ANCOVA

Analysis of covariance

The analysis of **covariance** is a hybrid of regression and ANOVA and measures the variance within each treatment level attributed to a continuous variable.

If a simple ANOVA can be expressed as $Y_i = \alpha + \beta_{ji} * A + \epsilon_i$, where j is each level of the treatment A and i is the replicate, then an ANCOVA can be expressed in two different ways:

- In each group, only the mean of the covariate differs, but the slope is the same across group: $Y_{ij} = \alpha + \beta_{ji} * A + \delta_i * cov + \epsilon_i$
- In each group, the mean as well as the slope of the cavariate differ:

$$Y_{ij} = \alpha + \beta_{ji} * A + \delta_i * cov + \gamma_{ji} * cov + \epsilon_i$$

Let's look at a concrete example

I got this simulation example from Kery (2010) Introduction to WinBUGS for Ecologists

Here, we consider the relationship between body mass and body length of the asp viper (*Vipera aspis*) in three populations: Pyrenees, Massif Central, Jura Mountains. We simulate the data inclduing an interaction effect and fit two different models to the simulated data:

```r
n.groups <- 3
n.sample <- 10
n <- n.groups * n.sample        # Total number of data points
x <- rep(1:n.groups, rep(n.sample, n.groups)) # Indicator for population
pop <- factor(x, labels = c("Pyrenees", "Massif Central", "Jura"))
