test.R

spinoza

2021-10-01

```r
#We will use the same data as for handout 2 when illustrating contrasts and multiple comparisons
library("readxl")
#We will also need the R package for performing multiple comparisons
library("multcomp")
```

```
## Loading required package: mvtnorm
```

```
## Loading required package: survival
```

```
## Loading required package: TH.data
```

```
## Loading required package: MASS
```

```
##
## Attaching package: 'TH.data'
```

```
## The following object is masked from 'package:MASS':
##
##     geyser
```

```r
setwd("~/filetopia/gdrive/alex/college/grad_math_classes/stat589_fa2021/hw/hw4")
h4.data = read_excel("handout2data.xlsx")
str(h4.data)
```

```
## tibble [25 x 11] (S3: tbl_df/tbl/data.frame)
##  $ strength: num [1:25] 7 7 15 11 9 12 17 12 18 18 ...
##  $ percent : num [1:25] 15 15 15 15 15 20 20 20 20 20 ...
##  $ 20g     : num [1:25] 24 28 37 30 NA NA NA NA NA NA ...
##  $ 30g     : num [1:25] 37 44 31 35 NA NA NA NA NA NA ...
##  $ 40g     : num [1:25] 42 47 52 38 NA NA NA NA NA NA ...
##  $ life    : num [1:25] 17.6 18.9 16.3 17.4 20.1 21.6 16.9 15.3 18.6 17.1 ...
##  $ fluid   : num [1:25] 1 1 1 1 1 1 2 2 2 2 ...
##  $ rate    : num [1:25] 575 542 530 539 570 565 593 590 579 610 ...
##  $ rf power: num [1:25] 160 160 160 160 160 180 180 180 180 180 ...
##  $ brand   : chr [1:25] "acme" "acme" "acme" "acme" ...
##  $ wear    : num [1:25] 2.1 2.4 2.5 2.3 2.2 2 1.9 2.1 2.2 2.4 ...
```

```r
#Example 4.1 (refer back to Example 2.2 on comparing veneer brands)
brand = as.factor(na.omit(h4.data$brand))
wear = na.omit(h4.data$wear)

#Recall the ANOVA test for equal means. We want to investigate the relationship further.
aov.mod = aov(wear~brand)
summary(aov.mod)
```

```
##             Df Sum Sq Mean Sq F value  Pr(>F)
## brand        4 0.6170 0.15425   7.404 0.00168 **
```

```
## Residuals   15 0.3125 0.02083
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#Here we are using glht to estimate a constrast defined as a comparison between the first 3 groups and
con.test = glht(aov.mod, linfct = mcp( brand = c(2,2,2,-3,-3)))

#summary is used to display the test results, confint is used to display the interval estimate
summary(con.test)
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: User-defined Contrasts
##
##
## Fit: aov(formula = wear ~ brand)
##
## Linear Hypotheses:
##        Estimate Std. Error t value Pr(>|t|)
## 1 == 0  -1.4250     0.3953  -3.605   0.0026 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

```
confint(con.test)
```

```
##
##   Simultaneous Confidence Intervals
##
## Multiple Comparisons of Means: User-defined Contrasts
##
##
## Fit: aov(formula = wear ~ brand)
##
## Quantile = 2.1314
## 95% family-wise confidence level
##
##
## Linear Hypotheses:
##        Estimate lwr     upr
## 1 == 0 -1.4250  -2.2675 -0.5825
```

```
#The code below is used to define a set of orthogonal contrasts
contrasts(brand) = cbind( c(2,2,2,-3,-3),
                          c(1,1,-2,0,0),
                          c(1,-1,0,0,0),
                          c(0,0,0,1,-1))

#We can check the defined contrasts for brand. Read down each column for the corresponding contrast.
brand
```

```
##  [1] acme  acme  acme  acme  ajax  ajax  ajax  ajax  champ champ champ champ
## [13] tuffy tuffy tuffy tuffy xtra  xtra  xtra  xtra
## attr(,"contrasts")
##       [,1] [,2] [,3] [,4]
## acme     2    1    1    0
```

```
## ajax     2    1   -1    0
## champ    2   -2    0    0
## tuffy   -3    0    0    1
## xtra    -3    0    0   -1
## Levels: acme ajax champ tuffy xtra
```

```
#We will re-fit the ANOVA model, now with our own contrasts defined as above
contr.mod = aov(wear~brand)

#We can get a decomposition of sum squares into specific effects
#The command split is used to specify the decompositon.
summary(contr.mod,split = list(brand=list("us-f"=1,"a-c"=2,"ac-aj"=3,"t-x"=4)))
```

