Two-Factor ANOVA and Interaction effects

author: Maria Paniw date: 09.02.2016

The simple case where you have one continuous response and one categorical predictor is not reality (maybe only in laboratory studies). In reality, we try to estimate the effect of several (categorical) factors on a response. So we enter the world of two- (or multi-) factor ANOVA. Now, we may have two types of effects: additive effects of our factors, and/or interaction effects.

Example: two factors additive effects

Let's simulate some data. We still assume we are measuring the weight of birds, but now we are doing it 45 sites, each described by a different combination of **vegetation cover** and mean summer **temperatures**

Vegetation cover consists of three levels: little, medium high Temparature also consists of three levels: cold, mild, hot We have five replicates for each combination beacause we measure the weight of 5 birds for each treatment combination.

```
nVeg= nTemp =3 # number of vegetation and temperature levels

nbirds=5 # samples per treatment combination
#Within each population, the weight of butterflies
sigma=1.5 # residual standaed deviation
```

```
n = nTemp*nVeg*3*5 #total number of samples

set.seed(20)
eps= rnorm(n,0,sigma) # random variation

#generate factor levels
veg=gl(n=nVeg, k=nTemp*nVeg,length=n)
temp=gl(n=nTemp, k=nTemp,length=n)
```

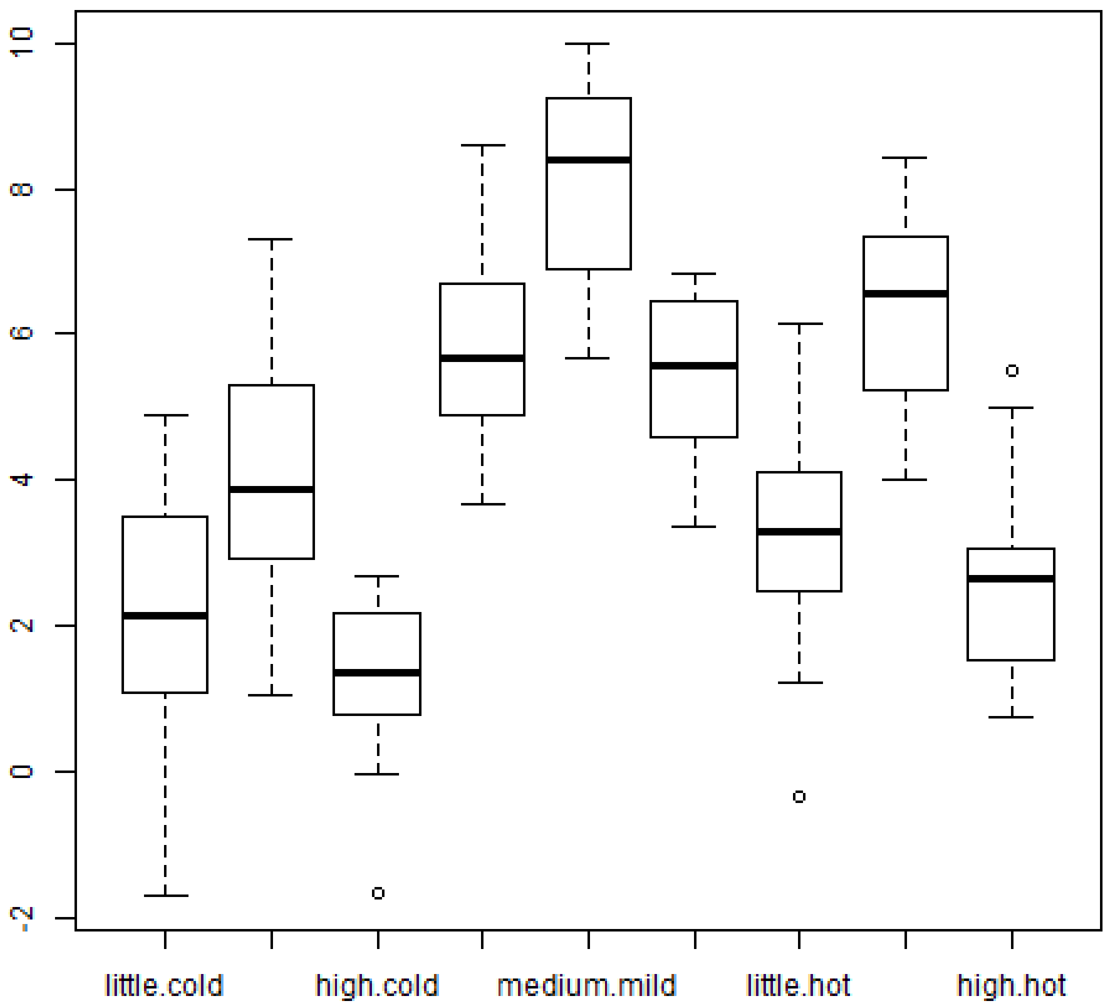
```
#Choose the baseline effect

baseline=2 #Intercept (weight at veg little and temp cold)

veg.effect=c(2,-1) # differences for each vegetation type to intercept
temp.effect=c(4,2) # difference of temperature to intercept
```

