R Demonstration – ANCOVA

Objective: The purpose of this week's session is to demonstrate how to perform an analysis of covariance (ANCOVA) in R, and how to plot the regression lines for each level of the factor in the ANCOVA.

Part I. Performing the ANCOVA

NOTE: This part of the exercise assumes that you have downloaded the dataset that reports the membrane potential (grouped by cation system) and its covariate (the log of the action ratio) and saved it in your `PCB6466` folder as a tab-delimited text file named *`potentials.txt`*. You also need to download the *`ANCOVA.R`* script and save it in your `PCB6466` folder.

After starting R, change the directory to your `PCB6466` folder and open the *`ANCOVA.R`* script. The first 3 lines of the script read, attach the dataset, and center the predictor:

```
## read and attach the ANCOVA data
ancova_data <- read.table("potentials.txt", header=T)
attach(ancova_data)
X <- as.numeric(scale(X))
```

Next, we use the familiar *`lm`* function to create a linear model. This time we relate the response variable (`Y`, the membrane potential) to both a continuous variable (`X`, the log of the action ratio) and a categorical variable (`Group`, the cation system). Since we are interested in the additive effects of the factor and its covariate and the potential interaction between them, we will use the `*` operator in our linear model specification (i.e. `Y ~ X * Group`). Then we use the *`anova`* function to produce summary statistics, including an ANOVA table, for our model:

```
model1 <- lm(Y ~ X * Group)
anova(model1)
```

The call to the *`anova`* function produces the following output:

```
Analysis of Variance Table

Response: Y
          Df Sum Sq Mean Sq   F value    Pr(>F)
X          1 4197.0  4197.0 1152.7173 4.431e-14 ***
Group      3 1768.6   589.5  161.9151 1.521e-10 ***
X:Group    3    0.8     0.3    0.0729    0.9735
Residuals 13   47.3     3.6
---
```

Notice that both the covariate (`X`) and the factor (`Group`) are highly significant ($P < 0.001$), but that their interaction term (`X:Group`) is not significant ($P > 0.05$).

As was discussed in the lecture, it is very important to be aware of the fact that the order in which you specify the terms in the linear model affects the ANCOVA results. The proper order is with the covariate listed first, as we did in `model1` above. To see how order affects the outcome, we can create another model in which we specify the factor before the covariate (i.e. `Y ~ Group * X`):

```
model2 <- lm(Y ~ Group * X)
anova(model2)
```

This model executes properly in R, but notice that the output is different from that above:

```
Analysis of Variance Table

Response: Y
          Df Sum Sq Mean Sq   F value    Pr(>F)
Group      3  390.7   130.2   35.7647 1.515e-06 ***
X          1 5574.9  5574.9 1531.1685 7.118e-15 ***
Group:X    3    0.8     0.3    0.0729    0.9735
Residuals 13   47.3     3.6
---
```

As shown by all the results highlighted in red in the table above, switching the order of the covariate and the factor has changed the sum of squares, mean squares, F-ratio, and P-values for both of these model components. While this did not affect our conclusions in this particular example, it certainly could in other situations. **For ANCOVA, therefore, always specify the covariate first in the linear model!**

Based on our results from `model1` above, we concluded that the interaction term was not significant. We can thus simplify our ANCOVA by using a model with only additive effects (i.e. `Y ~ X + Group`):

```
model3 <- lm(Y ~ X + Group)
anova(model3)
```

For the additive model, we get the following results:

```
Analysis of Variance Table

Response: Y
          Df Sum Sq Mean Sq F value    Pr(>F)
X          1 4197.0  4197.0 1395.25 < 2.2e-16 ***
Group      3 1768.6   589.5  195.98 8.005e-13 ***
Residuals 16   48.1     3.0
---
```

Our covariate (`X`) and the factor (`Group`) are still highly significant ($P < 0.001$), so now it is time to plot the results of our ANCOVA.