length <- runif(n, 45, 70)      # Obs. body length (cm) is rarely less than 45
```

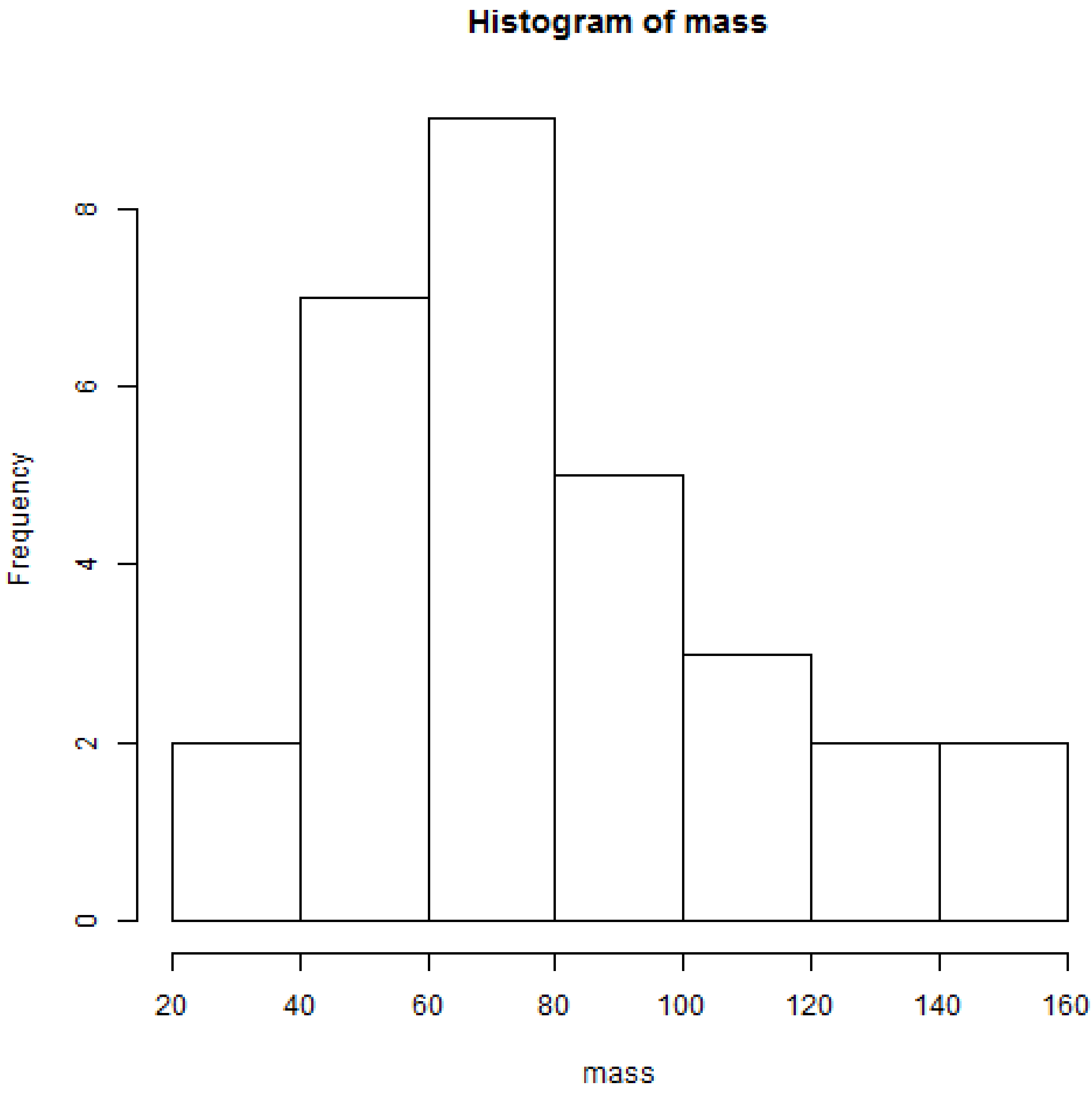
```r
Xmat <- model.matrix(~ pop*length)
print(Xmat, dig = 2)
```

```
   (Intercept) popMassif Central popJura length popMassif Central:length
1            1                 0       0     49                        0
2            1                 0       0     46                        0
3            1                 0       0     60                        0
4            1                 0       0     68                        0
5            1                 0       0     66                        0
6            1                 0       0     64                        0
7            1                 0       0     47                        0
8            1                 0       0     63                        0
9            1                 0       0     59                        0
10           1                 0       0     53                        0
11           1                 1       0     69                       69
12           1                 1       0     46                       46
13           1                 1       0     70                       70
14           1                 1       0     55                       55
```