The model matrix

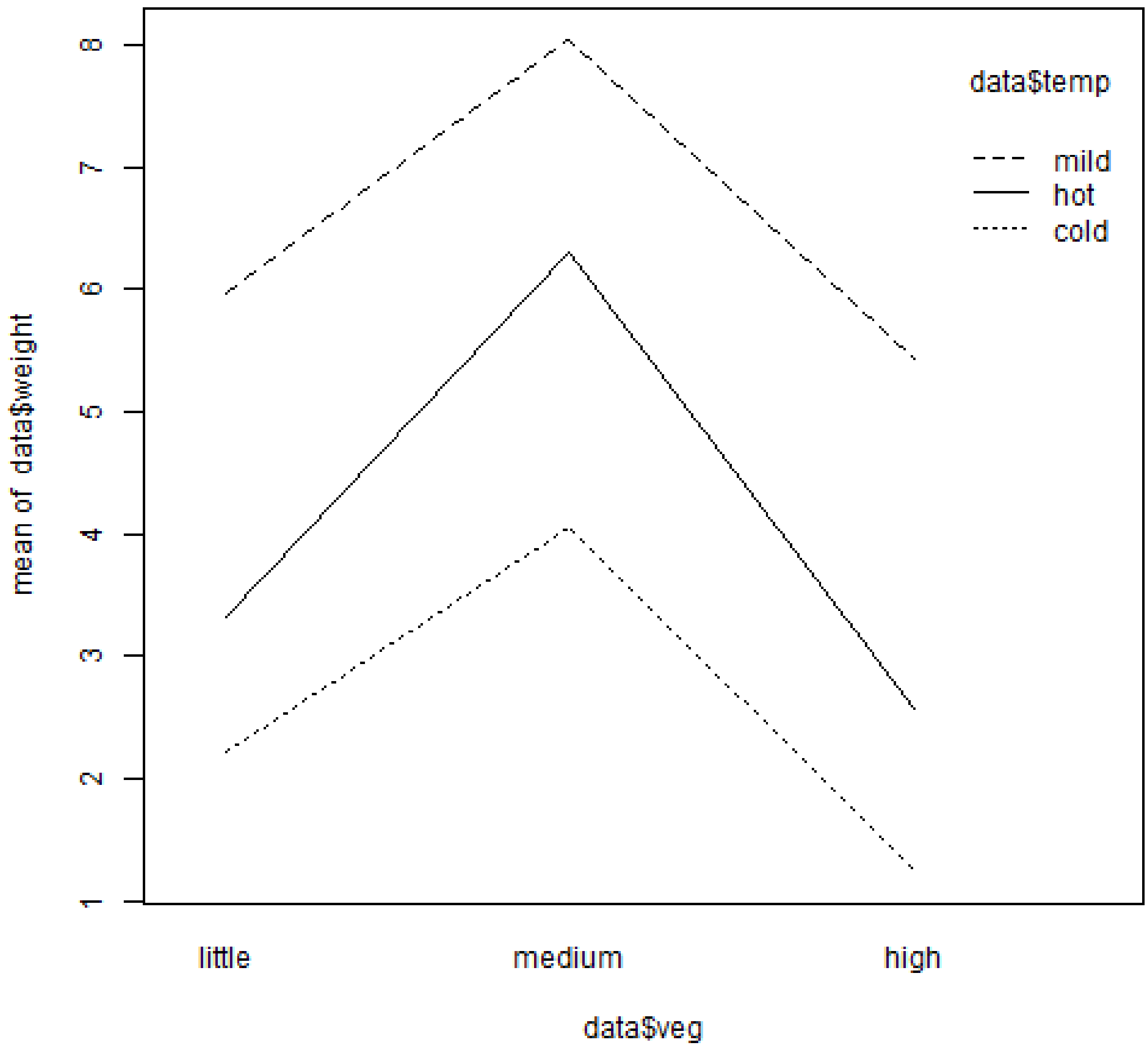
```
# Have a look at the model matrix

all.effects=c(baseline,veg.effect,temp.effect)
X=as.matrix(model.matrix(~veg+temp))

#Create the response
Y=as.numeric(as.matrix(X)%*%as.matrix(all.effects) +eps)
```

```r
# create a dataframe for easy plotting
# 1. rename levels:
levels(veg)=c("little","medium","high")
levels(temp)=c("cold","mild","hot")

# 2. Create a data frame
data=data.frame(weight=Y,veg=veg,temp=temp)
```

```r
# make some plots
boxplot(weight~veg+temp,data)
```



```r
# a better visualization:
interaction.plot(data$veg,data$temp,data$weight)
```




```r
# You can see that the effect is additive!
```

