

Chapter 11 Analysis of Variance | Chapter 12 Analysis of Covariance

Qianqian Shan

June 3, 2017

Chapter 11 Analysis of Variance

Regression and ANOVA are identical approaches except for the nature of the explanatory variables. The emphasis in ANOVA was traditionally on hypothesis testing.

One-way ANOVA

Assumptions for ANOVA:

1. random sampling
2. errors are normally distributed and independent
3. equal variance
4. additivity of treatment effects.

```
results <- read.table("yields.txt", header = TRUE)
attach(results)
names(results)
```

```
## [1] "sand" "clay" "loam"
```

```
results
```

```
##      sand clay loam
## 1      6   17   13
## 2     10   15   16
## 3      8    3    9
## 4      6   11   12
## 5     14   14   15
## 6     17   12   16
## 7      9   12   17
## 8     11    8   13
## 9      7   10   18
## 10     11   13   14
```

```
# group means
sapply(list(sand, clay, loam), mean)
```

```
## [1]  9.9 11.5 14.3
```

```
lapply(list(sand, clay, loam), mean)
```

```
## [[1]]
## [1] 9.9
##
## [[2]]
## [1] 11.5
```

```
##
## [[3]]
## [1] 14.3

#
frame <- stack(results)
head(frame, 3)

##   values ind
## 1      6 sand
## 2     10 sand
## 3      8 sand

names(frame) <- c("yield", "soil")
attach(frame)
tapply(yield, soil, var)

##      clay      loam      sand
## 15.388889  7.122222 12.544444

# test for homogeneous variance
fligner.test(yield ~ soil) # no significant difference

##
## Fligner-Killeen test of homogeneity of variances
##
## data:  yield by soil
## Fligner-Killeen:med chi-squared = 0.36507, df = 2, p-value =
## 0.8332

detach(results)
```

Effect sizes

1. Use `plot.design` which takes a formula when more than one factors.
2. Use `model.tables` which takes a model object when there is at least one factor, computes summary tables for model fits, especially complex **aov** fits.
3. Use `summary.lm`

Plots for interpreting one-way ANOVA:

- box and whisker
- barplots with error bars

```
model <- aov(yield ~ soil)

model.tables(model, se = TRUE) # compute standard error

## Tables of effects
##
##  soil
##  soil
## clay loam sand
## -0.4  2.4 -2.0
##
## Standard errors of effects
```

```

##          soil
##          1.081
## replic.    10
model.tables(model, type = "means", se = TRUE)

## Tables of means
## Grand mean
##
## 11.9
##
##  soil
##  soil
## clay loam sand
## 11.5 14.3  9.9
##
## Standard errors for differences of means
##          soil
##          1.529
## replic.    10
model.tables(model, type = "effects", se = TRUE)

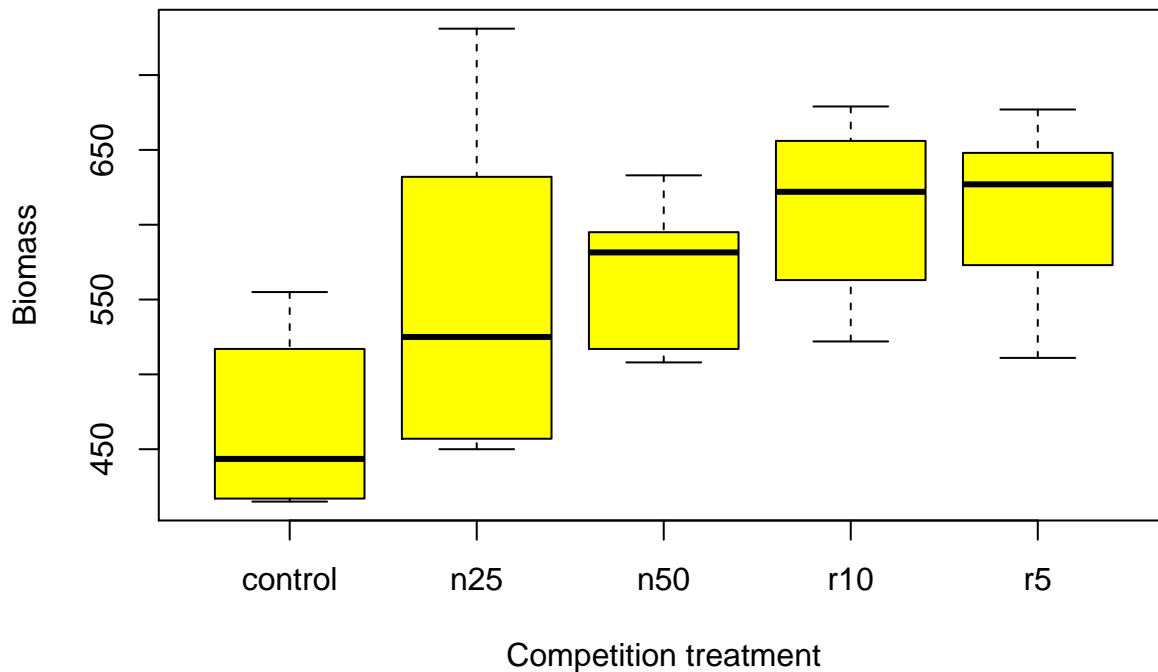
## Tables of effects
##
##  soil
##  soil
## clay loam sand
## -0.4  2.4 -2.0
##
## Standard errors of effects
##          soil
##          1.081
## replic.    10
detach(frame)

# plots

# boxplot
comp <- read.table("competition.txt", header = TRUE)
attach(comp)
names(comp)

## [1] "biomass" "clipping"
plot(clipping, biomass, xlab = "Competition treatment",
     ylab = "Biomass", col = "yellow")

