Chapter 7 of
Data Analysis for Experimental Design

Rich Gonzalez

August 23, 2009

1 Pairwise Tests

1.1 Bonferroni

The Bonferroni does not require a separate command because as described in the text one merely changes the Type I error criterion.

1.2 Tukey Test

The Tukey test is easy to run in R. It takes as argument the output from the aov() command (see R notes for Chapter 6). For the following I'm using the data with 5 groups from Chapter 6 (which is different than what appears in Chapter 7).

```
group <- c(rep(1,8), rep(2,8), rep(3,8), rep(4,8), rep(5,8))
data <- c(16, 18, 5, 12, 11, 12, 23, 19, 16, 7, 10, 4, 7, 23, 12, 13, 2,
  10, 9, 13, 11, 9, 13, 9, 5, 8, 8, 11, 1, 9, 5, 9, 7, 11, 12, 9, 14, 19, 16, 24)
group <- factor(group)
output <- aov(data~group)
summary(output)
TukeyHSD(output)
```

```
    diff          lwr        upr     p adj
2-1 -3.0 -10.06753766  4.0675377 0.7397203
```

```
3-1 -5.0 -12.06753766  2.0675377 0.2715178
4-1 -7.5 -14.56753766 -0.4324623 0.0329680
5-1 -0.5  -7.56753766  6.5675377 0.9995988
3-2 -2.0  -9.06753766  5.0675377 0.9246855
4-2 -4.5 -11.56753766  2.5675377 0.3729049
5-2  2.5  -4.56753766  9.5675377 0.8458204
4-3 -2.5  -9.56753766  4.5675377 0.8458204
5-3  4.5  -2.56753766 11.5675377 0.3729049
5-4  7.0  -0.06753766 14.0675377 0.0532717
```

2 Scheffe'

To my knowledge full use of Scheffe in R would require some programming. There is a plotting tool that uses Scheffe intervals in the CAR library.

I'll illustrate the use of R programming and its versatility. There are ways of working directly from the aov() output object rather than typing the actual numerical values, but I want to keep things relatively straightforward so the reader can see each step.

For the following I'm using the data with 5 groups from Chapter 6 (which is different than what appears in Chapter 7).

```
#you specify alpha level and degrees of freedom error
alpha.criterion <- .99
dferror <- 36

#compute group means
groupmeans <- by(data,group,mean)
k <- length(groupmeans)

#calculate F
F <- qf(alpha.criterion, k-1, dferror)

tprime <- sqrt( (k-1)*F)
tempdiff <- matrix(0,k*(k-1)/2,5)

#nested for loops to compute all possible pairwise differences; store in
#matrix tempdiff
counter <- 1
for (i in 1:(k-1))
  for (j in (i+1):k) {
tempdiff[counter,1] <- i
tempdiff[counter,2] <- j
```

```
tempdiff[counter,3] <- groupmeans[i]
tempdiff[counter,4] <- groupmeans[j]
tempdiff[counter,5] <- (groupmeans[i]-groupmeans[j])/se
counter <- counter+1
}
dimnames(tempdiff) <- list(NULL,c("group i", "group j", "mean i",
   "mean j", "normalized diff"))
print(tempdiff)
print(paste("test statistic to beat (absolute value): ",
  round(tprime,3),sep=""))
```

```
      group i group j mean i mean j normalized diff
 [1,]       1       2   14.5   11.5       3.8461538
 [2,]       1       3   14.5    9.5       6.4102564
 [3,]       1       4   14.5    7.0       9.6153846
 [4,]       1       5   14.5   14.0       0.6410256
 [5,]       2       3   11.5    9.5       2.5641026
 [6,]       2       4   11.5    7.0       5.7692308
 [7,]       2       5   11.5   14.0      -3.2051282
 [8,]       3       4    9.5    7.0       3.2051282
 [9,]       3       5    9.5   14.0      -5.7692308
[10,]       4       5    7.0   14.0      -8.9743590
[1] "test statistic to beat: 3.945"
```