_aov
reject <- ifelse(pvals <= 0.05, 1, 0)
TOE <- mean(reject)
TOE
data_frame

```

```

"""
b. Fit a one-way ANOVA model to the data contained in your data frame. Extract the p-value for the ANOVA F-test and create
a binary "reject" variable to note whether the test rejects H_0. (2)
"""

```

```

B <- 10000
r <- 3
n_vec <- c(5, 8, 10)
n_total <- sum(n_vec)
pvals <- numeric(B)

for (x in 1:B) {
  y <- numeric(n_total)
  label <- integer(n_total)
  index <- 1

  for (i in 1:r) {
    for (j in 1:n_vec[i]) {
      y[index] <- rnorm(1, mean = 0, sd = 1)
      label[index] <- i
      index <- index + 1
    }
  }

  data_frame <- data.frame(y, label)
  res_aov <- aov(y ~ as.factor(label), data = data_frame)
  pvals[x] <- summary(res_aov)[[1]]$`Pr(>F)`[1]
}

res_aov
summary(res_aov)

```

```

"""
c. Update your program to run in a loop. Generate data B=10,000 times and store 10,000

```


realizations of the binary rejection variable. Present the mean of your "reject" vector.

Note: because we have not violated any ANOVA assumptions yet, mean(reject) should return a value close to 0.05. (2)

```
#####
B <- 10000
r <- 3
n_vec <- c(5, 8, 10)
n_total <- sum(n_vec)
pvals <- numeric(B)

for (x in 1:B) {
  y <- numeric(n_total)
  label <- integer(n_total)
  index <- 1

  for (i in 1:r) {
    for (j in 1:n_vec[i]) {
      y[index] <- rnorm(1, mean = 0, sd = 1)
      label[index] <- i
      index <- index + 1
    }
  }

  data_frame <- data.frame(y, label)
  res_aov <- aov(y ~ as.factor(label), data = data_frame)
  pvals[x] <- summary(res_aov)[[1]]$`Pr(>F)`[1]
}

res_aov
summary(res_aov)
reject <- ifelse(pvals <= 0.05, 1, 0)
TOE <- mean(reject)
TOE
```

```
#####
d. We will now violate the equal variance assumption, running an updated simulation with B=10,000.
This time, rather than calling rnorm(.) with mean=0 and sd=1, we will set sd=1/n_i where n_i is the group sample size
(5, 8, or 10). What is the simulation-based type I error rate? (2)
#####
```

```
B <- 10000
r <- 3
n_vec <- c(5, 8, 10)
n_total <- sum(n_vec)
pvals <- numeric(B)

for (x in 1:B) {
  y <- numeric(n_total)
  label <- integer(n_total)
  index <- 1

  for (i in 1:r) {
    for (j in 1:n_vec[i]) {
      y[index] <- rnorm(1, mean = 0, sd = 1 / n_vec[i])
      label[index] <- i
      index <- index + 1
    }
  }

  data_frame <- data.frame(y, label)
  res_aov <- aov(y ~ as.factor(label), data = data_frame)
  pvals[x] <- summary(res_aov)[[1]]$`Pr(>F)`[1]
}

res_aov
summary(res_aov)
reject <- ifelse(pvals <= 0.05, 1, 0)
TOE <- mean(reject)
TOE
```

```
#####
e. Run another B=10,000 simulations, this time with the standard deviation of the normal random variable set to 1/(n_i^2).
Give the simulation-based type I error rate. (2)
#####
```

```

B <- 10000
r <- 3
n_vec <- c(5, 8, 10)
n_total <- sum(n_vec)
pvals <- numeric(B)

for (x in 1:B) {
  y <- numeric(n_total)
  label <- integer(n_total)
  index <- 1

  for (i in 1:r) {
    for (j in 1:n_vec[i]) {
      y[index] <- rnorm(1, mean = 0, sd = 1 / (n_vec[i]^2))
      label[index] <- i
      index <- index + 1
    }
  }

  data_frame <- data.frame(y, label)
  res_aov <- aov(y ~ as.factor(label), data = data_frame)
  pvals[x] <- summary(res_aov)[[1]]$`Pr(>F)`[1]
}

res_aov
summary(res_aov)
reject <- ifelse(pvals <= 0.05, 1, 0)
TOE <- mean(reject)
TOE

"""
f) Describe the impact of violating the equal variance assumption when the group sample sizes are not the same, as opposed to when the
group sample sizes were balanced (as in Lab 11C). (2)
"""

```