```



```
# error bars
error.bars <- function(yv, z, nn)
{
  xv <- barplot(yv, ylim = c(0, (max(yv) + max(z))),
    col = "green", names = nn,
    ylab = deparse(substitute(yv)))
  # add bars to barplot
  for (i in 1:length(xv)) {
    arrows(xv[i], yv[i] + z[i], xv[i], yv[i] - z[i],
      angle = 90, code = 3, length = 0.15)
    # length is the length of the edges of the arrow head (in inches).
  }
}
```

```
model <- aov(biomass ~ clipping)
summary(model)
```

```
##              Df Sum Sq Mean Sq F value    Pr(>F)
## clipping      4  85356    21339   4.302 0.00875 **
## Residuals    25 124020     4961
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
table(clipping) # equal size
```

```
## clipping
## control      n25      n50      r10      r5
##           6      6      6      6      6
```

```
str(summary(model)[[1]])
```

```
## Classes 'anova' and 'data.frame':  2 obs. of  5 variables:
## $ Df      : num  4 25
## $ Sum Sq  : num  85356 124020
## $ Mean Sq: num  21339 4961
```

```
## $ F value: num 4.3 NA
## $ Pr(>F) : num 0.00875 NA

sigma.square <- summary(model)[[1]][3][2, ] # the mse

se1 <- sqrt(sigma.square/6) # standard error of the means

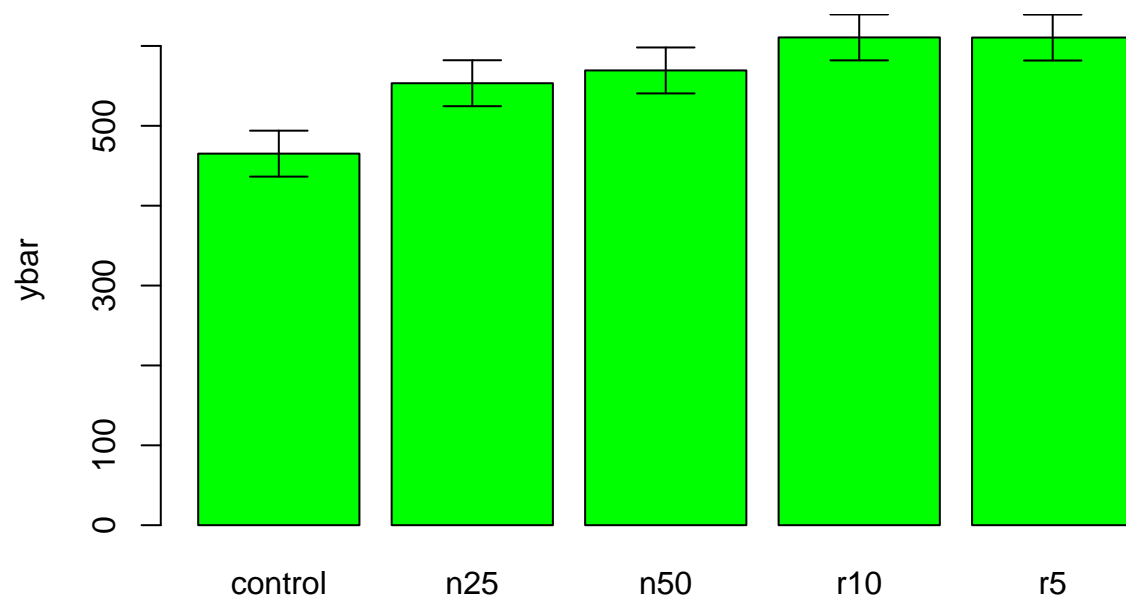
se <- rep(se1, 5)
se

## [1] 28.75417 28.75417 28.75417 28.75417 28.75417

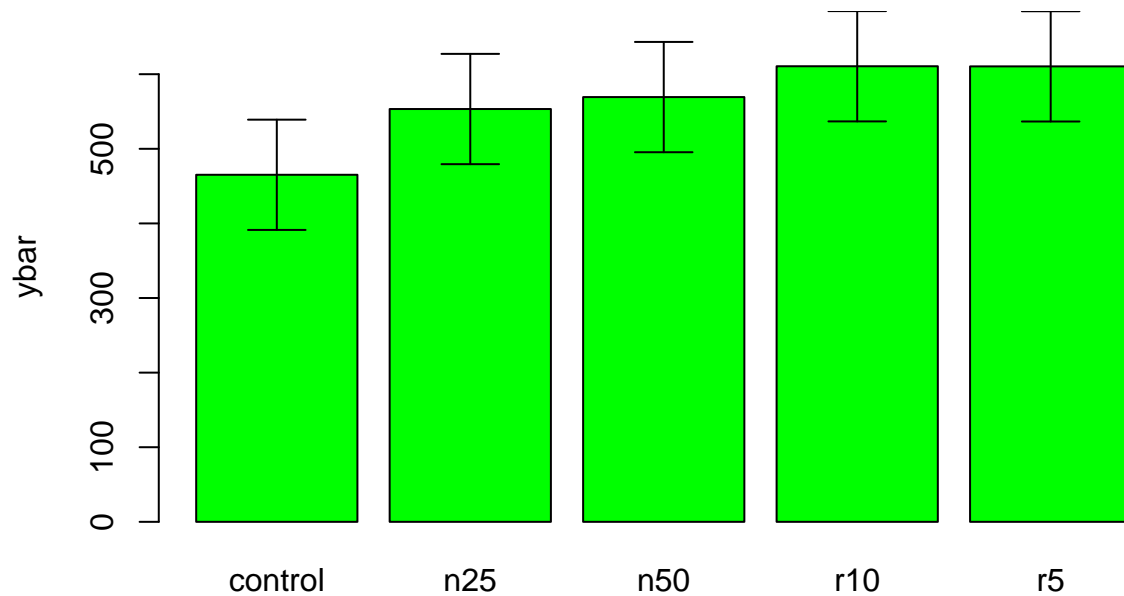
labels <- levels(clipping)

ybar <- tapply(biomass,clipping,mean)

error.bars(ybar, se, labels)
```



```
# use qt(0.975, 5)
error.bars(ybar, 2.570582*se, labels)
```



```
detach(comp)
```

Factorial experiments

Use `aov` or `lm` to fit a factorial analysis of variance.

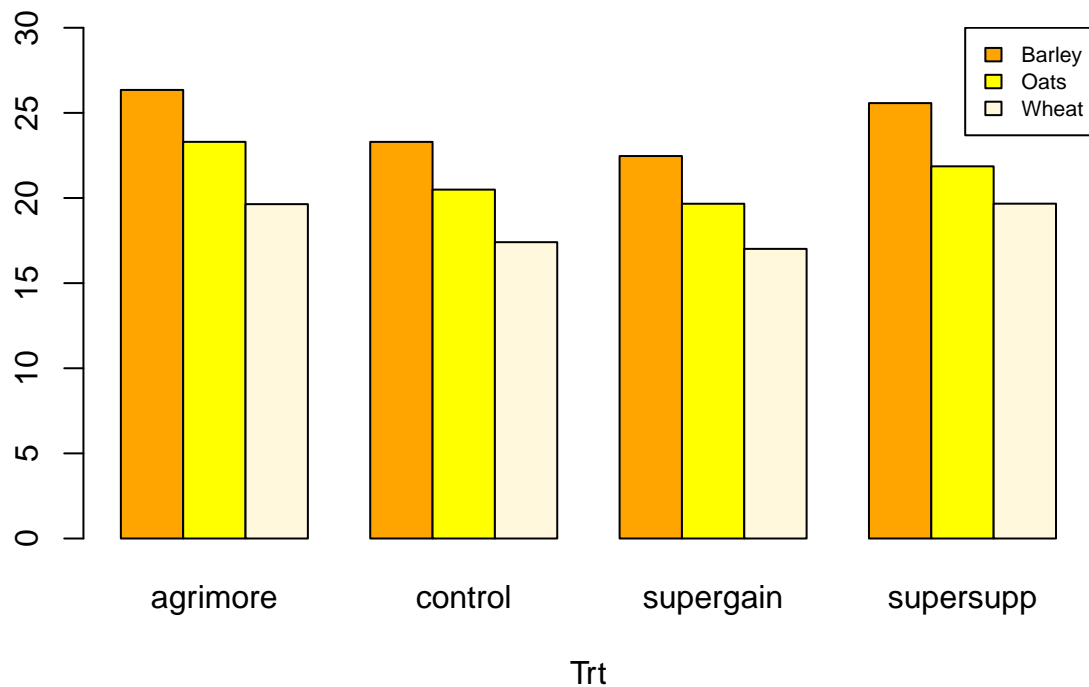
```
weights <- read.table("growth.txt", header = TRUE)
attach(weights)
names(weights)
```

```
## [1] "supplement" "diet"      "gain"
tapply(gain, list(diet,supplement), mean)
```

```
##      agrimore  control  supergain  supersupp
## barley 26.34848 23.29665  22.46612  25.57530
## oats   23.29838 20.49366  19.66300  21.86023
## wheat  19.63907 17.40552  17.01243  19.66834
```

```
barplot(tapply(gain, list(diet,supplement), mean),
        beside = TRUE, ylim=c(0, 30), xlab = "Trt",
        col = c("orange", "yellow", "cornsilk"))
```

```
labs <- c("Barley","Oats","Wheat")
legend("topright",labs,fill= c("orange","yellow","cornsilk"), cex = 0.7)
```



```
# full model
model <- aov(gain ~ diet * supplement)
summary(model)
```

```
##              Df Sum Sq Mean Sq F value    Pr(>F)
## diet           2  287.17   143.59   83.52 3.00e-14 ***
## supplement     3   91.88    30.63   17.82 2.95e-07 ***
## diet:supplement 6    3.41     0.57    0.33  0.917
## Residuals     36   61.89     1.72
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary.lm(model)
```

```
##
## Call:
## aov(formula = gain ~ diet * supplement)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.48756 -1.00368 -0.07452  1.03496  2.68069
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      26.3485     0.6556  40.191 < 2e-16 ***
## dietoats         -3.0501     0.9271  -3.290 0.002248 **
## dietwheat        -6.7094     0.9271  -7.237 1.61e-08 ***
## supplementcontrol -3.0518     0.9271  -3.292 0.002237 **
## supplementsupergain -3.8824     0.9271  -4.187 0.000174 ***
## supplementsupersupp -0.7732     0.9271  -0.834 0.409816
## dietoats:supplementcontrol  0.2471     1.3112   0.188 0.851571
## dietwheat:supplementcontrol  0.8183     1.3112   0.624 0.536512
## dietoats:supplementsupergain  0.2470     1.3112   0.188 0.851652
```

```

## dietwheat:supplementsupergain    1.2557    1.3112    0.958 0.344601
## dietoats:supplementsupersupp    -0.6650    1.3112   -0.507 0.615135
## dietwheat:supplementsupersupp    0.8024    1.3112    0.612 0.544381
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.311 on 36 degrees of freedom
## Multiple R-squared:  0.8607, Adjusted R-squared:  0.8182
## F-statistic: 20.22 on 11 and 36 DF,  p-value: 3.295e-12
# simplify the model by removing the interaction term
model <- aov(gain ~ diet + supplement)
summary.lm(model)

##
## Call:
## aov(formula = gain ~ diet + supplement)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.30792 -0.85929 -0.07713  0.92052  2.90615
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      26.1230     0.4408   59.258 < 2e-16 ***
## dietoats          -3.0928     0.4408  -7.016 1.38e-08 ***
## dietwheat         -5.9903     0.4408 -13.589 < 2e-16 ***
## supplementcontrol -2.6967     0.5090  -5.298 4.03e-06 ***
## supplementsupergain -3.3815     0.5090  -6.643 4.72e-08 ***
## supplementsupersupp -0.7274     0.5090  -1.429    0.16
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.247 on 42 degrees of freedom
## Multiple R-squared:  0.8531, Adjusted R-squared:  0.8356
## F-statistic: 48.76 on 5 and 42 DF,  p-value: < 2.2e-16
# further simplify the model by combining factors
supp2 <- factor(supplement)
levels(supp2)

## [1] "agrimore" "control" "supergain" "supersupp"
levels(supp2)[c(1, 4)] <- "best"
levels(supp2)[c(2, 3)] <- "worst"
levels(supp2)

## [1] "best" "worst"
# fit the model with new supplement levels
model2 <- aov(gain ~ diet + supp2)
anova(model, model2)

## Analysis of Variance Table
##
## Model 1: gain ~ diet + supplement
## Model 2: gain ~ diet + supp2