```
##                 Df Sum Sq Mean Sq F value  Pr(>F)
## brand            4 0.6170 0.15425   7.404 0.00168 **
##   brand: us-f    1 0.2707 0.27075  12.996 0.00260 **
##   brand: a-c     1 0.0937 0.09375   4.500 0.05097 .
##   brand: ac-aj   1 0.1512 0.15125   7.260 0.01664 *
##   brand: t-x     1 0.1013 0.10125   4.860 0.04352 *
## Residuals       15 0.3125 0.02083
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#We can also use glht to estimate the specific contrast effects
contr = glht(aov.mod, linfct = mcp( brand = rbind( c(2,2,2,-3,-3),
                                                   c(1,1,-2,0,0),
                                                   c(1,-1,0,0,0),
                                                   c(0,0,0,1,-1)) ))

#adjusted none indicates that we are not adjusting for multiple tests
#univariate_calpha indicates that we are not adjusting for multiple intervals (c in calpha stands for c
summary(contr,test = adjusted("none"))
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: User-defined Contrasts
##
##
## Fit: aov(formula = wear ~ brand)
##
## Linear Hypotheses:
##        Estimate Std. Error t value Pr(>|t|)
## 1 == 0  -1.4250     0.3953  -3.605   0.0026 **
## 2 == 0  -0.3750     0.1768  -2.121   0.0510 .
## 3 == 0   0.2750     0.1021   2.694   0.0166 *
## 4 == 0   0.2250     0.1021   2.205   0.0435 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- none method)
```

```
confint(contr,calpha = univariate_calpha())
```

```
##
##   Simultaneous Confidence Intervals
##
```

```
## Multiple Comparisons of Means: User-defined Contrasts
##
##
## Fit: aov(formula = wear ~ brand)
##
## Quantile = 2.1314
## 95% confidence level
##
##
## Linear Hypotheses:
##        Estimate  lwr       upr
## 1 == 0 -1.425000 -2.267529 -0.582471
## 2 == 0 -0.375000 -0.751791  0.001791
## 3 == 0  0.275000  0.057460  0.492540
## 4 == 0  0.225000  0.007460  0.442540
```

```r
#Using the same example, let's investigate some pairwise mulitple comparisons.
#Remember that Tukey is used to define pairwise comparisons, as the Tukey method is best suited for suc
comparisons.mod = glht(aov.mod, linfct = mcp( brand = "Tukey"))

#Fisher LSD tests, error probability controlled for each comparison
summary(comparisons.mod,test=univariate())
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: aov(formula = wear ~ brand)
##
## Linear Hypotheses:
##                     Estimate Std. Error t value Pr(>|t|)
## ajax - acme == 0   -2.750e-01  1.021e-01  -2.694  0.01664 *
## champ - acme == 0   5.000e-02  1.021e-01   0.490  0.63129
## tuffy - acme == 0   2.750e-01  1.021e-01   2.694  0.01664 *
## xtra - acme == 0    5.000e-02  1.021e-01   0.490  0.63129
## champ - ajax == 0   3.250e-01  1.021e-01   3.184  0.00616 **
## tuffy - ajax == 0   5.500e-01  1.021e-01   5.389 7.53e-05 ***
## xtra - ajax == 0    3.250e-01  1.021e-01   3.184  0.00616 **
## tuffy - champ == 0  2.250e-01  1.021e-01   2.205  0.04352 *
## xtra - champ == 0  -1.388e-16  1.021e-01   0.000  1.00000
## xtra - tuffy == 0  -2.250e-01  1.021e-01  -2.205  0.04352 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Univariate p values reported)
```

```r
#Tukey pairwise tests, error probability controlled across all comparisons
summary(comparisons.mod)
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
```

```
##
## Fit: aov(formula = wear ~ brand)
##
## Linear Hypotheses:
##                      Estimate Std. Error t value Pr(>|t|)
## ajax - acme == 0   -2.750e-01  1.021e-01  -2.694   0.1021
## champ - acme == 0   5.000e-02  1.021e-01   0.490   0.9871
## tuffy - acme == 0   2.750e-01  1.021e-01   2.694   0.1021
## xtra - acme == 0    5.000e-02  1.021e-01   0.490   0.9871
## champ - ajax == 0   3.250e-01  1.021e-01   3.184   0.0417 *
## tuffy - ajax == 0   5.500e-01  1.021e-01   5.389   <0.001 ***
## xtra - ajax == 0    3.250e-01  1.021e-01   3.184   0.0417 *
## tuffy - champ == 0  2.250e-01  1.021e-01   2.205   0.2305
## xtra - champ == 0  -1.388e-16  1.021e-01   0.000   1.0000
## xtra - tuffy == 0  -2.250e-01  1.021e-01  -2.205   0.2304
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

