

HW 6

Andrew Jowe

1

- a. The overall trend for factor A is that as we increase the dosage of the first ingredient, the number of hours of relief increases.
- b. The overall trend for factor B is that as we increase the dosage of the second ingredient, the number of hours of relief increases.
- c. I would use the smallest multiplier out of the three for precision since all multipliers can be used.
- d. If we assume no interaction effect, I would say increasing factor A is generally more important for increasing the number of hours of relief since there is a bigger increase from low to medium and medium to high for factor A compared to factor B.

2

- a. The 95% confidence interval is $[-0.9558324, -0.5441676]$.
- b. We are 95% certain that the difference of hours of relief on average between low dosage of factor A and low dosage of factor B is around 0.5441676 to 0.9558324 lower. This means for low dosage of either drug, people are better off with drug B.
- c. The 99% confidence interval for $\mu_{1.} - \mu_{3.}$ is $[-6.2566421, -5.6433579]$. The 99% confidence interval for $\mu_{.1} - \mu_{.3}$ is $[-4.6566421, -4.0433579]$.
- d. The confidence interval for $\mu_{1.} - \mu_{3.}$ tells us that we are 99% certain that difference of hours of relief on average between a low dosage of drug A and a high dosage of drug A is around 5.6433579 to 6.2566421 lower.
- e. The 90% confidence interval is $[-2.3958772, -1.8041228]$. We are 90% certain that the difference of hours of relief on average between low dosage of factor A and high dosage of factor A is around 1.8041228 to 2.3958772 lower for those who took low dosage of factor B.

3

- a. Yes, because 0 is not in our interval.
- b. Yes, because 0 is not in our interval.
- c. Yes, because 0 is not in our interval.
- d. I would have them take a high dosage of drug A and drug B because we have clearly shown higher dosages of both drugs yields better results.

4

- a. Yes, a weighted confidence interval changes the bounds compared to the unweighted confidence interval. Lets look at the equation for confidence interval:

$$- \sum \sum c_{ij} \bar{Y}_{ij} \pm t_{\alpha/2, df(SSE)} \sqrt{MSE \sum \sum \frac{c_{ij}^2}{n_{ij}}}$$

- We can tell that the weights can change not only the center of the confidence interval but also the spread of the confidence interval. Both of these will affect the bounds.
- b. I would suggest equal weights because unequal weights will make the result difficult to interpret.
- c. One reason to give unequal weights for smokers is due to the unequal sample sizes. Since the sample size of smokers is larger, we can assign a larger weight to this as it is more reliable information and indicates greater precision.
- d. One reason to give equal weights is because unequal weights makes interpretation of the result difficult.

5

- a. We are 95% certain that females on average have around 0.84 to 2.18 less lung capacity compared to males.
- b. It would be unusual because -3 is not in our confidence interval.
- c. We are 95% certain that nonsmokers on average have around 1.82 to 2.45 more lung capacity compared to smokers.
- d. Yes, because 0 is not in our confidence interval.

6

- a. Lets interpret the confidence interval for $\mu_{11} - \mu_{12}$. The difference of lung capacity on average between non smokers and smokers is 0.4 to 2.6 more for females.
- b. No, because our interval from (a) includes the interaction effect of A on B compared to (5c) which only accounted for factor B.
- c. No, because our interval from (a) includes the interaction effect of A on B compared to (5c) which only accounted for factor B.
- d. $\beta_0 = 7.30$ is our default modifier regardless of which group the subject belongs to. $\beta_1 = 1.32$ is our modifier if the subject is within Male group of factor A. $\beta_2 = -2.180$ is our modifier if the subject is within the smokers group of factor B.

7

- a. True. If there is no interaction effect, we don't expect a change in confidence interval if we choose to look at different groups in factor B when contrasting two means in factor A. We might as well compute the confidence interval for a contrast of $\mu_{i.}$ or $\mu_{.j}$.