```



```
##   Res.Df    RSS Df Sum of Sq    F Pr(>F)
## 1      42 65.296
## 2      44 71.284 -2    -5.9876 1.9257 0.1584

summary.lm(model2) # minimal model as all parameters are significant

##
## Call:
## aov(formula = gain ~ diet + supp2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.6716 -0.9432 -0.1918  0.9293  3.2698
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  25.7593     0.3674  70.106 < 2e-16 ***
## dietoats     -3.0928     0.4500  -6.873 1.76e-08 ***
## dietwheat    -5.9903     0.4500 -13.311 < 2e-16 ***
## supp2worst   -2.6754     0.3674  -7.281 4.43e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.273 on 44 degrees of freedom
## Multiple R-squared:  0.8396, Adjusted R-squared:  0.8286
## F-statistic: 76.76 on 3 and 44 DF,  p-value: < 2.2e-16

detach(weights)
```

Pseudoreplication: Nested designs and split plots

Deal with pitfalls of pseudoreplication:

- nested sampling as when repeated measurements are taken from the same individual
- split-plot analysis as when designed experiments have different treatments applied to plots of different sizes

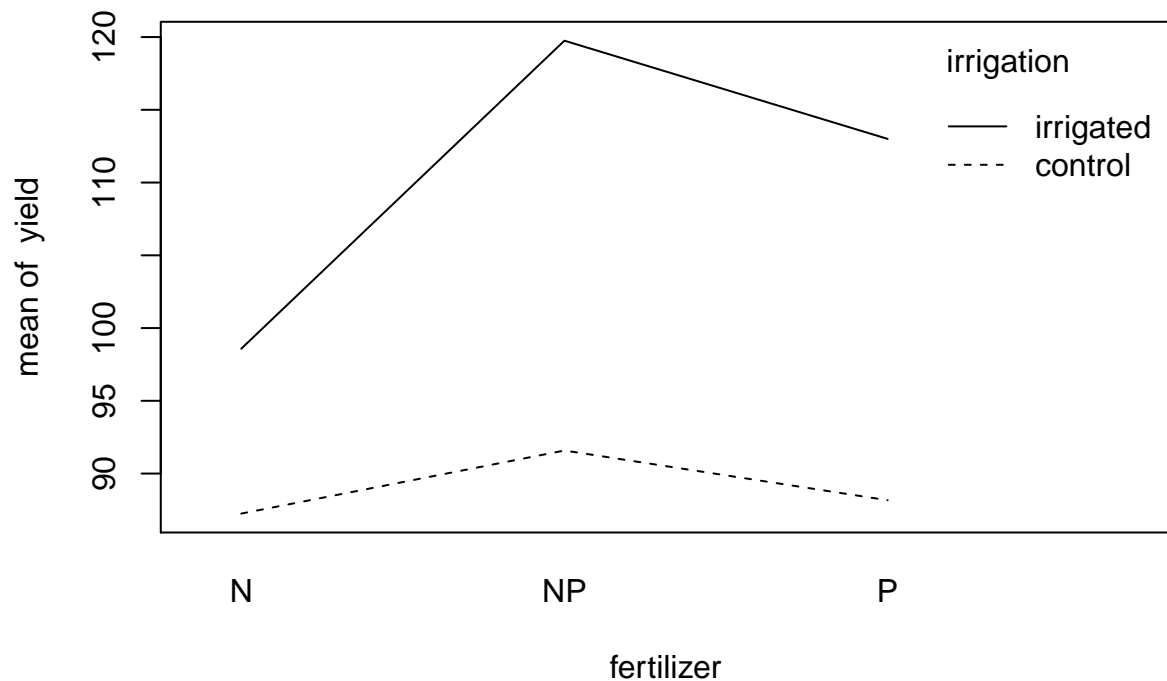
```
yields <- read.table("splityield.txt", header = TRUE)
attach(yields)
names(yields)

## [1] "yield"      "block"      "irrigation" "density"    "fertilizer"
# the Error structure has plot size from largest to smallest from left to right
# the smallest plot size(fertilizer in this case) doesn't need to appear in the Error term
model <- aov(yield ~ irrigation * density * fertilizer + Error(block/irrigation/density))
summary(model)

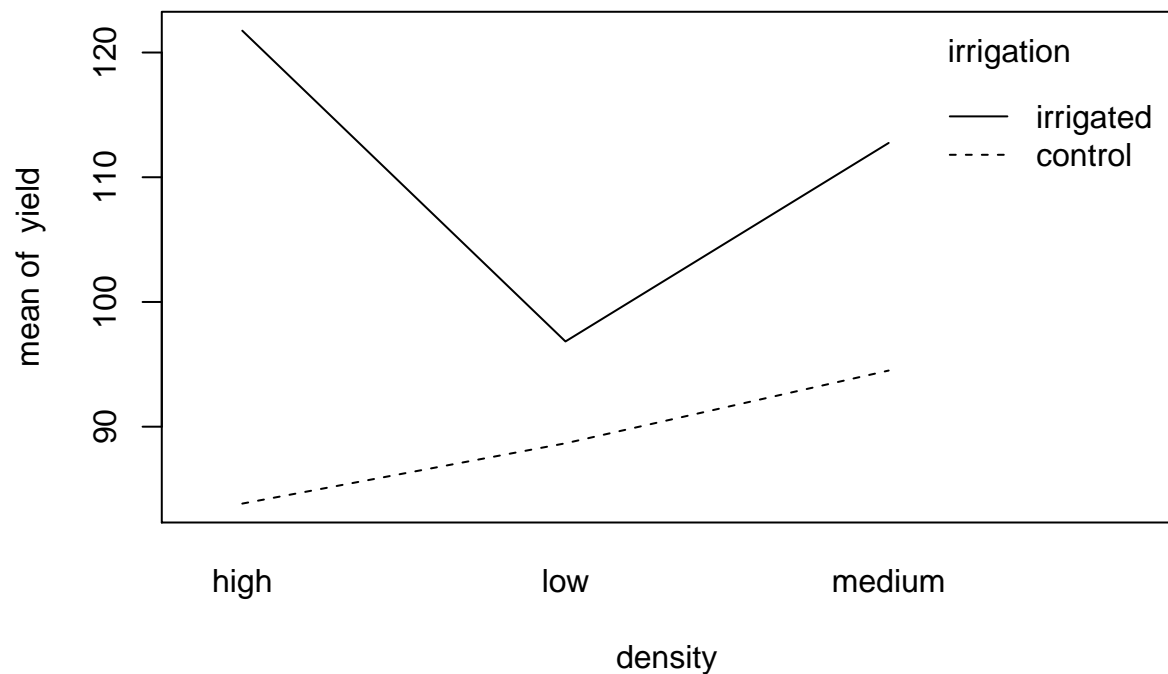
##
## Error: block
##              Df Sum Sq Mean Sq F value Pr(>F)
## Residuals    3  194.4    64.81
##
## Error: block:irrigation
##              Df Sum Sq Mean Sq F value Pr(>F)
## irrigation    1   8278    8278  17.59 0.0247 *
```

```
## Residuals    3    1412    471
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: block:irrigation:density
##              Df Sum Sq Mean Sq F value Pr(>F)
## density      2   1758   879.2    3.784 0.0532 .
## irrigation:density 2   2747  1373.5    5.912 0.0163 *
## Residuals    12   2788   232.3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: Within
##              Df Sum Sq Mean Sq F value    Pr(>F)
## fertilizer      2  1977.4   988.7   11.449 0.000142 ***
## irrigation:fertilizer 2   953.4   476.7    5.520 0.008108 **
## density:fertilizer    4   304.9    76.2    0.883 0.484053
## irrigation:density:fertilizer 4   234.7    58.7    0.680 0.610667
## Residuals           36  3108.8    86.4
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
interaction.plot(fertilizer, irrigation, yield)
```



```
interaction.plot(density, irrigation, yield)
```



```
detach(yields)
```

