HW#3 Solution

9/26/2019

HW 17.12 (Note the is the data used in Quiz#2, also in Problem 16.11)

```
# HW 17.12
HW11  <- read.table(
  url("https://raw.githubusercontent.com/npmldabook/Stat3119/master/Week2/CH16PR11.txt"))
# rename the variables
names(HW11)<- c("outcome", "machines", "units")

HW11$machines<- as.factor(HW11$machines)
head(HW11)
```

```
##   outcome machines units
## 1   -0.14        1     1
## 2    0.20        1     2
## 3    0.07        1     3
## 4    0.18        1     4
## 5    0.38        1     5
## 6    0.10        1     6
```

```
str(HW11)
```

```
## 'data.frame':    120 obs. of  3 variables:
##  $ outcome : num  -0.14 0.2 0.07 0.18 0.38 0.1 -0.04 -0.27 0.27 -0.21 ...
##  $ machines: Factor w/ 6 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ units   : int  1 2 3 4 5 6 7 8 9 10 ...
```

ANOVA analysis:

```
fit <- aov(outcome ~ machines, data = HW11)
# factor level means= fitted values
summary(fit)
```

```
##              Df Sum Sq Mean Sq F value   Pr(>F)
## machines      5  2.289  0.4579   14.78 3.64e-11 ***
## Residuals   114  3.531  0.0310
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

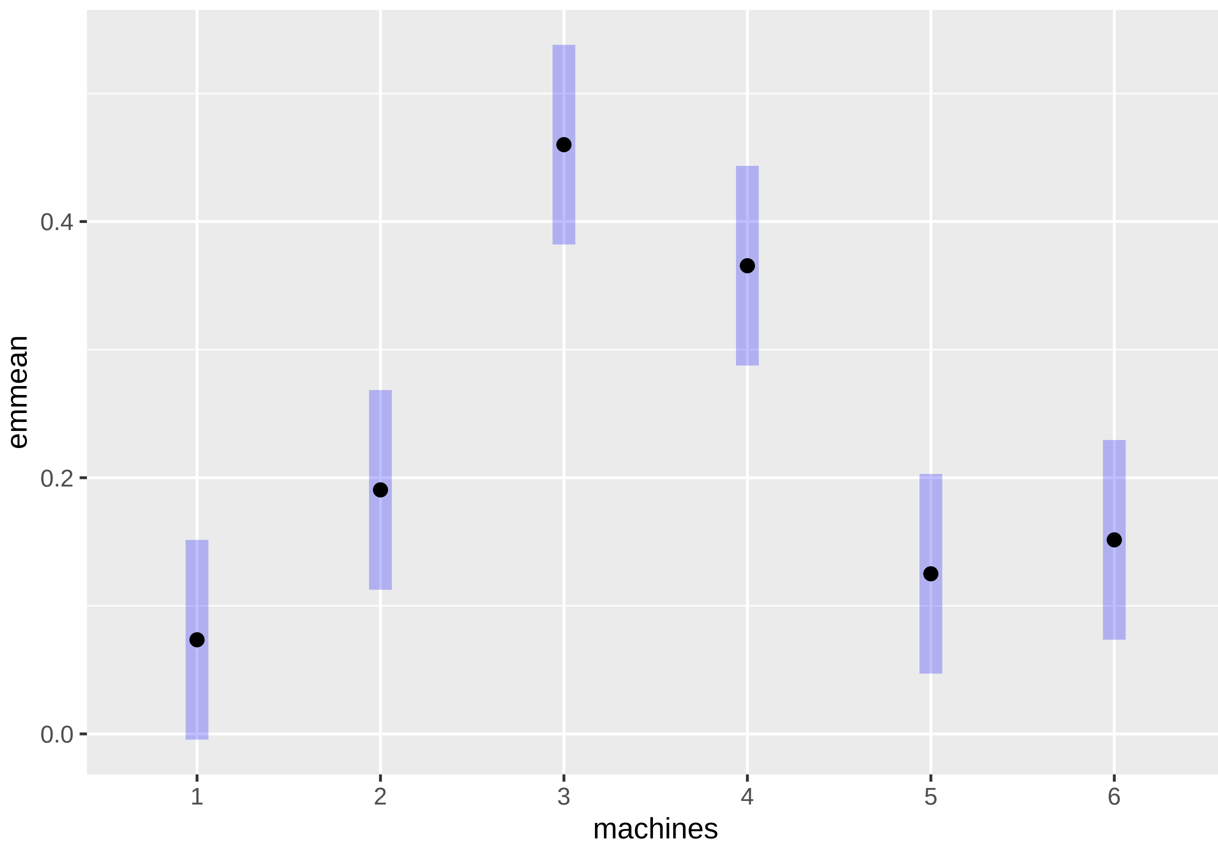
a-b) make main effect plot

