Answer to assignment4

November 07, 2015

Preparing data

```
setwd("~/Documents/R/DAE/Lab3/Lab3data")
bio <- read.csv("bio.csv")
str(bio)
```

```
## 'data.frame':    50 obs. of  4 variables:
##  $ Groupa: num  1.2 1.2 1.2 1.2 1.2 ...
##  $ Groupb: num  2.3 2.3 2.3 2.3 2.3 ...
##  $ Groupc: num  1.2 1.2 1.2 1.2 1.2 ...
##  $ Groupd: num  -1.548 -1.331 -0.205 1.11 1.11 ...
```

Power analysis

1. Calculate effect size using Cohen's d.

Cohen's d is defined as the difference between two means divided by a standard deviation for the data, i.e.

$$d = \frac{\bar{x}_1 - \bar{x}_2}{s}.$$

Jacob Cohen defined s, the pooled standard deviation, as (for two independent samples):

$$s = \sqrt{\frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}}$$

where s_1 and s_2 are the variances of group1 and group2, respectively.

The following code calculates the Cohen's of `Groupa` and `Groupb`.

```
n1 <- length(bio$Groupa)
n2 <- length(bio$Groupb)
s <- sqrt(((n1 -1) * var(bio$Groupa) + (n2 -1) * var(bio$Groupb)) / (n1 + n2 - 2))
(Cd <- (mean(bio$Groupa) - mean(bio$Groupb)) / s)
```

```
## [1] -0.1152128
```

Alternatively, you can call the existing function `cohen.d` from **effsize** package. A benfit of function `cohen.d` is that the confidence interval of Cohen's d is also provided.

```
install.packages("effsize", repos = "https://mirrors.tuna.tsinghua.edu.cn/CRAN/")
```

```
##
## The downloaded binary packages are in
##  /var/folders/dd/vn0g7h013fj78l0g4nlyqjph0000gn/T//Rtmp3Tv53P/downloaded_packages
```

```r
library(effsize)
cohen.d(bio$Groupa, bio$Groupb)
```

```
##
## Cohen's d
##
## d estimate: -0.1152128 (negligible)
## 95 percent confidence interval:
##        inf        sup
## -0.5164751  0.2860494
```

2. Calculate the sample size

After knowing the Cohen's d between `Groupa` and `Groupb` is -0.1152128, we call the `pwr.t.test` function from `pwr` package to calculate the sample size needed for detecting the difference between `Groupa` and `Groupb` at a power of 0.2, 0.5, 0.8 each and a significance level of 0.05 using t-tests.

```r
library(pwr)
pwr.t.test(d = Cd, sig.level = 0.05, power = 0.2, type = "two.sample",
           alternative = "two.sided")
```

```
##
##      Two-sample t test power calculation
##
##               n = 188.1372
##               d = 0.1152128
##       sig.level = 0.05
##           power = 0.2
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

```r
pwr.t.test(d = Cd, sig.level = 0.05, power = 0.5, type = "two.sample",
           alternative = "two.sided")
```

```
##
##      Two-sample t test power calculation
##
##               n = 579.6908
##               d = 0.1152128
##       sig.level = 0.05
##           power = 0.5
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

```r
pwr.t.test(d = Cd, sig.level = 0.05, power = 0.8, type = "two.sample",
           alternative = "two.sided")
```

```
##
##      Two-sample t test power calculation
```

```
##
##               n = 1183.553
##               d = 0.1152128
##       sig.level = 0.05
##           power = 0.8
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

Tukey test

To conduct a Tukey test on all four groups, we need to stack the `bio` data.

```
sbio <- stack(bio)
names(sbio) <- c("Values", "Groups")
str(sbio)
```

```
## 'data.frame':    200 obs. of  2 variables:
##  $ Values: num  1.2 1.2 1.2 1.2 1.2 ...
##  $ Groups: Factor w/ 4 levels "Groupa","Groupb",..: 1 1 1 1 1 1 1 1 1 1 ...
```

We can see there are 4 levels (i.e. `Groupa`, `Groupb`, `Groupc` and `Groupd`) for factor `Groups`.

1. Fit an ANOVA model

```
sbio.aov <- aov(Values ~ Groups, data = sbio)
summary(sbio.aov)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## Groups        3    0.5  0.1672   0.175  0.913
## Residuals   176  168.5  0.9573
## 20 observations deleted due to missingness
```

From the output, we can see that `Groups` is not significant, which means that we can't draw the conclusion that there exists difference on the mean between the four groups. So it's not necessary for us to further investigate the particular groups where there are differences using Tukey's multiple comparisons. However, as an exercise, we can still give it a try.

As there is only one independant factor `Groups`, it's an one-way analysis of variance. Hence, function `TukeyHSD` can be used to theoretically conduct the Tukey test. Typing `?TukeyHSD` to see the help document of function `TukeyHSD`, we know the function `TukeyHSD` "would only apply exactly to balanced designs where there are the same number of observations made at each level of the factor. This function incorporates an adjustment for sample size that produces sensible intervals for mildly unbalanced designs." As there are 20 NAs for `Groupd`, a Tukey test is more appropriate.

2. Conduct the Tukey test