Part II. Plotting the ANCOVA results

As discussed in the lecture and on pages 333-335 of the Gotelli & Ellison (2004) text, the proper way to plot the results of an ANCOVA is with the covariate on the x-axis and the response variable on the y-axis. Each replicate observation is plotted as a point with a different symbol used for each level of the factor. Finally, the regression lines fitted for the different levels of the factor are plotted using different line colors or symbols.

To plot our ANCOVA results in R, we need to obtain the intercept and slope coefficients for the regression line of each treatment level. Since we concluded that the interaction term in our model was not significant, we know that all 4 of our regression lines will have the same slope (though they will all have different y-intercept values). We will use the *summary.lm* function to obtain these values for our model:

```
> summary.lm(model3)

Call:
lm(formula = Y ~ X + Group)

Residuals:
     Min       1Q   Median       3Q      Max
-3.07213 -1.11170 -0.04373  1.06533  2.83495

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   20.778      0.713  29.141 2.71e-15 ***
X             17.652      0.410  43.050  < 2e-16 ***
GroupCa_Li   -21.196      1.166 -18.173 4.17e-12 ***
GroupCa_Na    -7.852      1.057  -7.429 1.43e-06 ***
GroupSr_Na     6.136      1.002   6.126 1.46e-05 ***
---
```

In the "Coefficients" section of the output above, we see all of the pieces of information we need to plot our regression lines. The first row, `(Intercept)`, contains the estimate for the y-intercept of the Ca-K group. The second row, `X`, contains the estimate for the slope coefficient. To extract these values, we will first save the results of the *summary.lm* function and then use the familiar `[]` brackets notation:

```
coeffs <- summary.lm(model3)$coefficients
slope <- coeffs[2,1]
intercept1 <- coeffs[1,1]
```

To obtain the y-intercept values for the other 3 groups (Ca-Li, Ca-Na, and Sr-Na), we must add the value listed for the coefficient estimate to `intercept1`, as shown below:

```
intercept2 <- intercept1 + coeffs[3,1]
intercept3 <- intercept1 + coeffs[4,1]
intercept4 <- intercept1 + coeffs[5,1]
```

Now we have the values we need to plot the regression lines, but we also need to plot the points representing each replicate observation. To do this, we will use the *split* function to split the c and Y variables into subsets by the factor Group:

```
## split the covariate and response data based on Group
split_c_pts <- split(c, Group)
split_Y_pts <- split(Y, Group)
```

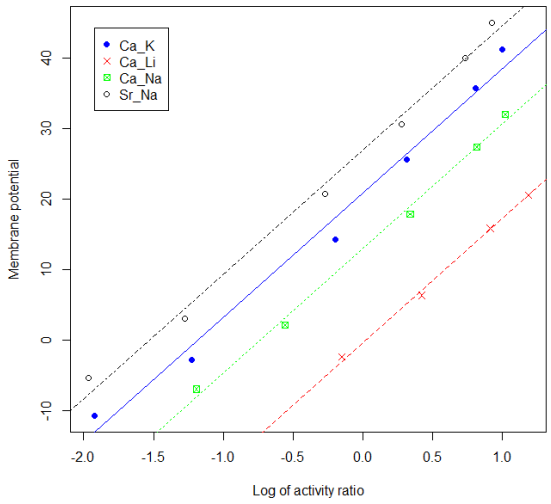
To draw these points, we will use the familiar *plot* function to setup the drawing environment. Next we will create a set of plotting symbols and store them in the plot_chars vector, and then we will use the *legend* function to label these symbols. Finally, we will use a series of calls to the *points* function to plot the points for each treatment level (in the script we also assign them different colors):

```
## plot the observations for each group
plot(c, Y, type="n", ylab= "Membrane potential", xlab="Log of activity ratio")
plot_chars <- c(16, 4, 7, 1)
legend("topleft", inset=0.05, legend=levels(Group), pch=plot_chars)
points(split_c_pts[[1]], split_Y_pts[[1]], pch=plot_chars[1])
points(split_c_pts[[2]], split_Y_pts[[2]], pch=plot_chars[2])
points(split_c_pts[[3]], split_Y_pts[[3]], pch=plot_chars[3])
points(split_c_pts[[4]], split_Y_pts[[4]], pch=plot_chars[4])
```