Variance component analysis

For random effects, we are often interested in how much of the variation in the response variable can be attributed to a given factor, this procedure is called **variance component analysis**.

```
rats <- read.table("rats.txt", header = TRUE)
attach(rats)
names(rats)
```

```
## [1] "Glycogen" "Treatment" "Rat" "Liver"
```

```
head(rats, 10) # a total of 6 rats used for 36 rows
```

```
##      Glycogen Treatment Rat Liver
## 1      131         1    1    1
## 2      130         1    1    1
## 3      131         1    1    2
## 4      125         1    1    2
## 5      136         1    1    3
## 6      142         1    1    3
## 7      150         1    2    1
## 8      148         1    2    1
## 9      140         1    2    2
## 10     143         1    2    2
```

```
Treatment <- factor(Treatment) # 3 treatment s
Rat <- factor(Rat)
Liver <- factor(Liver)
```

```
# analysis ignoring the pseudoreplication
```

```

model <- aov(Glycogen ~ Treatment)
summary(model) # show significant effects of Treatment but this is due to the pseudo replication

##              Df Sum Sq Mean Sq F value    Pr(>F)
## Treatment      2   1558    778.8    14.5 3.03e-05 ***
## Residuals     33   1773     53.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# df is wrong for residuals

# average away the pseudo replication
(means <- tapply(Glycogen, list(Treatment, Rat), mean))

##           1           2
## 1 132.5000 148.5000
## 2 149.6667 152.3333
## 3 134.3333 136.0000

(treat <- gl(3, 1, length = 6))

## [1] 1 2 3 1 2 3
## Levels: 1 2 3

# fit the non-pseudo replicated model
model <- aov(as.vector(means) ~ treat)
summary(model) # degree of freedom is correct now

##              Df Sum Sq Mean Sq F value Pr(>F)
## treat          2   259.6   129.80    2.929  0.197
## Residuals      3   132.9    44.31

# fit the correct analysis using aov with multiple error terms
# rats within treatment, liver within rat
model2 <- aov(Glycogen ~ Treatment + Error(Treatment/Rat/Liver))
summary(model2)

##
## Error: Treatment
##              Df Sum Sq Mean Sq
## Treatment      2   1558    778.8
##
## Error: Treatment:Rat
##              Df Sum Sq Mean Sq F value Pr(>F)
## Residuals     3   797.7   265.9
##
## Error: Treatment:Rat:Liver
##              Df Sum Sq Mean Sq F value Pr(>F)
## Residuals    12     594     49.5
##
## Error: Within
##              Df Sum Sq Mean Sq F value Pr(>F)
## Residuals    18     381     21.17

detach(rats)

```

Effect sizes in ANOVA: aov or lm ?

The difference between `lm` and `aov` is mainly in the form of the output from `summary`.

```
daphnia <- read.table("Daphnia.txt",header=T)
attach(daphnia)
names(daphnia)

## [1] "Growth.rate" "Water"          "Detergent"    "Daphnia"

# model using aov
modell1 <- aov(Growth.rate ~ Water * Detergent * Daphnia)
summary(modell1)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
## Water	1	1.99	1.985	2.850	0.097838 .
## Detergent	3	2.21	0.737	1.059	0.375478
## Daphnia	2	39.18	19.589	28.128	8.23e-09 ***
## Water:Detergent	3	0.17	0.058	0.084	0.968608
## Water:Daphnia	2	13.73	6.866	9.859	0.000259 ***
## Detergent:Daphnia	6	20.60	3.433	4.930	0.000532 ***
## Water:Detergent:Daphnia	6	5.85	0.975	1.399	0.234324
## Residuals	48	33.43	0.696		
## ---					
## Signif. codes:	0 '***'	0.001 '**'	0.01 '*'	0.05 '.'	0.1 ' ' 1

```
# model using lm
modell2 <- lm(Growth.rate ~ Water * Detergent * Daphnia)
summary(modell2)
```

```
##
## Call:
## lm(formula = Growth.rate ~ Water * Detergent * Daphnia)
##
## Residuals:
```

	Min	1Q	Median	3Q	Max
##	-1.4882	-0.5440	0.0239	0.3560	1.5250

```
##
## Coefficients:
```

	Estimate	Std. Error	t value
## (Intercept)	2.81126	0.48181	5.835
## WaterWear	-0.15808	0.68138	-0.232
## DetergentBrandB	-0.03536	0.68138	-0.052
## DetergentBrandC	0.47626	0.68138	0.699
## DetergentBrandD	-0.21407	0.68138	-0.314
## DaphniaClone2	0.49637	0.68138	0.728
## DaphniaClone3	2.05526	0.68138	3.016
## WaterWear:DetergentBrandB	0.46455	0.96361	0.482
## WaterWear:DetergentBrandC	-0.27431	0.96361	-0.285
## WaterWear:DetergentBrandD	0.21729	0.96361	0.225
## WaterWear:DaphniaClone2	1.38081	0.96361	1.433
## WaterWear:DaphniaClone3	0.43156	0.96361	0.448
## DetergentBrandB:DaphniaClone2	0.91892	0.96361	0.954
## DetergentBrandC:DaphniaClone2	-0.16337	0.96361	-0.170
## DetergentBrandD:DaphniaClone2	1.01209	0.96361	1.050
## DetergentBrandB:DaphniaClone3	-0.06490	0.96361	-0.067

```
## DetergentBrandC:DaphniaClone3      -0.80789    0.96361   -0.838
## DetergentBrandD:DaphniaClone3      -1.28669    0.96361   -1.335
## WaterWear:DetergentBrandB:DaphniaClone2 -1.26380    1.36275   -0.927
## WaterWear:DetergentBrandC:DaphniaClone2  1.35612    1.36275    0.995
## WaterWear:DetergentBrandD:DaphniaClone2  0.77616    1.36275    0.570
## WaterWear:DetergentBrandB:DaphniaClone3 -0.87443    1.36275   -0.642
## WaterWear:DetergentBrandC:DaphniaClone3 -1.03019    1.36275   -0.756
## WaterWear:DetergentBrandD:DaphniaClone3 -1.55400    1.36275   -1.140
##                                     Pr(>|t|)
## (Intercept)                        4.48e-07 ***
## WaterWear                          0.81753
## DetergentBrandB                     0.95883
## DetergentBrandC                     0.48794
## DetergentBrandD                     0.75475
## DaphniaClone2                       0.46986
## DaphniaClone3                       0.00408 **
## WaterWear:DetergentBrandB           0.63193
## WaterWear:DetergentBrandC           0.77712
## WaterWear:DetergentBrandD           0.82255
## WaterWear:DaphniaClone2             0.15835
## WaterWear:DaphniaClone3             0.65627
## DetergentBrandB:DaphniaClone2        0.34506
## DetergentBrandC:DaphniaClone2        0.86609
## DetergentBrandD:DaphniaClone2        0.29884
## DetergentBrandB:DaphniaClone3        0.94658
## DetergentBrandC:DaphniaClone3        0.40597
## DetergentBrandD:DaphniaClone3        0.18809
## WaterWear:DetergentBrandB:DaphniaClone2 0.35837
## WaterWear:DetergentBrandC:DaphniaClone2 0.32466
## WaterWear:DetergentBrandD:DaphniaClone2 0.57164
## WaterWear:DetergentBrandB:DaphniaClone3 0.52414
## WaterWear:DetergentBrandC:DaphniaClone3 0.45337
## WaterWear:DetergentBrandD:DaphniaClone3 0.25980
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8345 on 48 degrees of freedom
## Multiple R-squared:  0.7147, Adjusted R-squared:  0.578
## F-statistic: 5.227 on 23 and 48 DF, p-value: 7.019e-07
```