```
library(multcomp)
```

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

```
sbio.mc <- glht(sbio.aov, linfct = mcp(Groups = "Tukey"))
summary(sbio.mc)
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: aov(formula = Values ~ Groups, data = sbio)
##
## Linear Hypotheses:
##                        Estimate Std. Error t value Pr(>|t|)
## Groupb - Groupa == 0  0.12032    0.19569   0.615    0.927
## Groupc - Groupa == 0 -0.00400    0.19569  -0.020    1.000
## Groupd - Groupa == 0  0.04877    0.22596   0.216    0.996
## Groupc - Groupb == 0 -0.12431    0.19569  -0.635    0.920
## Groupd - Groupb == 0 -0.07155    0.22596  -0.317    0.989
## Groupd - Groupc == 0  0.05277    0.22596   0.234    0.995
## (Adjusted p values reported -- single-step method)
```

We can see that no significant p-value is produced for all pair-wise comparisions of the `Groups` levels. Actually, the `TukeyHSD` test gives the same results.

```
TukeyHSD(sbio.aov)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Values ~ Groups, data = sbio)
##
## $Groups
##                        diff        lwr       upr     p adj
## Groupb-Groupa  0.12031493 -0.3872496 0.6278795 0.9272505
## Groupc-Groupa -0.00400000 -0.5115646 0.5035646 0.9999969
## Groupd-Groupa  0.04876855 -0.5373165 0.6348536 0.9964374
## Groupc-Groupb -0.12431493 -0.6318795 0.3832496 0.9205027
## Groupd-Groupb -0.07154638 -0.6576314 0.5145387 0.9889775
## Groupd-Groupc  0.05276855 -0.5333165 0.6388536 0.9955005
```

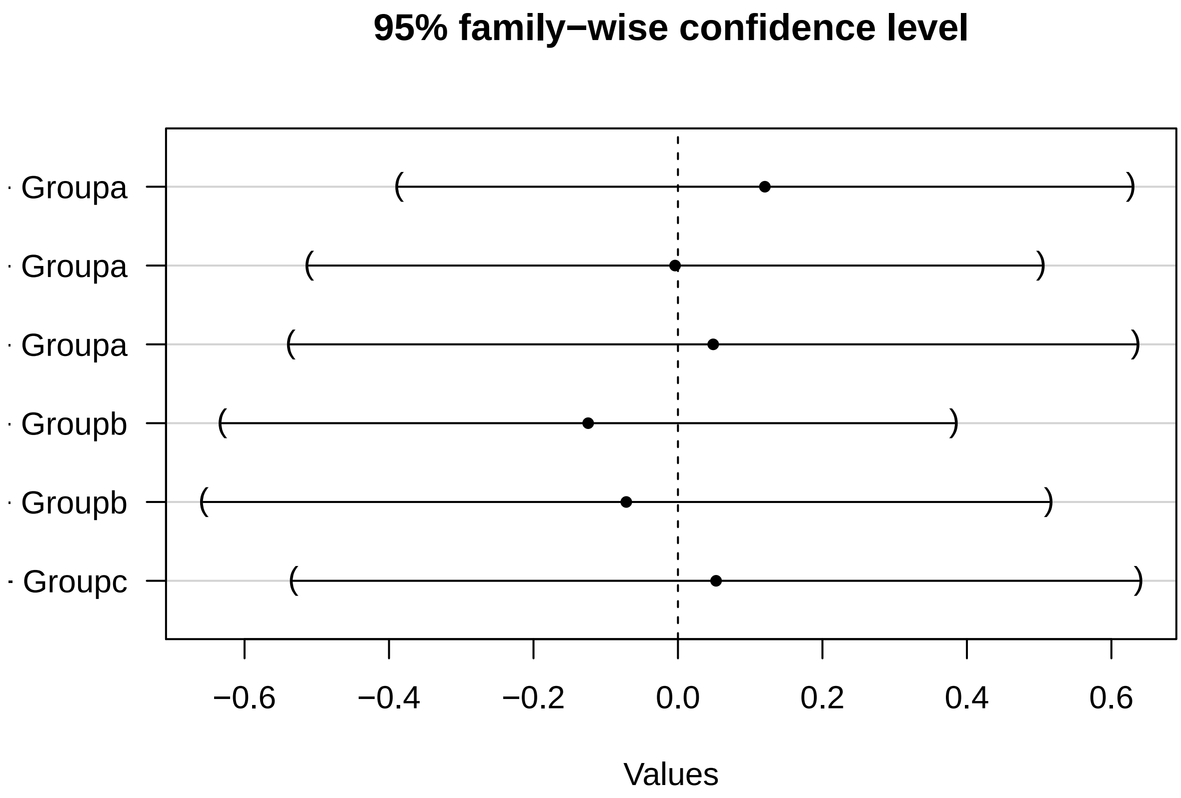
3. Calculate the two-sided, 95% confindence interval and plot the interval