- b. False. The multiplier used should be the smallest multiplier out of all the multipliers. Even though Bonferroni multiplier can be used for any contrast, it is not necessarily the smallest.
- c. False. These multipliers are used for creating confidence intervals that contrast means, not for finding the true population mean.
- d. True. The amount of parameters to estimate for the regression model is the same as the model we chose. In our case, we chose the no interaction model, which has $a + b - 1$ parameters.

I

- a. The 95% confidence interval for μ_1 is [84.1829, 175.1505].
- b. The 90% confidence interval for $\mu_1 - \mu_2$ is [-30.6478, -21.8686]. The 90% confidence interval for $\mu_1 - \mu_3$ is [-97.1478, -88.3686].
- c. The 99% confidence interval for $\mu_{11} - \mu_{21}$ is [7.923, 34.0407]. The 99% confidence interval for $\mu_{13} - \mu_{23}$ is [145.923, 172.0407].
- d. Our results from C shows that the average days it takes for a Small systems, Exp experience programmer is around 7.923 to 34.0407 more than Both systems, Exp experience programmer. The average days it takes for a Small systems, New experience programmer is around 145.923 to 172.0407 more than Both systems, New experience programmer.
- e. The largest difference of days we would expect for the average programmer with different types of experience is around 97.1478.
- f. Yes, experienced programmers do have significantly lower averages than other categories, because none of the confidence intervals in b contain 0.

II

a.
$$Y_{ijk} = b_0 + b_1X_{A,Small} + b_2X_{B,Med} + b_3X_{B,New} + b_4X_{A,Small}X_{B,Med} + b_5X_{A,Small}X_{B,New} + \epsilon_{ijk}$$

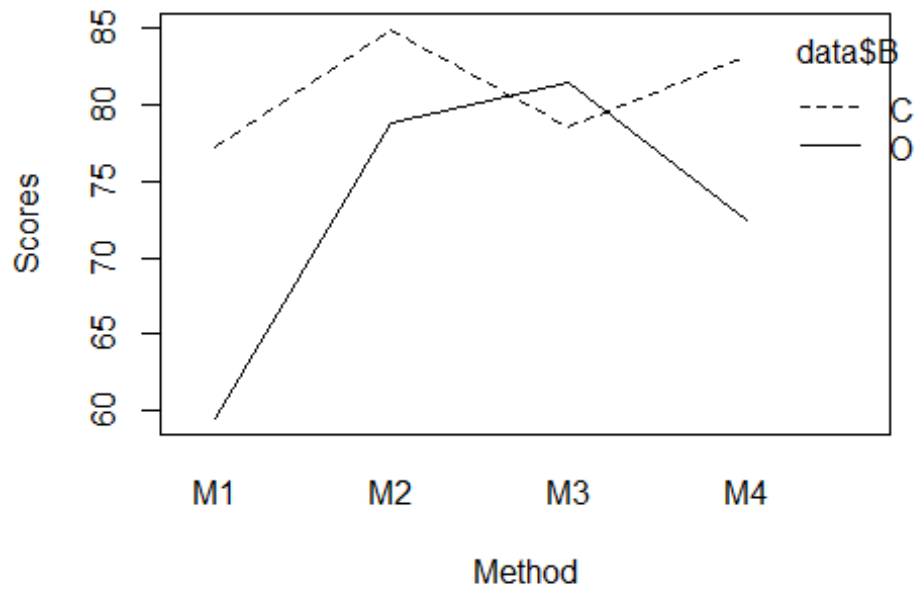
$$Y_{ijk} = 38.75 + 21.75X_{A,2} + 6X_{B,2} + 23.5X_{B,3} + 40X_{A,2}X_{B,2} + 138X_{A,2}X_{B,3} + \epsilon_{ijk}$$

There are 6 parameters total.