```
summary.lm(model1)
```

```
##
## Call:
## aov(formula = Growth.rate ~ Water * Detergent * Daphnia)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.4882 -0.5440  0.0239  0.3560  1.5250
##
## Coefficients:
##                                     Estimate Std. Error t value
## (Intercept)                        2.81126    0.48181   5.835
## WaterWear                          -0.15808    0.68138  -0.232
## DetergentBrandB                     -0.03536    0.68138  -0.052
```

```

## DetergentBrandC          0.47626    0.68138    0.699
## DetergentBrandD         -0.21407    0.68138   -0.314
## DaphniaClone2           0.49637    0.68138    0.728
## DaphniaClone3           2.05526    0.68138    3.016
## WaterWear:DetergentBrandB 0.46455    0.96361    0.482
## WaterWear:DetergentBrandC -0.27431    0.96361   -0.285
## WaterWear:DetergentBrandD 0.21729    0.96361    0.225
## WaterWear:DaphniaClone2   1.38081    0.96361    1.433
## WaterWear:DaphniaClone3   0.43156    0.96361    0.448
## DetergentBrandB:DaphniaClone2 0.91892    0.96361    0.954
## DetergentBrandC:DaphniaClone2 -0.16337    0.96361   -0.170
## DetergentBrandD:DaphniaClone2 1.01209    0.96361    1.050
## DetergentBrandB:DaphniaClone3 -0.06490    0.96361   -0.067
## DetergentBrandC:DaphniaClone3 -0.80789    0.96361   -0.838
## DetergentBrandD:DaphniaClone3 -1.28669    0.96361   -1.335
## WaterWear:DetergentBrandB:DaphniaClone2 -1.26380    1.36275   -0.927
## WaterWear:DetergentBrandC:DaphniaClone2 1.35612    1.36275    0.995
## WaterWear:DetergentBrandD:DaphniaClone2 0.77616    1.36275    0.570
## WaterWear:DetergentBrandB:DaphniaClone3 -0.87443    1.36275   -0.642
## WaterWear:DetergentBrandC:DaphniaClone3 -1.03019    1.36275   -0.756
## WaterWear:DetergentBrandD:DaphniaClone3 -1.55400    1.36275   -1.140
## Pr(>|t|)
## (Intercept)             4.48e-07 ***
## WaterWear                0.81753
## DetergentBrandB          0.95883
## DetergentBrandC          0.48794
## DetergentBrandD          0.75475
## DaphniaClone2            0.46986
## DaphniaClone3            0.00408 **
## WaterWear:DetergentBrandB 0.63193
## WaterWear:DetergentBrandC 0.77712
## WaterWear:DetergentBrandD 0.82255
## WaterWear:DaphniaClone2  0.15835
## WaterWear:DaphniaClone3  0.65627
## DetergentBrandB:DaphniaClone2 0.34506
## DetergentBrandC:DaphniaClone2 0.86609
## DetergentBrandD:DaphniaClone2 0.29884
## DetergentBrandB:DaphniaClone3 0.94658
## DetergentBrandC:DaphniaClone3 0.40597
## DetergentBrandD:DaphniaClone3 0.18809
## WaterWear:DetergentBrandB:DaphniaClone2 0.35837
## WaterWear:DetergentBrandC:DaphniaClone2 0.32466
## WaterWear:DetergentBrandD:DaphniaClone2 0.57164
## WaterWear:DetergentBrandB:DaphniaClone3 0.52414
## WaterWear:DetergentBrandC:DaphniaClone3 0.45337
## WaterWear:DetergentBrandD:DaphniaClone3 0.25980
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8345 on 48 degrees of freedom
## Multiple R-squared:  0.7147, Adjusted R-squared:  0.578
## F-statistic: 5.227 on 23 and 48 DF, p-value: 7.019e-07

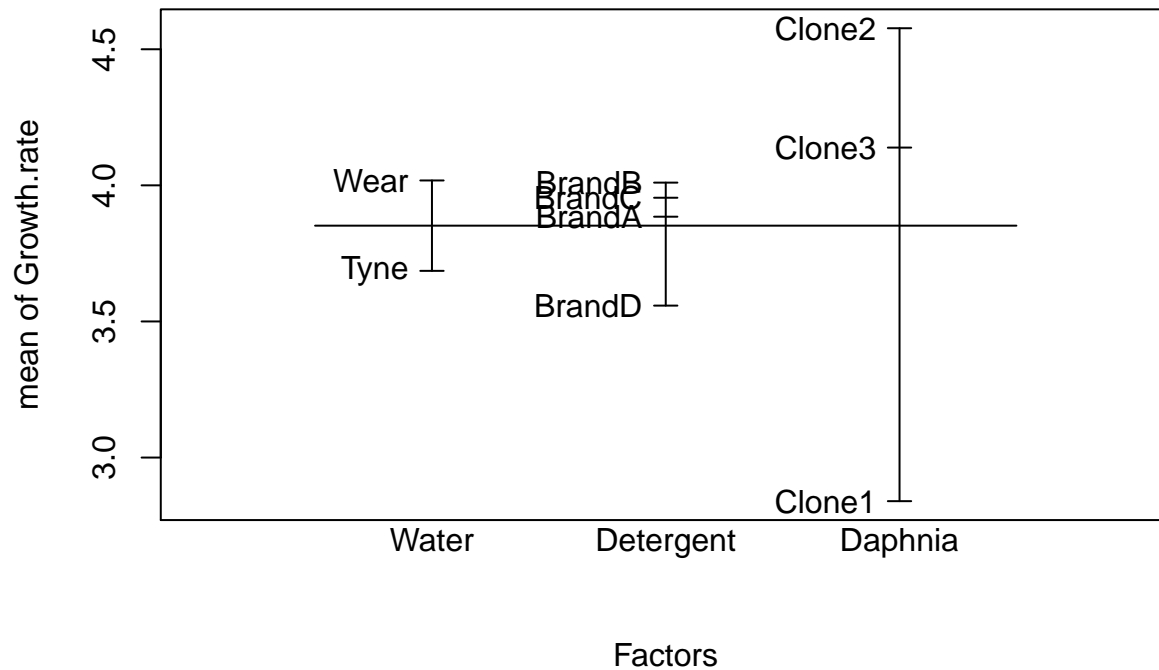
```

```
summary.aov(model2)
```

```
##              Df Sum Sq Mean Sq F value    Pr(>F)
## Water          1   1.99   1.985    2.850 0.097838 .
## Detergent       3   2.21   0.737    1.059 0.375478
## Daphnia         2  39.18  19.589   28.128 8.23e-09 ***
## Water:Detergent  3   0.17   0.058    0.084 0.968608
## Water:Daphnia    2  13.73   6.866    9.859 0.000259 ***
## Detergent:Daphnia 6  20.60   3.433    4.930 0.000532 ***
## Water:Detergent:Daphnia 6   5.85   0.975    1.399 0.234324
## Residuals      48  33.43   0.696
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# summarize the effect sizes
```

```
plot.design(Growth.rate ~ Water * Detergent * Daphnia)
```



```
model.tables(model1, "means", se = TRUE)
```

```
## Tables of means
## Grand mean
##
## 3.851905
##
## Water
## Water
## Tyne Wear
## 3.686 4.018
##
## Detergent
## Detergent
## BrandA BrandB BrandC BrandD
## 3.885 4.010 3.955 3.558
```