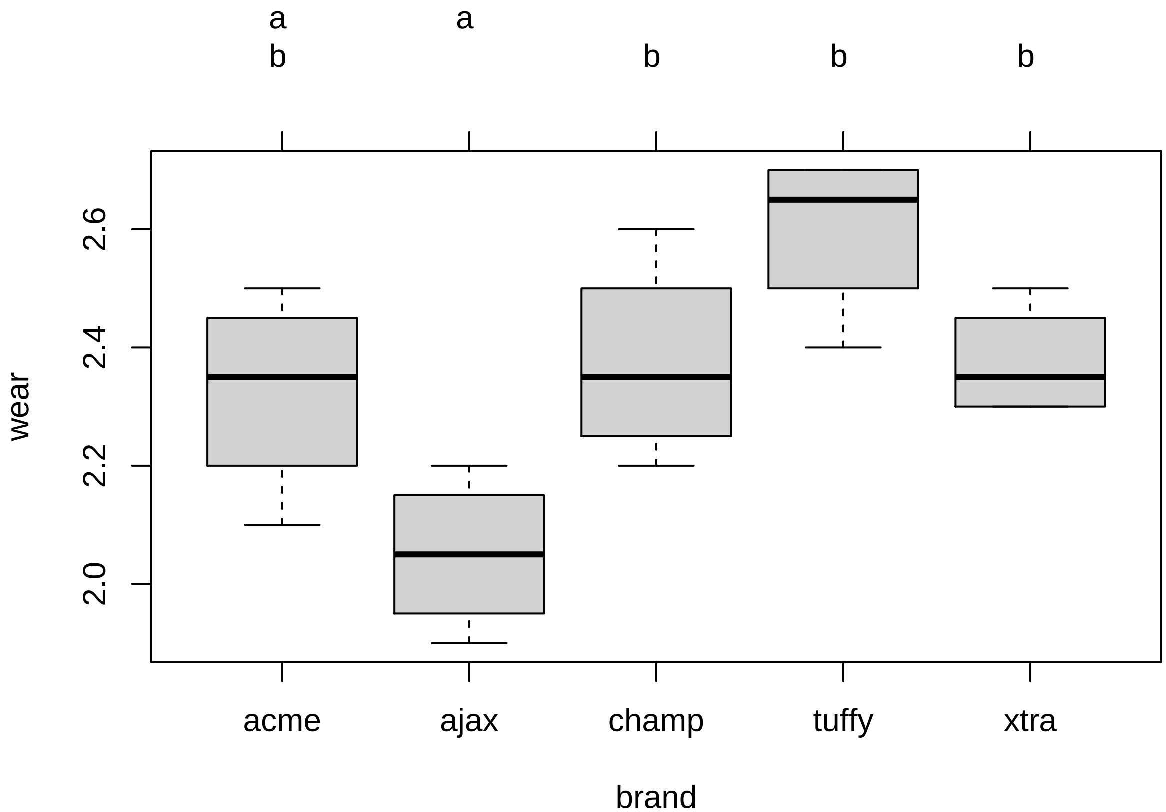
```
#We can summarize the results for Fisher and Tukey with a compact letter display
cld(summary(comparisons.mod,test=univariate()))
```

```
##  acme  ajax champ tuffy  xtra
##   "b"   "a"   "b"   "c"   "b"
```

```
cld(summary(comparisons.mod))
```

```
##  acme  ajax champ tuffy  xtra
##  "ab"   "a"   "b"   "b"   "b"
```