Model fitting

Finally, let's fit a model

The *effects* parameterization of the model, as we have seen from the model matrix above, may be expressed as:

$$weight_i = \alpha + \beta_{j(i)} * veg_i + \delta_{k(i)} * temp_i + \epsilon_i$$

The means parameterization does not work well with more than 1 factor in `lm`. It does in Bayesian statistics, though.

```r
#In R, the according model is:
mod1=lm(weight~veg+temp,data)

summary(mod1)
```

```
Call:
lm(formula = weight ~ veg + temp, data = data)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.8896 -0.9177  0.0314  1.1044  3.0244 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   1.9964     0.2930   6.814 3.20e-10 ***
vegmedium     2.2919     0.3210   7.140 5.90e-11 ***
veghigh      -0.7514     0.3210  -2.341   0.0208 *  
tempmild      3.9656     0.3210  12.355  < 2e-16 ***
temphot       1.5586     0.3210   4.856 3.38e-06 ***

Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.523 on 130 degrees of freedom
Multiple R-squared:  0.6602,	Adjusted R-squared:  0.6497 
F-statistic: 63.14 on 4 and 130 DF,  p-value: < 2.2e-16
```

Now, if I want to know the mean of **tempmild**, given veg is little?

```r
coef(mod1)[1]+coef(mod1)[4]
```

```
(Intercept) 
   5.962055 
```

If I want the mean of **temphot**, given veg is medium?

```r
coef(mod1)[1]+coef(mod1)[2]+coef(mod1)[5]
```

```
(Intercept) 
   5.846918 
```

If just want the average weight at **temphot**?

```r
(coef(mod1)[1]+coef(mod1)[5]+coef(mod1)[1]+coef(mod1)[2]+coef(mod1)[5]+
  coef(mod1)[1]+coef(mod1)[3]+coef(mod1)[5])/3
```

```
(Intercept) 
   4.068542 
```

Interactions!!!!