```

##
## Daphnia
## Daphnia
## Clone1 Clone2 Clone3
## 2.840 4.577 4.139
##
## Water:Detergent
## Detergent
## Water BrandA BrandB BrandC BrandD
## Tyne 3.662 3.911 3.814 3.356
## Wear 4.108 4.109 4.095 3.760
##
## Water:Daphnia
## Daphnia
## Water Clone1 Clone2 Clone3
## Tyne 2.868 3.806 4.383
## Wear 2.812 5.348 3.894
##
## Detergent:Daphnia
## Daphnia
## Detergent Clone1 Clone2 Clone3
## BrandA 2.732 3.919 5.003
## BrandB 2.929 4.403 4.698
## BrandC 3.071 4.773 4.019
## BrandD 2.627 5.214 2.834
##
## Water:Detergent:Daphnia
## , , Daphnia = Clone1
##
## Detergent
## Water BrandA BrandB BrandC BrandD
## Tyne 2.811 2.776 3.288 2.597
## Wear 2.653 3.082 2.855 2.656
##
## , , Daphnia = Clone2
##
## Detergent
## Water BrandA BrandB BrandC BrandD
## Tyne 3.308 4.191 3.621 4.106
## Wear 4.530 4.615 5.925 6.322
##
## , , Daphnia = Clone3
##
## Detergent
## Water BrandA BrandB BrandC BrandD
## Tyne 4.867 4.766 4.535 3.366
## Wear 5.140 4.630 3.504 2.303
##
##
## Standard errors for differences of means
## Water Detergent Daphnia Water:Detergent Water:Daphnia
## 0.1967 0.2782 0.2409 0.3934 0.3407
## replic. 36 18 24 9 12
## Detergent:Daphnia Water:Detergent:Daphnia

```

```
##                0.4818                0.6814
## replic.        6                3
detach(daphnia)
```

Multiple comparisons

- TukeyHSD for Tukey's honest significant differences
- pairwise.t.test for adjusted p values for all comparisons

```
data <- read.table("Fungi.txt", header = TRUE)
attach(data)
names(data)
```

```
## [1] "Habitat"      "Fugus.yield"
# test whether there is any variation in fungus yield to explain
model <- aov(Fugus.yield ~ Habitat)
summary(model)
```

```
##                Df Sum Sq Mean Sq F value Pr(>F)
## Habitat        15   7527    501.8   72.14 <2e-16 ***
## Residuals     144   1002     7.0
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Tukey's test for p values
head(TukeyHSD(model)[[1]])
```

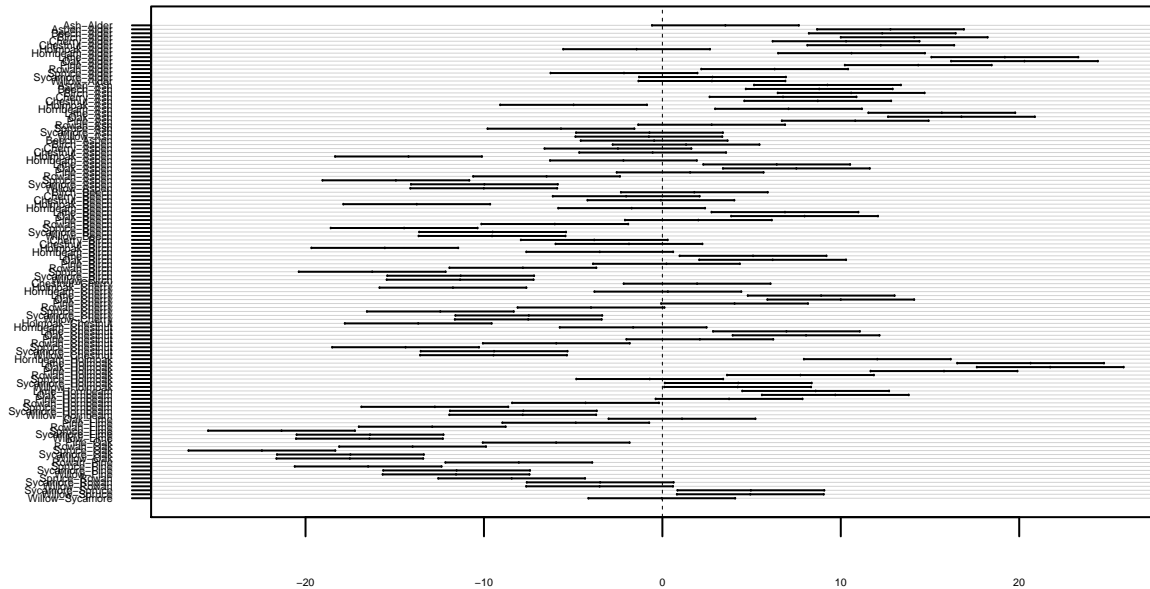
```
##                diff          lwr          upr          p adj
## Ash-Alder       3.532928 -0.5808096  7.646665 1.844088e-01
## Aspen-Alder     12.785744  8.6720067 16.899481 3.208545e-14
## Beech-Alder     12.323653  8.2099161 16.437391 6.050715e-14
## Birch-Alder     14.113482  9.9997441 18.227219 0.000000e+00
## Cherry-Alder    10.295088  6.1813503 14.408825 7.818191e-13
## Chestnut-Alder  12.241079  8.1273416 16.354816 6.050715e-14
```

```
str(TukeyHSD(model))
```

```
## List of 1
## $ Habitat: num [1:120, 1:4] 3.53 12.79 12.32 14.11 10.3 ...
## .. attr(*, "dimnames")=List of 2
## .. ..$ : chr [1:120] "Ash-Alder" "Aspen-Alder" "Beech-Alder" "Birch-Alder" ...
## .. ..$ : chr [1:4] "diff" "lwr" "upr" "p adj"
## - attr(*, "class")= chr [1:2] "TukeyHSD" "multicomp"
## - attr(*, "orig.call")= language aov(formula = Fugus.yield ~ Habitat)
## - attr(*, "conf.level")= num 0.95
## - attr(*, "ordered")= logi FALSE
```

```
# or plot it
plot(TukeyHSD(model), las = 1, cex.axis = 0.3)
```

95% family-wise confidence level



Differences in mean levels of Habitat

```
# use pairwise.t.test
pairwise.t.test(Fugus.yield, Habitat)
```

```
##
## Pairwise comparisons using t tests with pooled SD
##
## data: Fugus.yield and Habitat
##
##      Alder  Ash   Aspen  Beech  Birch  Cherry  Chestnut  Holmoak
## Ash      0.10011 -      -      -      -      -      -      -
## Aspen    < 2e-16 6.3e-11 -      -      -      -      -      -
## Beech    < 2e-16 5.4e-10 1.000000 -      -      -      -      -
## Birch    < 2e-16 1.2e-13 1.000000 1.000000 -      -      -      -
## Cherry   4.7e-13 2.9e-06 0.87474 1.000000 0.04943 -      -      -
## Chestnut < 2e-16 7.8e-10 1.000000 1.000000 1.000000 1.000000 -      -
## Holmoak  1.000000 0.00181 < 2e-16 < 2e-16 < 2e-16 3.9e-16 < 2e-16 -
## Hornbeam 1.1e-13 8.6e-07 1.000000 1.000000 0.10057 1.000000 1.000000 < 2e-16
## Lime     < 2e-16 < 2e-16 1.1e-05 1.9e-06 0.00131 3.3e-10 1.4e-06 < 2e-16
## Oak      < 2e-16 < 2e-16 1.4e-07 2.0e-08 2.7e-05 1.9e-12 1.5e-08 < 2e-16
## Pine     < 2e-16 3.9e-14 1.000000 1.000000 1.000000 0.02757 1.000000 < 2e-16
## Rowan    1.8e-05 0.51826 8.5e-06 4.7e-05 3.9e-08 0.03053 6.2e-05 5.3e-08
## Spruce   1.000000 0.00016 < 2e-16 < 2e-16 < 2e-16 < 2e-16 < 2e-16 1.000000
## Sycamore 0.50084 1.000000 2.1e-12 1.9e-11 3.3e-15 1.5e-07 2.7e-11 0.01586
## Willow   0.51826 1.000000 1.9e-12 1.6e-11 2.8e-15 1.4e-07 2.4e-11 0.01702
##
##      Hornbeam  Lime   Oak   Pine   Rowan   Spruce   Sycamore
## Ash          -      -      -      -      -      -      -
## Aspen        -      -      -      -      -      -      -
## Beech        -      -      -      -      -      -      -
## Birch        -      -      -      -      -      -      -
```