```
library(emmeans)

# get the estimate, SE, df and CI
(Est.mean<- emmeans(fit, ~ machines))
```

```
##   machines emmean     SE  df lower.CL upper.CL
##   1        0.0735 0.0394 114 -0.00445    0.151
##   2        0.1905 0.0394 114  0.11255    0.268
##   3        0.4600 0.0394 114  0.38205    0.538
##   4        0.3655 0.0394 114  0.28755    0.443
##   5        0.1250 0.0394 114  0.04705    0.203
##   6        0.1515 0.0394 114  0.07355    0.229
##
## Confidence level used: 0.95
```

```r
# plot the main effects with 95% CIs
plot(Est.mean, horizontal=F)
```



Results: CI for $\mu 1$ $(-.0045, .151)$.

c) get 95% CI for D=mu2-mu1

```r
pairs(Est.mean, adjust = "none" )
```

```
##   contrast estimate     SE  df t.ratio p.value
##   1 - 2     -0.1170 0.0557 114 -2.102  0.0377
##   1 - 3     -0.3865 0.0557 114 -6.945  <.0001
##   1 - 4     -0.2920 0.0557 114 -5.247  <.0001
```

```
##  1 - 5     -0.0515 0.0557 114 -0.925  0.3567
##  1 - 6     -0.0780 0.0557 114 -1.402  0.1638
##  2 - 3     -0.2695 0.0557 114 -4.843  <.0001
##  2 - 4     -0.1750 0.0557 114 -3.145  0.0021
##  2 - 5      0.0655 0.0557 114  1.177  0.2417
##  2 - 6      0.0390 0.0557 114  0.701  0.4849
##  3 - 4      0.0945 0.0557 114  1.698  0.0922
##  3 - 5      0.3350 0.0557 114  6.020  <.0001
##  3 - 6      0.3085 0.0557 114  5.543  <.0001
##  4 - 5      0.2405 0.0557 114  4.322  <.0001
##  4 - 6      0.2140 0.0557 114  3.845  0.0002
##  5 - 6     -0.0265 0.0557 114 -0.476  0.6349
```

Results: We can first obtain estimate is 0.117, with SE =0.0557, df=114 . Then we get 95% CI as follows:

```r
D= 0.1170
SE= 0.0557
df=114
LCI= D - qt(.975, df)*SE
UCI= D + qt(.975, df)*SE
paste("95% CI is (", round(LCI,3), round(UCI,3), ").")
```

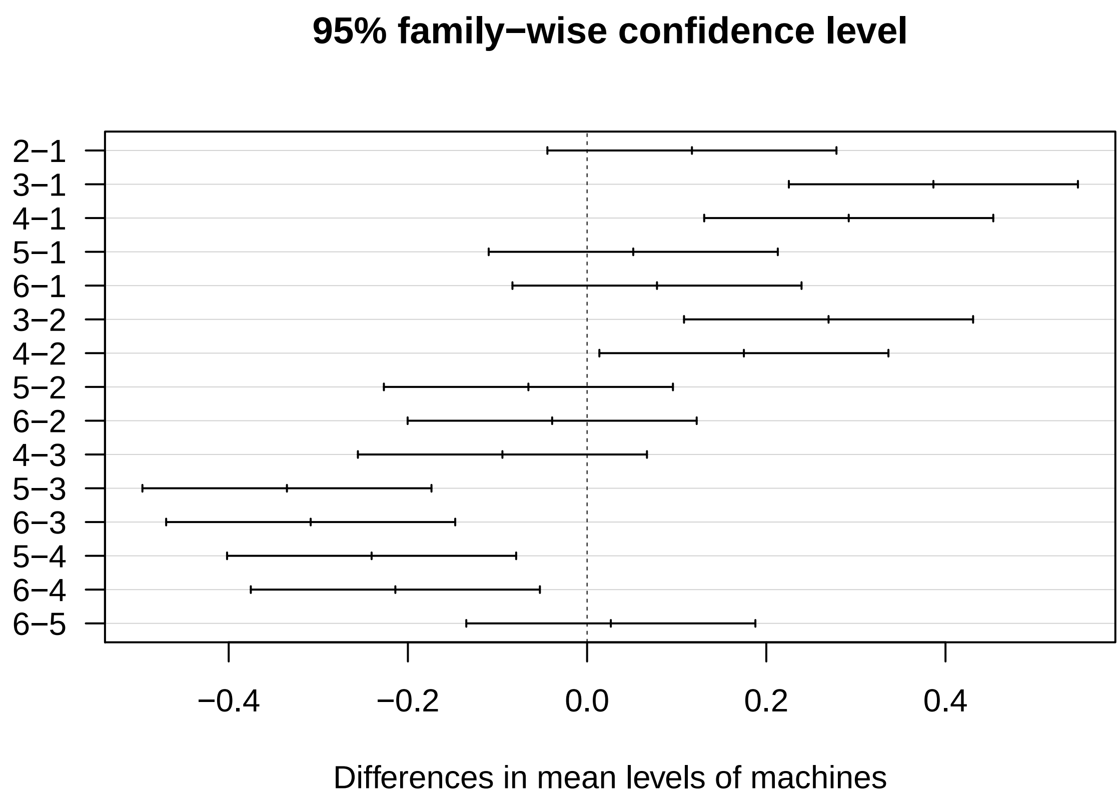
```
## [1] "95% CI is ( 0.007 0.227 )."
```

d. get the pairwise comparison plot