Let's simulate data including interaction effects. We will use the same simulation as above, with one little add-on.

```
nVeg= nTemp =3 # number of vegetation and temperature levels

nbirds=5 # samples per treatment combination
#Within each population, the weight of butterflies
sigma=1.5 # residual standaed deviation

n = nTemp*nVeg*3*5 #total number of samples

set.seed(20)
eps= rnorm(n,0,sigma) # random variation

#generate factor levels
veg=gl(n=nVeg, k=nTemp*nVeg,length=n)
temp=gl(n=nTemp, k=nTemp,length=n)
```

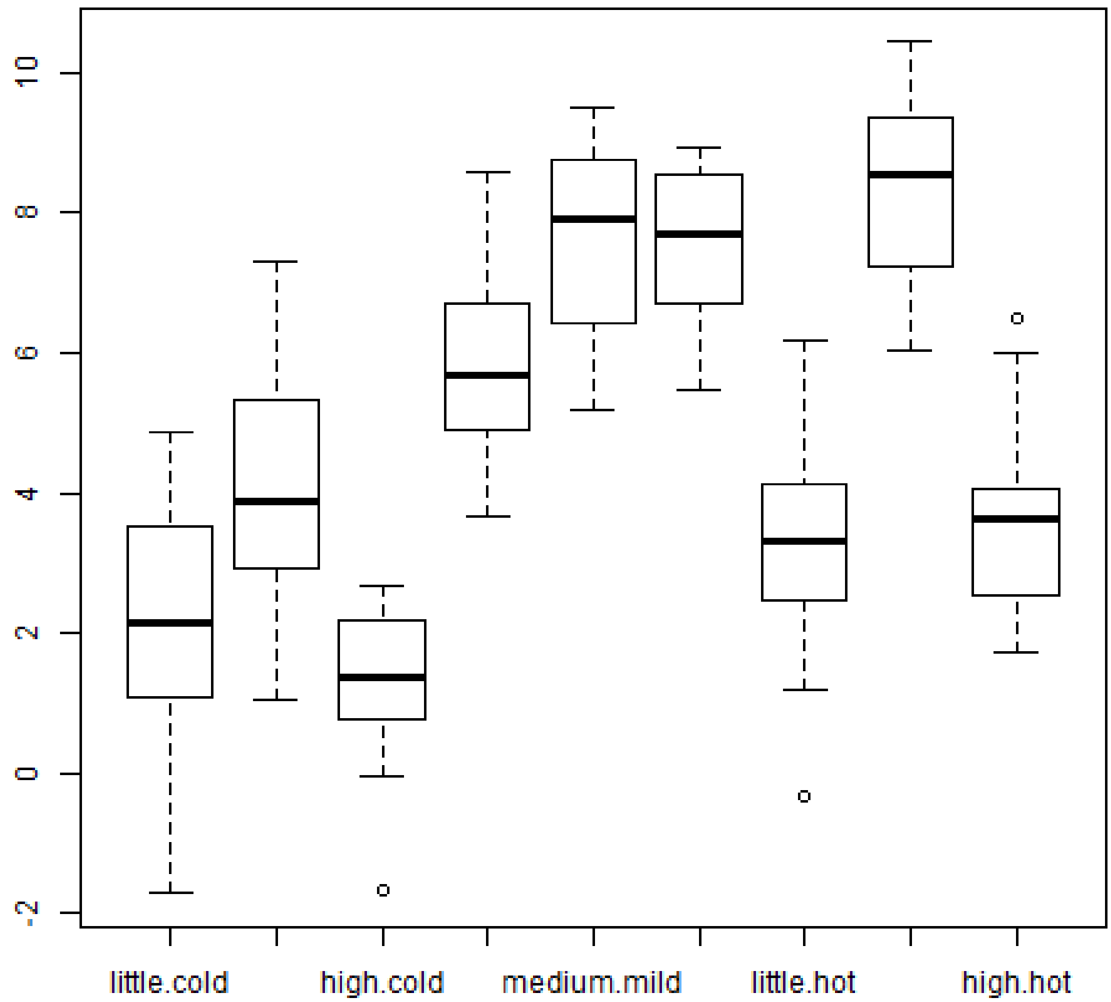
```
#Choose the baseline effect

baseline=2 #Intercept (weight at veg little and temp cold)

veg.effect=c(2,-1) # differences for each vegetation type to intercept
temp.effect=c(4,2) # difference of temperature to intercept

# We add an interaction effect!
int.effect=c(-.5,2.1,2,1)
```

