Lab 9: Basic Code For All Data Sets

Outline

- Make 4 datasets of different antler measurements (mass, basal circumference, main beam length, inside spread)
- ANOVA omnibus F-test
- Pairwise t-tests
- TukeyHSD tests
- Plot effect sizes from TukeyHSD

Preliminaries: Make datasets

Datset 1) Antler mass data

```
mass<-c(472.5114,722.7774,548.6796,464.3168,577.2037,406.7964,555.8052,657.5226,734.7416,919.2983,665.6099
,570.3627,429.5572,520.9726,346.0116,639.3096,544.8374,657.6512,802.2045,443.339,545.9901,791.3983,884.580
3,763.2926,782.1005,840.2083,417.7051,735.8959,759.7498,877.0629)
n <- length(mass)/3
diet <- c(rep("Hi.Hi",n),rep("Hi.Lo",n),rep("Lo.Hi",n))
df.mass <- data.frame(antler.mass = mass,
                      diet = diet)
rm(list = c("mass", "diet","n"))
```

Datset 2) Antler mass circumference data

```
circum <- c(94.85394,83.21354,64.94166,96.9141,77.03692,113.2507,90.95746,95.83833,82.66041,97.46359,91.61
469,96.4177,100.5873,99.3556,94.23969,57.94891,134.9324,95.01268,128.2977,129.307,70.39667,133.7504,121.07
67,105.3898,86.81835,106.1906,136.0613,74.95854,112.6994,110.5714)
n <- length(circum)/3
diet <- c(rep("Hi.Hi",n),rep("Hi.Lo",n),rep("Lo.Hi",n))
df.circum <- data.frame(antler.circum = circum,
                        diet = diet)
rm(list = c("circum", "diet","n"))
```

Datset 3) Antler beam length data

```
beam <- c(335.6464,410.2204,425.6838,355.1535,346.7697,399.8559,425.4827,447.6792,397.116,428.6749,406.034
,380.342,437.5513,444.8916,437.9827,391.446,425.6301,404.2163,461.0517,568.163,457.6777,459.0168,270.5814,
440.9724,412.0543,428.0725,247.2619,386.4118,423.5378,505.5221)
n <- length(beam)/3
diet <- c(rep("Hi.Hi",n),rep("Hi.Lo",n),rep("Lo.Hi",n))
df.beam <- data.frame(antler.beam = beam,
                      diet = diet)
rm(list = c("beam", "diet","n"))
```

Datset 4) Antler spread data

```
spread <- c(271.4356,291.3746,346.3026,345.6056,234.1441,339.7039,291.5156,275.9856,223.7703,316.9263,375.
1109,384.1469,396.6894,272.4158,356.8185,393.6393,412.3185,425.7551,390.7663,381.4302,341.7604,350.6951,30
6.0166,354.8825,312.0478,459.7191,415.1186,321.8994,281.539,372.0439)
n <- length(spread)/3
diet <- c(rep("Hi.Hi",n),rep("Hi.Lo",n),rep("Lo.Hi",n))
df.spread <- data.frame(antler.spread = spread,
```

```
categories <- levels(df.mass$diet)

plot3means(means = mean.mass,
           se = se.mass,
           categories = categories,
           y.axis.label = "Anterl mass (g)"
           )
```

Plot raw data
```
boxplot(antler.mass ~ diet,df.mass)
```

ANOVA omnibus F-test

Build models

```
model.null <- lm(antler.mass ~ 1, data = df.mass)
model.alt  <- lm(antler.mass ~ diet, data = df.mass)
```

Compare models to get F statistic and p-values

```
anova(model.null, model.alt)

## Analysis of Variance Table
##
## Model 1: antler.mass ~ 1
## Model 2: antler.mass ~ diet
##   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
## 1     29 747262
## 2     27 575719  2    171543 4.0225 0.02958 *
```

Save model output

```
mass.anova.table <- anova(model.null, model.alt)
```

Pairwise t-test

summary(df.mass)

Make table of pairwise t-test

```
pairwise.t.test(x = df.mass$antler.mass,
                g = df.mass$diet,
                p.adjust.method = "none")

##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  df.mass$antler.mass and df.mass$diet
##
##       Hi.Hi Hi.Lo
## Hi.Lo 0.506 -
## Lo.Hi 0.050 0.011
##
## P value adjustment method: none
```

```
sd.circum <- sd(df.circum$antler.circum)
n.circum <- length(mean(df.circum$antler.circum))
```

Standard errors
```
sd.circum/sqrt(n.circum)
```

```
## [1] 20.62602
```

Summary stats by diet treatment

Use tapply to get means and standard deviations by group.