```r
TukeyHSD(fit)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = outcome ~ machines, data = HW11)
##
## $machines
##        diff        lwr        upr     p adj
## 2-1  0.1170 -0.0443194  0.2783194 0.2934937
## 3-1  0.3865  0.2251806  0.5478194 0.0000000
## 4-1  0.2920  0.1306806  0.4533194 0.0000106
## 5-1  0.0515 -0.1098194  0.2128194 0.9392011
## 6-1  0.0780 -0.0833194  0.2393194 0.7260015
## 3-2  0.2695  0.1081806  0.4308194 0.0000588
## 4-2  0.1750  0.0136806  0.3363194 0.0252432
## 5-2 -0.0655 -0.2268194  0.0958194 0.8469184
## 6-2 -0.0390 -0.2003194  0.1223194 0.9815028
## 4-3 -0.0945 -0.2558194  0.0668194 0.5359056
## 5-3 -0.3350 -0.4963194 -0.1736806 0.0000003
## 6-3 -0.3085 -0.4698194 -0.1471806 0.0000029
## 5-4 -0.2405 -0.4018194 -0.0791806 0.0004684
## 6-4 -0.2140 -0.3753194 -0.0526806 0.0026737
## 6-5  0.0265 -0.1348194  0.1878194 0.9968910
```

```r
plot(TukeyHSD(fit), las=1)
```



Results: We can conclude with 95% family confidence that machine 3 and machine 4 are more effective than other machines. However, machines 1,2,5, 6 do not differ significantly.

e-f. simultaneous procedure

```r
#Bonferroni multiple,
 g = 3 ; df= 114
alpha=0.1
(B=qt(1- alpha/(2*g), df))
```

```
## [1] 2.154152
```

Note: We are getting the simultaneous CIs for $D1 = \mu_1 - \mu_4$, $D2 = \mu_1 - \mu_5$ and $D3 = \mu_4 - \mu_5$. From the previous output: D1= -0.2920, D2= -0.0515, D3= 0.2405, SE= 0.0557, we can get 90% simultaneous CI with Bonferroni correction as follows:

```r
CI<- data.frame(D= c(D1= -0.2920 ,D2= -0.0515,D3= 0.2405))
SE= 0.0557
CI$LCI <- CI$D- B*SE
CI$UCI <- CI$D+ B*SE
CI$include0<- ( CI$LCI<0 & CI$UCI>0)
CI
```

```
##           D        LCI         UCI include0
## D1 -0.2920 -0.4119863 -0.17201372    FALSE
## D2 -0.0515 -0.1714863  0.06848628     TRUE
## D3  0.2405  0.1205137  0.36048628    FALSE
```

We may also use contrast() and confint() in emmeans packages to mutiple comparisons

```
List = list(
D1= c(1, 0, 0, -1,0,0),
D2= c(1, 0, 0, 0,-1,0),
D3= c(0, 0, 0, 1,-1,0)
)
(L= contrast(Est.mean, List, adjust="Bonferroni"))
```

```
##  contrast estimate     SE  df t.ratio p.value
##  D1        -0.2920 0.0557 114 -5.247  <.0001
##  D2        -0.0515 0.0557 114 -0.925  1.0000
##  D3         0.2405 0.0557 114  4.322  0.0001
##
## P value adjustment: bonferroni method for 3 tests
```

```
confint(L, level=0.9)
```

```
##  contrast estimate     SE  df lower.CL upper.CL
##  D1        -0.2920 0.0557 114   -0.412  -0.1721
##  D2        -0.0515 0.0557 114   -0.171   0.0684
##  D3         0.2405 0.0557 114    0.121   0.3604
##
## Confidence level used: 0.9
## Conf-level adjustment: bonferroni method for 3 estimates
```

Results: From the 90% simultaneous CI , we can check if CI contains 0 or not. This results show that D1 and D3 are significantly different from 0, suggesting μ_4 is significantly different from μ_1 and μ_5 with a family confidence of 90%, but μ_1 and μ_5 were not significantly different,