- b. b_0 represents the estimated average for Medium Systems, High (Exp) Experience
- c. b_1 represents the estimated average difference between Medium Systems, High Experience and Small Systems, High Experience ($\mu_{21} - \mu_{11}$).
- d. b_4 represents the interaction effect between Small Systems and Medium Experience.
- e. b_5 represents the interaction effect between Small Systems and New Experience.

III.a

Assuming that factor *A* is Method and factor *B* is Room.



III.b.1

Here are the means for factor *A* alone:

	x
M1	70.100
M2	81.875
M3	80.000
M4	77.850

III.b.2

Here are the means for factor *B* alone:

	x
C	80.84783
O	74.40476

III.b.3

Here are the means for both factors A and B :

	x
C	80.84783
O	74.40476

III.c.1

Here are the counts for factor A alone:

	x
M1	20
M2	24
M3	24
M4	20

III.c.2

Here are the counts for factor B alone:

	x
C	46
O	42

III.c.3

Here are the means for both factors A and B :

	C	O
M1	12	8
M2	12	12
M3	12	12
M4	10	10

III.d

The full model is: $Y_{ijk} = \mu_{..} + \gamma_i + \delta_j + (\gamma\delta)_{ij} + \epsilon_{ijk}$

The reduced model is: $Y_{ijk} = \mu_{..} + \gamma_i + \delta_j + \epsilon_{ijk}$

H_0 : There is no interaction effect. Don't use the full model.

H_a : There is interaction effect. Use the full model.

$$\alpha = 0.01$$

$$F_s = 3.1869621$$

$$p = 0.0281564$$

Since $p > \alpha$, we accept H_0 . Therefore, we do not use the full model.

III.e

$$Y_{ijk} = b_0 + b_1X_{A,M2} + b_2X_{A,M3} + b_3X_{A,M4} + b_4X_{B,O} + \epsilon_{ijk}$$

$$Y_{ijk} = 72.9771 + 12.4943X_{A,M2} + 10.6193X_{A,M3} + 8.4693X_{A,M4} + -7.1927X_{B,O} + \epsilon_{ijk}$$

III.f

We would choose the Tukey multiplier because that multiplier assumes that we are making all possible pairwise confidence intervals. We can have the multiplier only account for factor B.

Appendix

```
# Functions
# Plus minus function
plus.minus = function(value) {
  return(c(-value, value))
}

# Find means
find.means = function(the.data, fun.name = mean){
  a = length(unique(the.data[,2]))
  b = length(unique(the.data[,3]))
  means.A = by(the.data[,1], the.data[,2], fun.name)
  means.B = by(the.data[,1], the.data[,3], fun.name)
  means.AB = by(the.data[,1], list(the.data[,2], the.data[,3]), fun.name)
  MAB = matrix(means.AB, nrow = b, ncol = a, byrow = TRUE)
  colnames(MAB) = names(means.A)
  rownames(MAB) = names(means.B)
  MA = as.numeric(means.A)
  names(MA) = names(means.A)
  MB = as.numeric(means.B)
  names(MB) = names(means.B)
```

```

MAB = t(MAB)
results = list(A = MA, B = MB, AB = MAB)
return(results)
}

# Give me multipliers
find.mult = function(alpha,a,b,dfSSE,g,group){
  if(group == "A"){
    Tuk = round(qtukey(1-alpha,a,dfSSE)/sqrt(2),3)
    Bon = round(qt(1-alpha/(2*g), dfSSE ),3)
    Sch = round(sqrt((a-1)*qf(1-alpha, a-1, dfSSE)),3)
  }else if(group == "B"){
    Tuk = round(qtukey(1-alpha,b,dfSSE)/sqrt(2),3)
    Bon = round(qt(1-alpha/(2*g), dfSSE ),3)
    Sch = round(sqrt((b-1)*qf(1-alpha, b-1, dfSSE)),3)
  }else if(group == "AB"){
    Tuk = round(qtukey(1-alpha,a*b,dfSSE)/sqrt(2),3)
    Bon = round(qt(1-alpha/(2*g), dfSSE ),3)
    Sch = round(sqrt((a*b-1)*qf(1-alpha, a*b-1, dfSSE)),3)
  }
  results = c(Bon, Tuk,Sch)
  names(results) = c("Bonferroni","Tukey","Scheffe")
  return(results)
}

give.me.CI = function(the.data,MSE,equal.weights = TRUE,multiplier,group,cs){
  if(sum(cs) != 0 & sum(cs !=0 ) != 1){
    return("Error - you did not input a valid contrast")
  }else{
    the.means = find.means(the.data)
    the.ns =find.means(the.data,length)
    nt = nrow(the.data)
    a = length(unique(the.data[,2]))
    b = length(unique(the.data[,3]))
    if(group == "A"){
      if(equal.weights == TRUE){
        a.means = rowMeans(the.means$AB)
        est = sum(a.means*cs)
        mul = rowSums(1/the.ns$AB)
        SE = sqrt(MSE/b^2 * (sum(cs^2*mul)))
        N = names(a.means)[cs!=0]
        CS = paste("(",cs[cs!=0],")",sep = "")
        fancy = paste(paste(CS,N,sep = ""),collapse = "+")
        names(est) = fancy
      } else{
        a.means = the.means$A
        est = sum(a.means*cs)
        SE = sqrt(MSE*sum(cs^2*(1/the.ns$A)))
        N = names(a.means)[cs!=0]
        CS = paste("(",cs[cs!=0],")",sep = "")