Now that we have plotted the points, we can draw the regression line for each treatment level. As noted earlier, the slope is constant for all 4 groups but they each have a different y-intercept value. Thus, we use a series of calls to the *abline* function in which we pass the appropriate intercept value:

```
## plot the regression lines for each group
abline(intercept1, slope, lty=1)
abline(intercept2, slope, lty=2)
abline(intercept3, slope, lty=3)
abline(intercept4, slope, lty=4)
```

The plot for our ANCOVA results is shown below:



Part III. ANCOVA using OpenBUGS

The following code estimates the intercepts values of each group "a" as well as the slope of the covariate "c" using OpenBUGS:

```
library(R2OpenBUGS)
n_obs <- length(Group)
z <-c(rep(1,4), rep(2,5), rep(3,6), rep(4,6))

### Fitting the model
# Write model
Ancova<-function()
## Priors
{
   c ~ dnorm(0,1.0E-6)
 for (i in 1:4)
   {
     a[i] ~ dnorm(0,1.0E-6)
   }
 tau ~ dgamma(0.001,0.001)

## Likelihood
 for (i in 1:n)
   {
   mean[i] <- a[z[i]] + c*x[i]
   Y[i]  ~ dnorm(mean[i],tau)
      }
  }
write.model(Ancova,"Ancova.txt")

# Bundle data
win.data <- list(Y=Y,x=X,z=z, n=n_obs)

# Inits function
inits <- function()
   list(a=c(0,0,0,0), c=0, tau=runif(1))

# Parameters to estimate
params <- c("a","c")

# MCMC settings
nc = 3
ni=50000
nb=5000
nt=100

# Start Gibbs sampler
out <- bugs(data = win.data, inits = inits, parameters = params, model =
"Ancova.txt",
n.thin = nt, n.chains = nc, n.burnin = nb, n.iter = ni, codaPkg = TRUE, digits=5)
library(coda)
reg.coda<-read.bugs(out)
results<-summary(reg.coda)
results
```

We get the following output (remember your results may be slightly different), that shows us the estimates of the parameters and their credibility intervals directly (no need to add/substract). Notice that the order in which they are given does not necessarily correspond to the order of the Frequentist results (by default, OpenBUGS calculates the parameters in the order you entered them; whereas R does them alphabetically). Compare them to the values estimated with the Frequentist analysis (shown below):

```
> slope
[1] 17.6517
> intercept1 [Ca_K]
[1] 20.77757
> intercept2 [Ca_Li]
[1] -0.4186184
> intercept3 [Ca_Na]
[1] 12.92533
> intercept4 [Sr_Na]
[1] 26.91373

1. Empirical mean and standard deviation for each variable,plus standard error of
the mean:

            Mean     SD Naive SE Time-series SE
a[1]     -0.4203 0.9626 0.002620       0.002592 intercept Ca_Li
a[2]     12.9258 0.8313 0.002262       0.002344 intercept Ca_Na
a[3]     20.7801 0.7625 0.002075       0.002120 intercept Ca_K
a[4]     26.9163 0.7635 0.002078       0.002359 intercept Sr_Na
c        17.6529 0.4383 0.001193       0.001267 slope
deviance 84.0629 4.0837 0.011115       0.010415

2. Quantiles for each variable:

           2.5%   25%     50%     75%  97.5%
a[1]     -2.333 -1.04 -0.4228  0.2034  1.486
a[2]     11.270 12.39 12.9200 13.4600 14.580
a[3]     19.270 20.29 20.7800 21.2700 22.290
a[4]     25.400 26.42 26.9100 27.4100 28.430
c        16.790 17.37 17.6500 17.9400 18.530
deviance 78.460 81.06 83.2900 86.2300 94.060
```

As usual we finish by detaching the data set:

```
## detach the data
detach(ancova_data)
```

PS. If we have chosen to run the model with the interaction (if it had been significant), we would need parameter estimates for the four slope terms, just as we needed four intercept terms. The basic way to calculate and plot them would be the same, and as with the intercept, the Bayesian output will give the estimates directly, together with their credibility intervals.