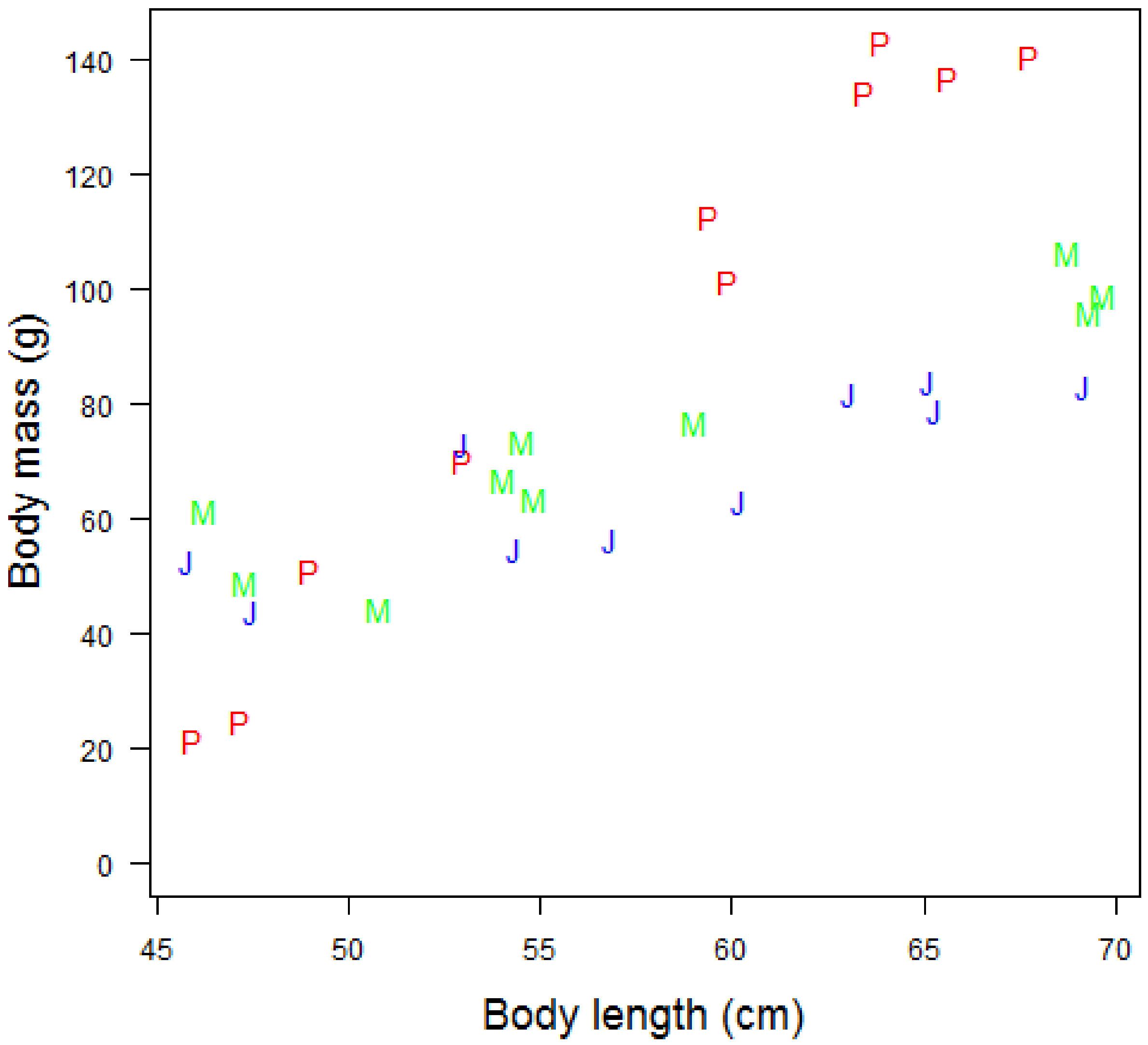
```
15           1                 1       0     47                       47
16           1                 1       0     69                       69
17           1                 1       0     55                       55
18           1                 1       0     54                       54
19           1                 1       0     51                       51
20           1                 1       0     59                       59
21           1                 0       1     65                        0
22           1                 0       1     69                        0
23           1                 0       1     57                        0
24           1                 0       1     54                        0
25           1                 0       1     60                        0
26           1                 0       1     65                        0
27           1                 0       1     63                        0
28           1                 0       1     46                        0
29           1                 0       1     47                        0
30           1                 0       1     53                        0
   popJura:length
1               0
2               0
3               0
4               0
5               0
6               0
7               0
8               0
9               0
10              0
11              0
12              0
13              0
14              0
15              0
16              0
17              0
18              0
19              0
20              0
21             65
22             69
23             57
24             54
25             60
26             65
27             63
28             46
29             47
30             53
attr(,"assign")
[1] 0 1 1 2 3 3
attr(,"contrasts")
attr(,"contrasts")$pop
[1] "contr.treatment"
```

```r
beta.vec <- c(-250, 150, 200, 6, -3, -4)
```

```r
lin.pred <- Xmat[,] %*% beta.vec    # Value of lin.predictor
eps <- rnorm(n = n, mean = 0, sd = 10)  # residuals
mass <- lin.pred + eps          # response = lin.pred + residual
hist(mass)              # Inspect what we've created
```



```r
matplot(cbind(length[1:10], length[11:20], length[21:30]), cbind(mass[1:10], mass[11:20],
mass[21:30]), ylim = c(0, max(mass)), ylab = "Body mass (g)", xlab = "Body length (cm)",
col = c("Red","Green","Blue"), pch = c("P","M","J"), las = 1, cex = 1.2, cex.lab = 1.5)
```