```

```

    fancy = paste(paste(CS,N,sep = ""),collapse = "+")
    names(est) = fancy
  }
} else if(group == "B"){
  if(equal.weights == TRUE){
    b.means = colMeans(the.means$AB)
    est = sum(b.means*cs)
    mul = colSums(1/the.ns$AB)
    SE = sqrt(MSE/a^2 * (sum(cs^2*mul)))
    N = names(b.means)[cs!=0]
    CS = paste("(",cs[cs!=0],")",sep = "")
    fancy = paste(paste(CS,N,sep = ""),collapse = "+")
    names(est) = fancy
  } else{
    b.means = the.means$B
    est = sum(b.means*cs)
    SE = sqrt(MSE*sum(cs^2*(1/the.ns$B)))
    N = names(b.means)[cs!=0]
    CS = paste("(",cs[cs!=0],")",sep = "")
    fancy = paste(paste(CS,N,sep = ""),collapse = "+")
    names(est) = fancy
  }
} else if(group == "AB"){
  est = sum(cs*the.means$AB)
  SE = sqrt(MSE*sum(cs^2/the.ns$AB))
  names(est) = "someAB"
}
the.CI = est + c(-1,1)*multiplier*SE
results = c(est,the.CI)
names(results) = c(names(est),"lower bound","upper bound")
return(results)
}
}

# 2.a
a <- 3
b <- 3
nidot <- 12
ndotj <- 12
nT <- 36

alpha <- 0.05
g <- 1

SSE <- 1.63
dfsSE <- nT - a * b
MSE <- SSE / dfsSE

mulldot <- 3.88
mudot1 <- 4.63