```
sbio.ci <- confint(sbio.mc, level = 0.95)
sbio.ci
```

```
##
##   Simultaneous Confidence Intervals
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
```

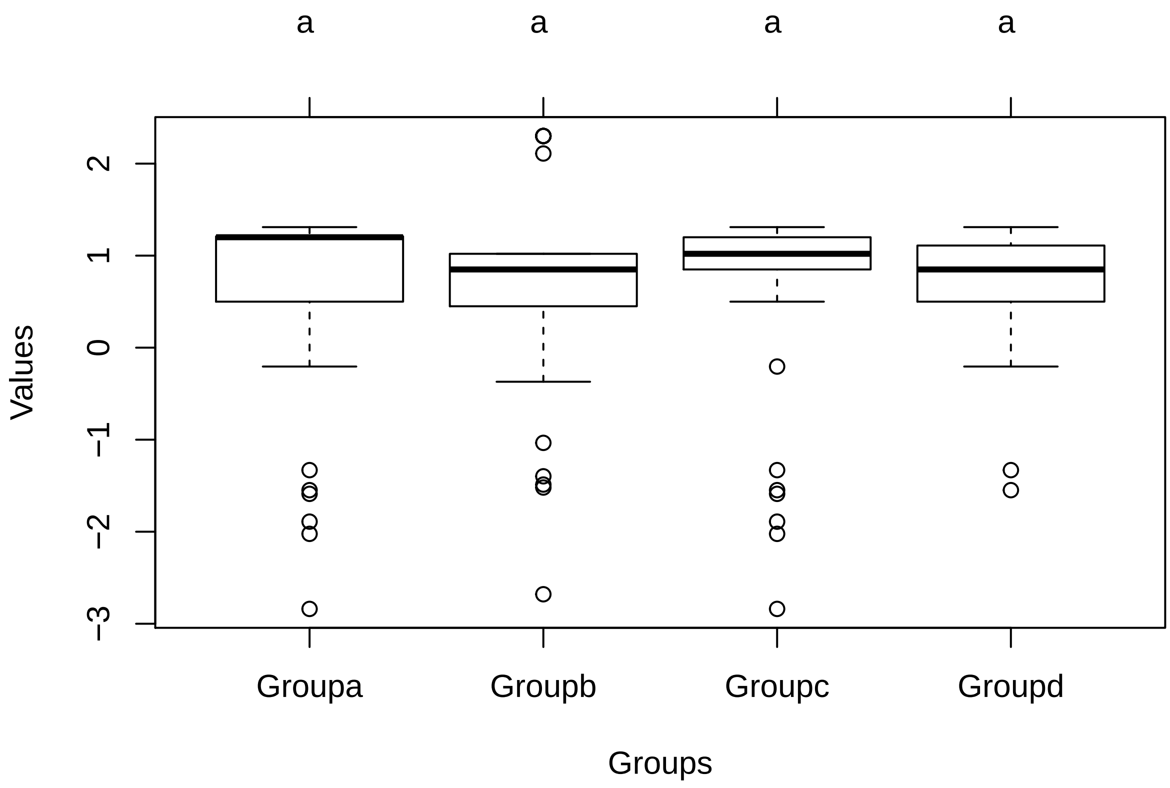
```
## Fit: aov(formula = Values ~ Groups, data = sbio)
##
## Quantile = 2.5906
## 95% family-wise confidence level
##
##
## Linear Hypotheses:
##                        Estimate lwr      upr
## Groupb - Groupa == 0  0.12031 -0.38662  0.62725
## Groupc - Groupa == 0 -0.00400 -0.51094  0.50294
## Groupd - Groupa == 0  0.04877 -0.53660  0.63413
## Groupc - Groupb == 0 -0.12431 -0.63125  0.38262
## Groupd - Groupb == 0 -0.07155 -0.65691  0.51382
## Groupd - Groupc == 0  0.05277 -0.53260  0.63813
```

```
plot(sbio.ci, xlab = "Values")
```



4. Produce a compact letter display of all pair-wise comparisons

```
sbio.cld <- cld(sbio.mc)
plot(sbio.cld)
```


a
a
a
a
2
1
0
−1
−2
−3
Values
Groupa
Groupb
Groupc
Groupd
Groups