The model matrix

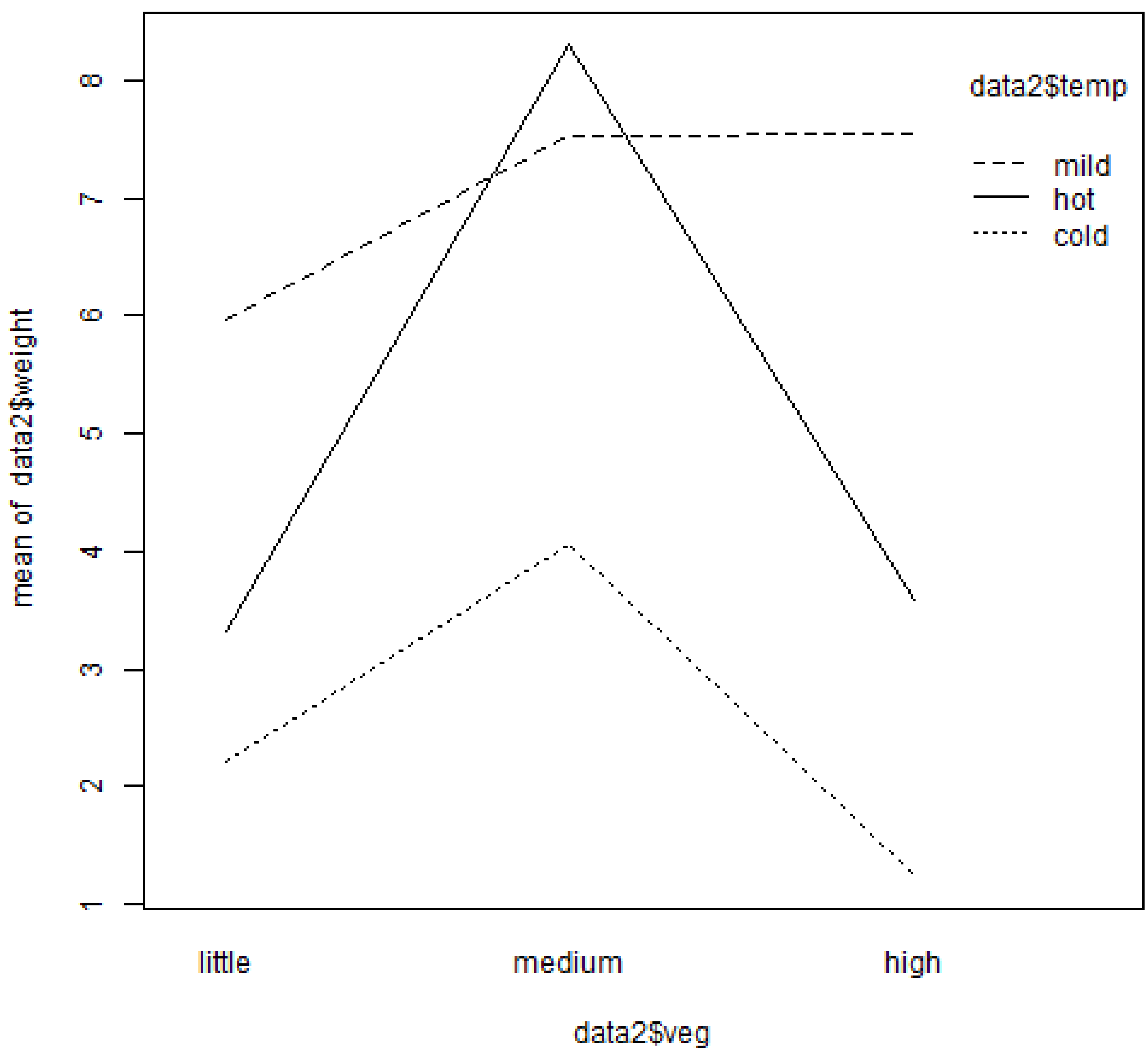
```
# Have a look at the model matrix

all.effects=c(baseline,veg.effect,temp.effect,int.effect)
X=as.matrix(model.matrix(~veg*temp))

#Create the response
Y=as.numeric(as.matrix(X)%*%as.matrix(all.effects) +eps)
```

```
# create a dataframe for easy plotting
# 1. rename levels:
levels(veg)=c("little","medium","high")
levels(temp)=c("cold","mild","hot")

# 2. Create a data frame
data2=data.frame(weight=Y,veg=veg,temp=temp)
```

```r
# make some plots
boxplot(weight~veg*temp,data2)
```




```r
# a better visualization:
 interaction.plot(data2$veg,data2$temp,data2$weight)
```




```
# You can see that the effect is additive!
```