Common slope - different intercepts model

```
mod.csl=lm(mass ~ pop + length)
```

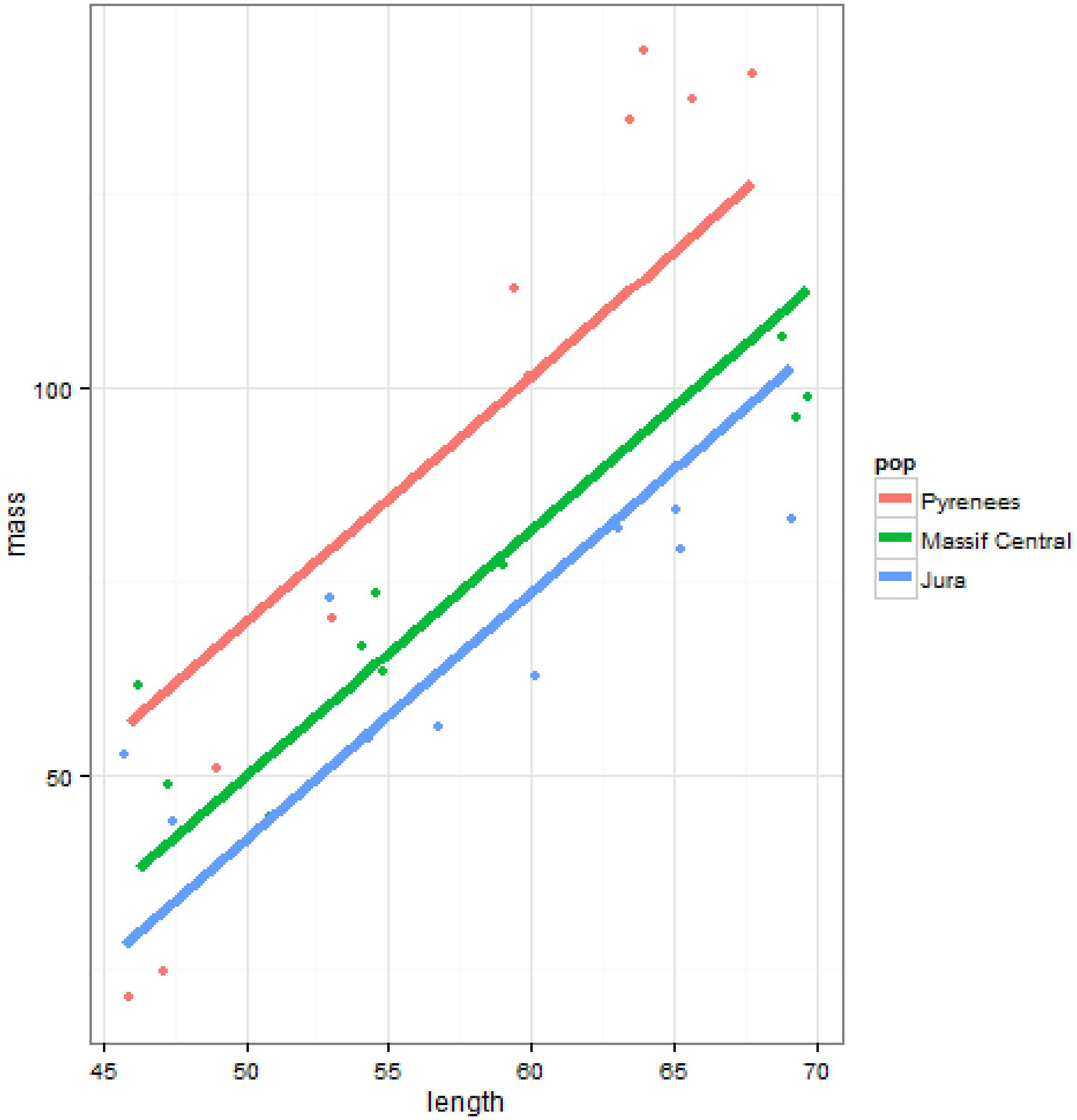
Visually, it would look like:

```
library(ggplot2)
```

```
data=data.frame(mass=mass,length=length,pop=pop,
                pred=predict(mod.csl))

plot.csl = ggplot(data, aes(x=length, y=mass,col=pop)) + geom_point() +
  geom_line(aes(x=length, y=pred),size=1.5)+
  theme_bw()
```

```
plot.csl
```



Different slopes - different intercepts model

```
mod.dsl=lm(mass ~ pop * length)
```