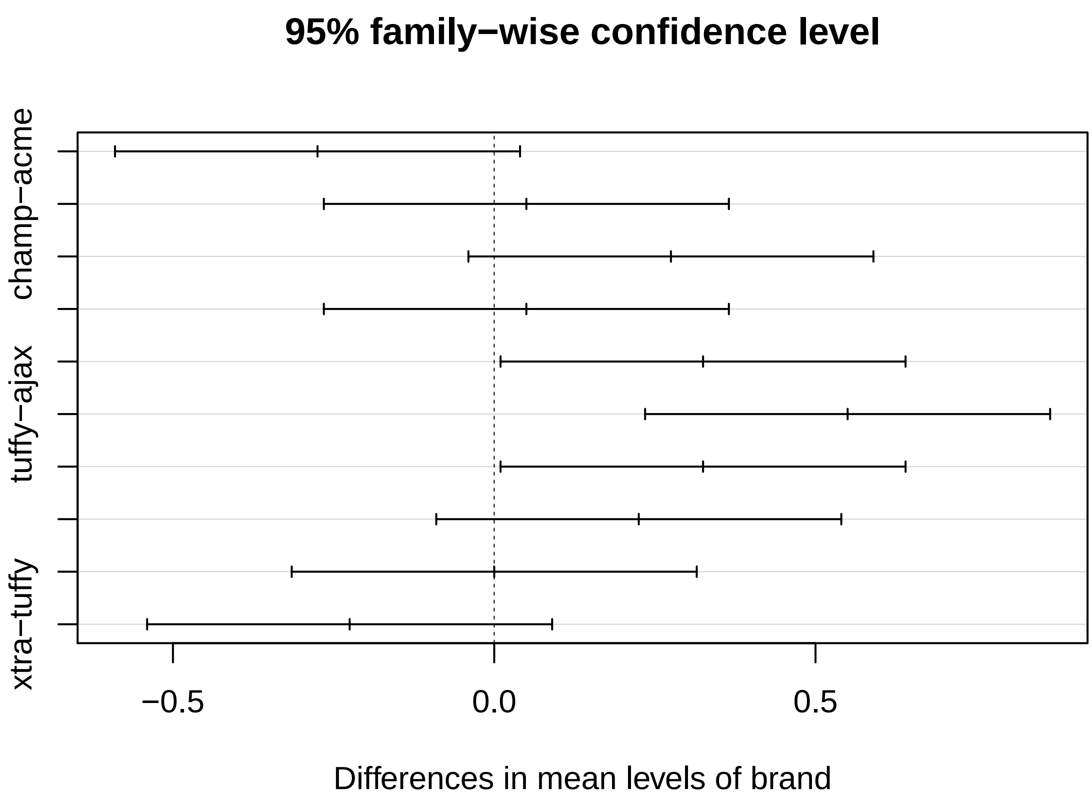
```
#We can also display the data as a boxplot, with the Tukey groupings
plot(cld(summary(comparisons.mod)))
```




```
#We could also use the TukeyHSD command with the model defined through aov (rather than glht)
TukeyHSD(aov.mod)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = wear ~ brand)
##
## $brand
##                      diff          lwr        upr     p adj
## ajax-acme   -2.750000e-01 -0.590159973 0.04015997 0.1021412
## champ-acme   5.000000e-02 -0.265159973 0.36515997 0.9871310
## tuffy-acme   2.750000e-01 -0.040159973 0.59015997 0.1021412
## xtra-acme    5.000000e-02 -0.265159973 0.36515997 0.9871310
## champ-ajax   3.250000e-01  0.009840027 0.64015997 0.0417456
## tuffy-ajax   5.500000e-01  0.234840027 0.86515997 0.0006152
## xtra-ajax    3.250000e-01  0.009840027 0.64015997 0.0417456
## tuffy-champ  2.250000e-01 -0.090159973 0.54015997 0.2304525
## xtra-champ  -4.440892e-16 -0.315159973 0.31515997 1.0000000
## xtra-tuffy  -2.250000e-01 -0.540159973 0.09015997 0.2304525
```

```
#A plot of the interval estimates with the Tukey adjustment
plot(TukeyHSD(aov.mod))
```




```r
#We can use the Tukey Q distribution to check the above calculations
#For example, t = 2.694, p-adj = .1021
t0 = 2.694
a = 5
n = 4
df = a*(n-1)
ptukey(sqrt(2)*t0,a,df,lower.tail=FALSE)
```

```
## [1] 0.1022201
```

```r
#Here are the calculations for the least significant differences (margin of errors) for Fisher and Tuke
mse = .02083

m_lsd = qt(.025,df,lower.tail = FALSE)*sqrt(2*mse/n)
m_lsd
```

```
## [1] 0.2175228
```

```r
m_tukey = qtukey(.05,a,df,lower.tail = FALSE)*sqrt(mse/n)
m_tukey
```

```
## [1] 0.3151348
```

```r
#Here is the calculation for Tukey comparison error rate
2*pt(qtukey(.05,a,df,lower.tail = FALSE)/sqrt(2),df,lower.tail = FALSE)
```

```
## [1] 0.007499933
```

```r
#Here is the calculation for Fisher family error rate
ptukey(qt(.025,df,lower.tail=FALSE)*sqrt(2),a,df,lower.tail = FALSE)
```

```
## [1] 0.2575972
```