Calcualte means with tapply
```
mean.circum <- tapply(df.circum$antler.circum,
                      df.circum$diet,
                      FUN = mean)
```

Calcualte sd with tapply
```
sd.circum <- tapply(df.circum$antler.circum,
                    df.circum$diet,
                    FUN = sd)
```

Calcualte sample size with tapply
```
n.circum <- tapply(df.circum$antler.circum,
                   df.circum$diet,
                   FUN = length)
```

Calcualte SD
```
se.circum <- sd.circum/sqrt(n.circum)
```

Plot the means

This requires the plot3means function from https://rpubs.com/brouwern/plot3means

```
categories <- levels(df.circum$diet)
```

```
plot3means(means = mean.circum,
           se = se.circum,
           categories = categories,
           y.axis.label = "Anterl circum (g)"
           )
```

Plot raw data
```
boxplot(antler.circum ~ diet,df.circum)
```

```
TukeyHSD(circum.aov)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = antler.circum ~ diet, data = df.circum)
##
## $diet
##                  diff        lwr       upr     p adj
## Hi.Lo-Hi.Hi 13.058302  -9.197247 35.31385 0.3280747
## Lo.Hi-Hi.Hi 16.078251  -6.177298 38.33380 0.1915525
## Lo.Hi-Hi.Lo  3.019949 -19.235600 25.27550 0.9396505
```

Plot effect sizes from TukeyHSD

Save output of TukeyHSD to an object

```
circum.tuk <- TukeyHSD(circum.aov)
```

Use plotTukeyHSD from https://rpubs.com/brouwern/plotTukeyHSD

```
plotTukeysHSD(circum.tuk)
```

Antler main beam data

Data analysis

Summary stats for all data

Basic summary stats
```
mean(df.beam$antler.beam)
```

```
## [1] 412.0233
```

```
var(mean(df.beam$antler.beam))
```

```
## [1] NA
```

```
sd.beam <- sd(df.beam$antler.beam)
n.beam <- length(mean(df.beam$antler.beam))
```

Standard errors
```
sd.beam/sqrt(n.beam)
```

```
## [1] 61.19076
```

Summary stats by diet treatment

Use tapply to get means and standard deviations by group.

```
anova(model.null, model.alt)
```

```
## Analysis of Variance Table
##
## Model 1: antler.beam ~ 1
## Model 2: antler.beam ~ diet
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     29 108585
## 2     27  99981  2    8603.7 1.1617 0.3281
```

Save model output

```
beam.anova.table <- anova(model.null, model.alt)
```

Pairwise t-test

summary(df.beam)

Make table of pairwise t-test

```
pairwise.t.test(x = df.beam$antler.beam,
                g = df.beam$diet,
                p.adjust.method = "none")
```

```
##
##   Pairwise comparisons using t tests with pooled SD
##
## data:  df.beam$antler.beam and df.beam$diet
##
##       Hi.Hi Hi.Lo
## Hi.Lo 0.17  -
## Lo.Hi 0.83  0.24
##
## P value adjustment method: none
```

Save output

```
beam.pairwise.p <- pairwise.t.test(x = df.beam$antler.beam,
                g = df.beam$diet,
                p.adjust.method = "none")
```

TukeyHSD p-values

Fit model with aov() instead of lm()

```
beam.aov <- aov(antler.beam ~ diet, data = df.beam)
```

Calcaulte corrected p valeus and effect sizes

TukeyHSD(beam.aov)

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = antler.beam ~ diet, data = df.beam)
##
## $diet
##                  diff       lwr       upr      p adj
## Hi.Lo-Hi.Hi  38.50262 -28.97224 105.97748 0.3476276
```

Calcualte sd with tapply

```
sd.spread <- tapply(df.spread$antler.spread,
                    df.spread$diet,
                    FUN = sd)
```

Calcualte sample size with tapply

```
n.spread <- tapply(df.spread$antler.spread,
                   df.spread$diet,
                   FUN = length)
```

Calcualte SD

```
se.spread <- sd.spread/sqrt(n.spread)
```

Plot the means

This requires the plot3means function from https://rpubs.com/brouwern/plot3means

```
par(mfrow = c(1,1))
categories <- levels(df.spread$diet)
plot3means(means = mean.spread,
           se = se.spread,
           categories = categories,
           x.axis.label = "Diet treatment",
           y.axis.label = "Anter spread (mm)",axis.adjust = 0.01
           )
```

Plot raw data

```
boxplot(antler.spread ~ diet,df.spread)
```

ANOVA omnibus F-test

Build models

```
model.null <- lm(antler.spread ~ 1, data = df.spread)
model.alt  <- lm(antler.spread ~ diet, data = df.spread)
```

Compare models to get F statistic and p-values

```
anova(model.null, model.alt)
```

```
## Analysis of Variance Table
##
## Model 1: antler.spread ~ 1
## Model 2: antler.spread ~ diet
##   Res.Df   RSS Df Sum of Sq      F   Pr(>F)
## 1     29 96798
## 2     27 58919  2     37879 8.6793 0.001228 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Use plotTukeyHSD from https://rpubs.com/brouwern/plotTukeyHSD

```
plotTukeysHSD(spread.tuk)
```