```

## Cherry - - - - -
## Chestnut - - - - -
## Holmoak - - - - -
## Hornbeam - - - - -
## Lime 1.3e-09 - - - - -
## Oak 8.4e-12 1.00000 - - - - -
## Pine 0.05975 0.00253 6.1e-05 - - - - -
## Rowan 0.01380 < 2e-16 < 2e-16 1.5e-08 - - - - -
## Spruce < 2e-16 < 2e-16 < 2e-16 < 2e-16 2.5e-09 - - - - -
## Sycamore 4.2e-08 < 2e-16 < 2e-16 1.1e-15 0.10218 0.00187 - - - - -
## Willow 3.8e-08 < 2e-16 < 2e-16 9.3e-16 0.10057 0.00203 1.00000
##
## P value adjustment method: holm
# try other adjusted method
pairwise.t.test(Fugus.yield, Habitat, p.adjust.method = "none")

##
## Pairwise comparisons using t tests with pooled SD
##
## data: Fugus.yield and Habitat
##
## Alder Ash Aspen Beech Birch Cherry Chestnut Holmoak
## Ash 0.00323 - - - - -
## Aspen < 2e-16 8.9e-13 - - - - -
## Beech < 2e-16 7.8e-12 0.69581 - - - - -
## Birch < 2e-16 1.4e-15 0.26218 0.13135 - - - - -
## Cherry 5.9e-15 5.6e-08 0.03645 0.08761 0.00150 - - - - -
## Chestnut < 2e-16 1.1e-11 0.64494 0.94428 0.11461 0.10116 - - - - -
## Holmoak 0.22299 4.3e-05 < 2e-16 < 2e-16 < 2e-16 < 2e-16 < 2e-16 -
## Hornbeam 1.3e-15 1.6e-08 0.06625 0.14673 0.00342 0.79462 0.16697 < 2e-16
## Lime < 2e-16 < 2e-16 2.3e-07 3.6e-08 3.0e-05 4.7e-12 2.5e-08 < 2e-16
## Oak < 2e-16 < 2e-16 2.4e-09 3.2e-10 5.6e-07 2.5e-14 2.2e-10 < 2e-16
## Pine < 2e-16 4.7e-16 0.18945 0.08938 0.84742 0.00079 0.07715 < 2e-16
## Rowan 3.6e-07 0.02057 1.7e-07 1.0e-06 6.3e-10 0.00090 1.4e-06 9.0e-10
## Spruce 0.07026 3.6e-06 < 2e-16 < 2e-16 < 2e-16 < 2e-16 < 2e-16 0.54952
## Sycamore 0.01855 0.54035 2.8e-14 2.6e-13 < 2e-16 2.7e-09 3.8e-13 0.00043
## Willow 0.01993 0.52226 2.4e-14 2.2e-13 < 2e-16 2.3e-09 3.3e-13 0.00047
## Hornbeam Lime Oak Pine Rowan Spruce Sycamore
## Ash - - - - -
## Aspen - - - - -
## Beech - - - - -
## Birch - - - - -
## Cherry - - - - -
## Chestnut - - - - -
## Holmoak - - - - -
## Hornbeam - - - - -
## Lime 2.0e-11 - - - - -
## Oak 1.1e-13 0.35260 - - - - -
## Pine 0.00187 6.5e-05 1.3e-06 - - - - -
## Rowan 0.00036 < 2e-16 < 2e-16 2.3e-10 - - - - -
## Spruce < 2e-16 < 2e-16 < 2e-16 < 2e-16 3.8e-11 - - - - -
## Sycamore 7.1e-10 < 2e-16 < 2e-16 < 2e-16 0.00365 4.6e-05 - - - - -
## Willow 6.1e-10 < 2e-16 < 2e-16 < 2e-16 0.00335 5.1e-05 0.97796
##

```

```
## P value adjustment method: none
# a package for multiple comparisons
# install.packages("multcomp")
detach(data)
```

Multivariate analysis of variance using manova

```
data <- read.table("manova.txt", header = TRUE)
attach(data)
names(data)

## [1] "tear"      "gloss"      "opacity"    "rate"      "additive"
# three response variables
Y <- cbind(tear, gloss, opacity)

model <- manova(Y ~ rate * additive)

summary(model)

##              Df  Pillai approx F num Df den Df    Pr(>F)
## rate          1 0.61814   7.5543     3    14 0.003034 **
## additive      1 0.47697   4.2556     3    14 0.024745 *
## rate:additive 1 0.22289   1.3385     3    14 0.301782
## Residuals     16
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
# look at each of the three response variables separately
summary.aov(model)

## Response tear :
##              Df Sum Sq Mean Sq F value    Pr(>F)
## rate          1 1.7405 1.74050 15.7868 0.001092 **
## additive      1 0.7605 0.76050  6.8980 0.018330 *
## rate:additive 1 0.0005 0.00050  0.0045 0.947143
## Residuals     16 1.7640 0.11025
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Response gloss :
##              Df Sum Sq Mean Sq F value    Pr(>F)
## rate          1 1.3005 1.30050  7.9178 0.01248 *
## additive      1 0.6125 0.61250  3.7291 0.07139 .
## rate:additive 1 0.5445 0.54450  3.3151 0.08740 .
## Residuals     16 2.6280 0.16425
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Response opacity :
##              Df Sum Sq Mean Sq F value    Pr(>F)
## rate          1  0.421  0.4205  0.1036 0.7517
## additive      1  4.901  4.9005  1.2077 0.2881
## rate:additive 1  3.960  3.9605  0.9760 0.3379
```

```
## Residuals      16 64.924  4.0578
summary.manova(model)

##              Df  Pillai approx F num Df den Df    Pr(>F)
## rate          1 0.61814   7.5543     3    14 0.003034 **
## additive      1 0.47697   4.2556     3    14 0.024745 *
## rate:additive  1 0.22289   1.3385     3    14 0.301782
## Residuals     16
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

detach(data)
```

Chapter 12 Analysis of Covariance

The response variable is continuous, and there is at least one continuous explanatory variable and at least one categorical explanatory variable.

```
regrowth <- read.table("ipomopsis.txt", header = TRUE)
attach(regrowth)
names(regrowth)

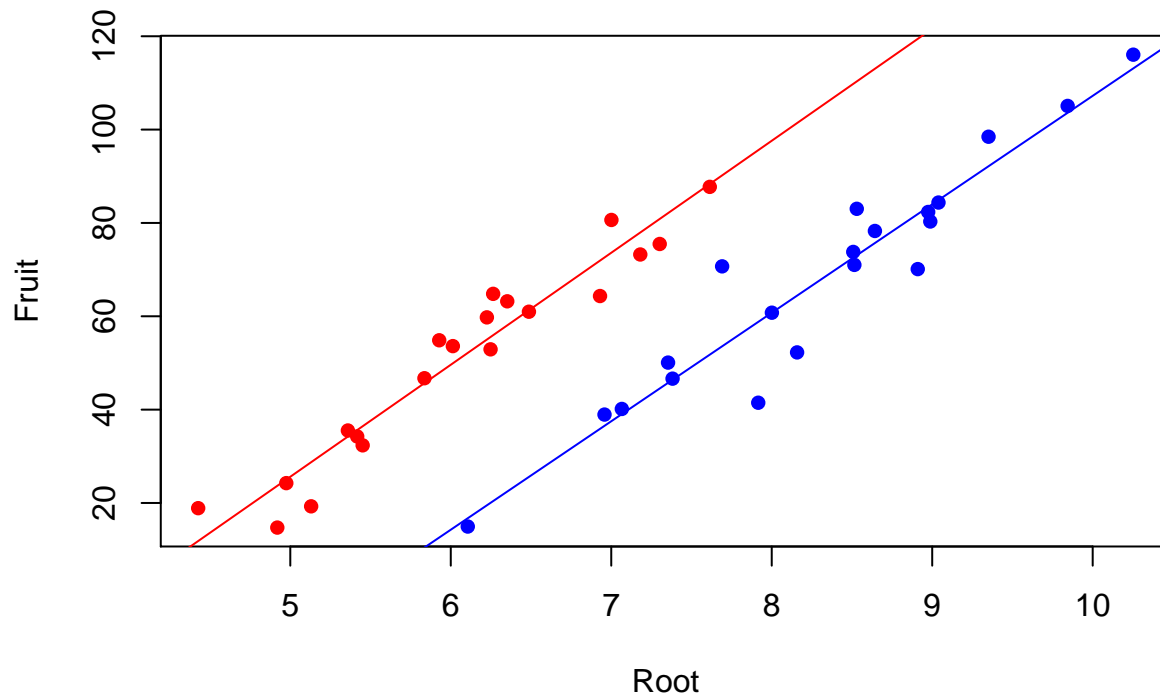
## [1] "Root"      "Fruit"     "Grazing"
# plot Fruit ~ Root, with different colors for different Grazing
as.numeric(Grazing) # 2, 1