```



```

all.mult <- find.mult(alpha = alpha, a = a, b = b,
                      dfSSE = dfSSE, g = g, group = "AB")
the.mult <- min(all.mult)
the.CI <- mu1dot - mudot1 + plus.minus(
  the.mult * sqrt(MSE * (1 / nidot + 1 / ndotj))
)

# 2.c
alpha <- 0.01
g <- 2

mu3dot <- 9.83
mudot3 <- 8.98

all.mult <- find.mult(alpha = alpha, a = a, b = b,
                      dfSSE = dfSSE, g = g, group = "AB")
the.mult <- min(all.mult)
the.CI.1 <- mu1dot - mu3dot + plus.minus(
  the.mult * sqrt(MSE * (1 / nidot + 1 / nidot))
)
the.CI.2 <- mudot1 - mudot3 + plus.minus(
  the.mult * sqrt(MSE * (1 / ndotj + 1 / ndotj))
)

# 2.e
alpha <- 0.1
g <- 1

nij <- 4

mu11 <- 2.48
mu13 <- 4.58

all.mult <- find.mult(alpha = alpha, a = a, b = b,
                      dfSSE = dfSSE, g = g, group = "AB")
the.mult <- min(all.mult)
the.CI <- mu11 - mu13 + plus.minus(
  the.mult * sqrt(MSE * (1 / nij + 1 / nij))
)

# I.a
data <- read.csv("Prog.csv")

data.I.a <- subset(data, type == "Small")
result <- t.test(data.I.a$days)
conf_interval <- round(result$conf.int, 4)

# I.b
alpha <- 0.1

```

```

n <- nrow(data)
a <- length(unique(data$type))
b <- length(unique(data$years))

names(data) = c("Y", "A", "B")
AB = lm(Y ~ A * B, data)
SSE <- AB$residuals ^ 2
dfSSE <- n - a * b
MSE <- SSE / dfSSE

the.mult <- find.mult(alpha, a, b, dfSSE, 1, "B")[2]

CI.1 <- give.me.CI(data, MSE, equal.weights = TRUE, the.mult, "B", c(1, -1,
0))
CI.1 <- round(CI.1, 4)

CI.2 <- give.me.CI(data, MSE, equal.weights = TRUE, the.mult, "B", c(1, 0, -
1))
CI.2 <- round(CI.2, 4)

# I.c
alpha <- 0.01

the.mult <- find.mult(alpha, a, b, dfSSE, 1, "AB")[3]

# Note - i indices are flipped. In R, i = 1 is Both
# However, in the HW, i = 1 is Small
AB.cs <- matrix(0, nrow = a, ncol = b)
AB.cs[2, 1] <- 1
AB.cs[1, 1] <- -1
CI.1 <- give.me.CI(data, MSE, equal.weights = TRUE, the.mult, "AB", AB.cs)
CI.1 <- round(CI.1, 4)

AB.cs <- matrix(0, nrow = a, ncol = b)
AB.cs[2, 3] <- 1
AB.cs[1, 3] <- -1
CI.2 <- give.me.CI(data, MSE, equal.weights = TRUE, the.mult, "AB", AB.cs)
CI.2 <- round(CI.2, 4)

# II
co <- coef(AB)

# III.a
data <- read.csv("Teaching.csv")
names(data) = c("Y", "A", "B")
interaction.plot(x.factor = data$A, trace.factor = data$B, response = data$Y,
xlab = "Method", ylab = "Scores")

```

```

# III.b
the.means <- find.means(data)

# III.b.1
knitr::kable(the.means[1])

# III.b.2
knitr::kable(the.means[2])

# III.b.3
knitr::kable(the.means[2])

# III.c
the.counts <- find.means(data, fun.name = length)

# III.c.1
knitr::kable(the.counts[1])

# III.c.2
knitr::kable(the.counts[2])

# III.c.3
knitr::kable(the.counts[3])

# III.d
the.data <- data
AB = lm(Y ~ A*B,the.data)
A.B = lm(Y ~ A + B,the.data)
all.models = list(AB,A.B)
SSE = t(as.matrix(sapply(all.models,function(M) sum(M$residuals^2))))
colnames(SSE) = c("AB","(A+B)")
rownames(SSE) = "SSE"

# Conduct ANOVA test
anova_result <- anova(A.B, AB)

# Get the F-statistic and p-value
f_statistic <- anova_result$F[2] # Extracting F-statistic from the first row
p_value <- anova_result$Pr[2]   # Extracting p-value from the first row

# III.e
the.params <- round(coef(A.B), 4)
# knitr::kable(the.params)

```