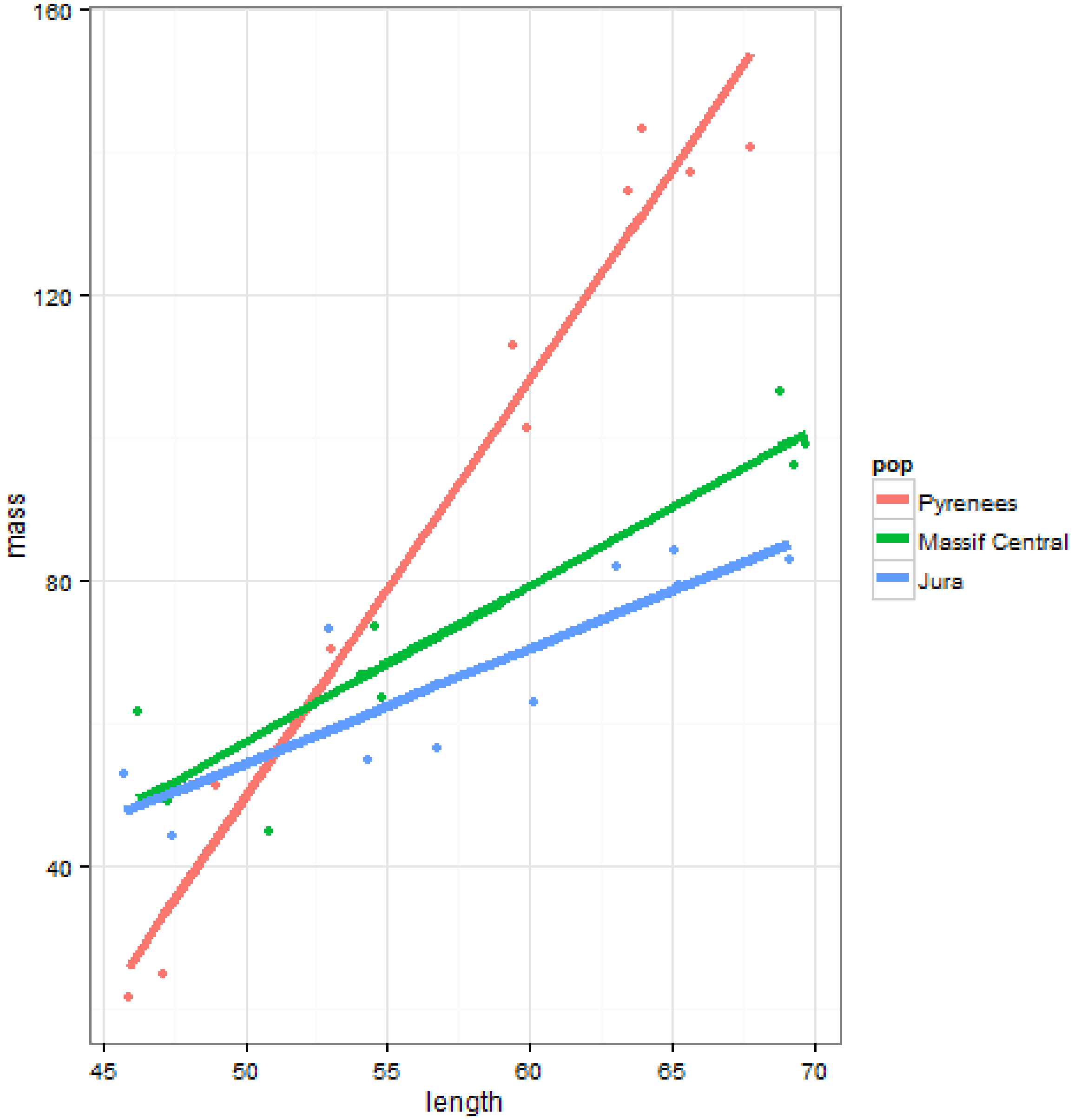
Visually, it would look like:

```
library(ggplot2)

data=data.frame(mass=mass,length=length,pop=pop,
                pred=predict(mod.dsl))

plot.dsl = ggplot(data, aes(x=length, y=mass,col=pop)) + geom_point() +
  geom_line(aes(x=length, y=pred),size=1.5)+
  theme_bw()
```

```
plot.dsl
```


160
120
80
40
mass
45
50
55
60
65
70
length
pop
Pyrenees
Massif Central
Jura