Model fitting: Interactions

The *effects* parameterization of the model, as we have seen from the model matrix above, may be expressed as:

$$weight_i = \alpha + \beta_{j(i)} * veg_i + \delta_{k(i)} * temp_i + \gamma_{jk(i)} * veg_i * temp_i + \epsilon_i$$

The means parameterization is:

$$weight_i = \alpha_{jk(i)} * veg_i * temp_i + \epsilon_i$$

But again, it does not work well with more than 1 factor in `lm`.

```r
#In R, the according model is:
mod2=lm(weight~veg*temp,data2)

summary(mod2)
```

```
Call:
lm(formula = weight ~ veg * temp, data = data2)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.9357 -1.0558  0.0591  0.9886  3.2512 

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)    
(Intercept)          2.2237     0.3942   5.641 1.06e-07 ***
vegmedium            1.8378     0.5575   3.296 0.001273 ** 
veghigh             -0.9792     0.5575  -1.756 0.081444 .  
tempmild             3.7352     0.5575   6.700 6.23e-10 ***
temphot              1.1071     0.5575   1.986 0.049226 *  
vegmedium:tempmild  -0.2661     0.7884  -0.338 0.736274    
veghigh:tempmild     2.5573     0.7884   3.244 0.001511 ** 
vegmedium:temphot    3.1285     0.7884   3.968 0.000121 ***
veghigh:temphot      1.2261     0.7884   1.555 0.122416    

Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.527 on 126 degrees of freedom
Multiple R-squared:  0.7263,    Adjusted R-squared:  0.7089 
F-statistic: 41.79 on 8 and 126 DF,  p-value: < 2.2e-16
```

Now, if I want to know the mean of **tempmild**, given veg is little?

```r
coef(mod2)[1]+coef(mod2)[4]
```

```
(Intercept) 
   5.958968 
```

If I want the mean of **temphot**, given veg is medium?

```r
coef(mod2)[1]+coef(mod2)[2]+coef(mod2)[5]+coef(mod2)[8]
```

```
(Intercept) 
   8.297042 
```

If just want the average weight at **tempmild**?

```r
(coef(mod2)[1]+coef(mod2)[4]+coef(mod2)[1]+coef(mod2)[2]+coef(mod2)[4]+coef(mod2)[6]+
  coef(mod2)[1]+coef(mod2)[3]+coef(mod2)[4]+coef(mod2)[7])/3
```

```
(Intercept)
   7.008876
```

This is a bit tricky. You can also fit an ANOVA in R using the `aov` function

```
mod3=aov(weight~veg*temp,data)
model.tables(mod3, type="means", se=T)
```

```
Tables of means
Grand mean

4.351331

 veg
veg
little medium   high
 3.838  6.130  3.086

 temp
temp
 cold  mild   hot
2.510 6.476 4.069

 veg:temp
        temp
veg      cold  mild  hot
  little 2.224 5.959 3.331
  medium 4.061 8.031 6.297
  high   1.245 5.437 2.578

Standard errors for differences of means
           veg   temp veg:temp
        0.3219 0.3219   0.5575
replic.     45     45       15
```