```
#Tukey multiple, df=114
 df= 114
alpha=0.1
1/sqrt(2)* qtukey(1-alpha, nm=6, df)
```

```
## [1] 2.623256
```

Results: Tukey multiple= 2.623 > Bonferroni multiple. Bonferroni procedure is more efficient.

HW 17.17

a) Estimate the contrast and get 95% CI

```
L = list(
L= c(1/2, 1/2, -1/2, -1/2,0,0))
(L1= contrast(Est.mean, L, adjust="none"))
```

```
##  contrast estimate     SE  df t.ratio p.value
##  L          -0.281 0.0394 114 -7.134  <.0001
```

```
confint(L1)
```

```
##  contrast estimate     SE  df lower.CL upper.CL
##  L          -0.281 0.0394 114   -0.359   -0.203
##
## Confidence level used: 0.95
```

b) Estimate the contrast and get 95% simultaneous CI.

```
#Bonferroni multiple,
g=7 #
qt(1- 0.1/(2*g), df)
```

```
## [1] 2.488175
```

```
# Scheffe multiple r-1=5, n_T-r=df=114
r= 6
sqrt((r-1)*qf(.90, (r-1) , df))
```

```
## [1] 3.080963
```

Results: This include contrasts so we can use Bonferroni or Scheffe procedure, and the Bonferroni procedure is more efficient by comparing the Bonferroni and Scheffe multiples.

```
L = list(
D1= c(1,-1,0,0,0,0),
D2= c(0,0,1,-1,0,0),
D3= c(0,0,0,0, 1,-1),
L1= c(1/2, 1/2, -1/2, -1/2,0,0),
L2= c(1/2, 1/2, 0, 0, -1/2, -1/2),
L3= c(1/4, 1/4, -1/2, -1/2,1/4,1/4),
L4= c(1/4, 1/4, 1/4, 1/4, -1/2, -1/2))
(Contrasts= contrast(Est.mean, L, adjust="Bonferroni"))
```

```
##  contrast estimate     SE  df t.ratio p.value
##  D1       -0.11700 0.0557 114 -2.102  0.2640
##  D2        0.09450 0.0557 114  1.698  0.6455
##  D3       -0.02650 0.0557 114 -0.476  1.0000
##  L1       -0.28075 0.0394 114 -7.134  <.0001
##  L2       -0.00625 0.0394 114 -0.159  1.0000
##  L3       -0.27763 0.0341 114 -8.147  <.0001
##  L4        0.13412 0.0341 114  3.936  0.0010
##
## P value adjustment: bonferroni method for 7 tests
```

```r
confint(Contrasts, level=0.90)
```

```
##  contrast estimate     SE  df lower.CL upper.CL
##  D1       -0.11700 0.0557 114  -0.2555   0.0215
##  D2        0.09450 0.0557 114  -0.0440   0.2330
##  D3       -0.02650 0.0557 114  -0.1650   0.1120
##  L1       -0.28075 0.0394 114  -0.3787  -0.1828
##  L2       -0.00625 0.0394 114  -0.1042   0.0917
##  L3       -0.27763 0.0341 114  -0.3624  -0.1928
##  L4        0.13412 0.0341 114   0.0493   0.2189
##
## Confidence level used: 0.9
## Conf-level adjustment: bonferroni method for 7 estimates
```

Results: Either by examining the adjusted p-value (or by checking of CI containing 0 or not), we can conclude that the contrasts L1, L3, L4 were significantly different from 0.

HW 17.25

With only 4 comparisons, Bonferroni is often better than Scheffe. We will try a few different sample sizes and compare with the required CI width. From the formula for the estimated variance, the pairwise comparison L1 and L2 have the same variance, bigger than the variance for L3 and L4. So we only need to check the width for L1 or L2. Assume we can use MSE= 0.031 as the estimated the true variance. If you assume the true variance differently, the sample size will be different.

$$var(L_1) = \sigma^2/n[1^2 + (-1)^2] = 2\sigma^2/n$$

```r
# Write a function to check CI for a given sample size per group
CIwidth <- function(n){
  newdf= 6*n-6
  g= 4
  B= qt(1- 0.05/(2*g), newdf)
  MSE=0.031
  SE.L=  sqrt(2*MSE/n)
  # CI width
  B*SE.L
}
```

Then we can try several cases until the precision is reached. The sample size is the input for the above function

```r
CIwidth(50)
```

```
## [1] 0.08849782
```

```r
CIwidth(60)
```

```
## [1] 0.08070243
```

```r
CIwidth(61)
```

```
## [1] 0.08003135
```

```r
CIwidth(62)
```

```
## [1] 0.07937674
```

Results: It shows that a sample n=62 per group is required if the precision for each of these comparisons is not exceed +/-0.08.