## [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1
## [36] 1 1 1 1 1

plot(Root, Fruit, pch = 16, col = c("blue", "red")[as.numeric(Grazing)])
levels(Grazing)

## [1] "Grazed"    "Ungrazed"
```

```
abline(lm(Fruit[Grazing == "Grazed"] ~ Root[Grazing == "Grazed"]), col = "blue")
abline(lm(Fruit[Grazing == "Ungrazed"] ~ Root[Grazing == "Ungrazed"]), col = "red")
```



```
tapply(Fruit, Grazing, mean)
```

```
##   Grazed Ungrazed
##  67.9405  50.8805
```

```
t.test(Fruit ~ Grazing)
```

```
##
##  Welch Two Sample t-test
##
## data:  Fruit by Grazing
## t = 2.304, df = 37.306, p-value = 0.02689
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   2.061464 32.058536
## sample estimates:
##   mean in group Grazed mean in group Ungrazed
##               67.9405               50.8805
```

```
# fit different slopes and intercepts
ancova <- lm(Fruit ~ Grazing * Root)
```

```
summary(ancova)
```

```
##
## Call:
## lm(formula = Fruit ~ Grazing * Root)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.3177  -2.8320   0.1247   3.8511  17.1313
##
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -125.173     12.811  -9.771 1.15e-11 ***
## GrazingUngrazed    30.806     16.842   1.829  0.0757 .
## Root            23.240      1.531  15.182 < 2e-16 ***
## GrazingUngrazed:Root    0.756      2.354   0.321  0.7500
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.831 on 36 degrees of freedom
## Multiple R-squared:  0.9293, Adjusted R-squared:  0.9234
## F-statistic: 157.6 on 3 and 36 DF,  p-value: < 2.2e-16
```

```
anova(ancova)
```

```
## Analysis of Variance Table
##
## Response: Fruit
##              Df Sum Sq Mean Sq F value    Pr(>F)
## Grazing         1  2910.4   2910.4  62.3795 2.262e-09 ***
## Root            1 19148.9  19148.9 410.4201 < 2.2e-16 ***
## Grazing:Root    1     4.8     4.8   0.1031    0.75
## Residuals      36  1679.6    46.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# a simplified model
```

```
ancova2 <- update(ancova, ~ . - Grazing:Root)
```

```
# compare the two models
```

```
anova(ancova, ancova2)
```

```
## Analysis of Variance Table
##
## Model 1: Fruit ~ Grazing * Root
## Model 2: Fruit ~ Grazing + Root
##   Res.Df    RSS Df Sum of Sq    F Pr(>F)
## 1      36 1679.7
## 2      37 1684.5 -1    -4.8122 0.1031  0.75
```

```
# check if the Grazing effect is significant or not
```

```
ancova3 <- update(ancova2, ~ . - Grazing)
```

```
anova(ancova2, ancova3) # it's significant
```

```
## Analysis of Variance Table
##
## Model 1: Fruit ~ Grazing + Root
## Model 2: Fruit ~ Root
##   Res.Df    RSS Df Sum of Sq    F    Pr(>F)
## 1      37 1684.5
## 2      38 6948.8 -1   -5264.4 115.63 6.107e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(ancova2)
```

```
##
```



```
## Call:
## lm(formula = Fruit ~ Grazing + Root)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.1920  -2.8224   0.3223   3.9144  17.3290
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -127.829     9.664  -13.23 1.35e-15 ***
## GrazingUngrazed    36.103     3.357   10.75 6.11e-13 ***
## Root           23.560     1.149   20.51 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.747 on 37 degrees of freedom
## Multiple R-squared:  0.9291, Adjusted R-squared:  0.9252
## F-statistic: 242.3 on 2 and 37 DF,  p-value: < 2.2e-16
# use "step" starts with the full model and do simplification based on AIC
step(ancova)

## Start:  AIC=157.5
## Fruit ~ Grazing * Root
##
##              Df Sum of Sq    RSS    AIC
## - Grazing:Root  1     4.8122 1684.5 155.61
## <none>                        1679.7 157.50
##
## Step:  AIC=155.61
## Fruit ~ Grazing + Root
##
##              Df Sum of Sq    RSS    AIC
## <none>                        1684.5 155.61
## - Grazing    1     5264.4  6948.8 210.30
## - Root       1    19148.9 20833.4 254.22
##
## Call:
## lm(formula = Fruit ~ Grazing + Root)
##
## Coefficients:
##      (Intercept)  GrazingUngrazed          Root
##          -127.83           36.10           23.56
detach(regrowth)
```

ANCOVA with two factors and one continuous covariate

```
Gain <- read.table("Gain.txt",header=T)
attach(Gain)
names(Gain)

## [1] "Weight"  "Sex"     "Age"     "Genotype" "Score"
```

```
# maximal model with 24 parameters
m1 <- lm(Weight ~ Sex * Age * Genotype)
summary(m1)
```

```
##
## Call:
## lm(formula = Weight ~ Sex * Age * Genotype)
##
## Residuals:
```

	Min	1Q	Median	3Q	Max
	-0.40218	-0.12043	-0.01065	0.12592	0.44687

```
##
## Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	7.80053	0.24941	31.276	< 2e-16 ***
Sexmale	-0.51966	0.35272	-1.473	0.14936
Age	0.34950	0.07520	4.648	4.39e-05 ***
GenotypeCloneB	1.19870	0.35272	3.398	0.00167 **
GenotypeCloneC	-0.41751	0.35272	-1.184	0.24429
GenotypeCloneD	0.95600	0.35272	2.710	0.01023 *
GenotypeCloneE	-0.81604	0.35272	-2.314	0.02651 *
GenotypeCloneF	1.66851	0.35272	4.730	3.41e-05 ***
Sexmale:Age	-0.11283	0.10635	-1.061	0.29579
Sexmale:GenotypeCloneB	-0.31716	0.49882	-0.636	0.52891
Sexmale:GenotypeCloneC	-1.06234	0.49882	-2.130	0.04010 *
Sexmale:GenotypeCloneD	-0.73547	0.49882	-1.474	0.14906
Sexmale:GenotypeCloneE	-0.28533	0.49882	-0.572	0.57087
Sexmale:GenotypeCloneF	-0.19839	0.49882	-0.398	0.69319
Age:GenotypeCloneB	-0.10146	0.10635	-0.954	0.34643
Age:GenotypeCloneC	-0.20825	0.10635	-1.958	0.05799 .
Age:GenotypeCloneD	-0.01757	0.10635	-0.165	0.86970
Age:GenotypeCloneE	-0.03825	0.10635	-0.360	0.72123
Age:GenotypeCloneF	-0.05512	0.10635	-0.518	0.60743
Sexmale:Age:GenotypeCloneB	0.15469	0.15040	1.029	0.31055
Sexmale:Age:GenotypeCloneC	0.35322	0.15040	2.349	0.02446 *
Sexmale:Age:GenotypeCloneD	0.19227	0.15040	1.278	0.20929
Sexmale:Age:GenotypeCloneE	0.13203	0.15040	0.878	0.38585
Sexmale:Age:GenotypeCloneF	0.08709	0.15040	0.579	0.56616

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2378 on 36 degrees of freedom
## Multiple R-squared:  0.9742, Adjusted R-squared:  0.9577
## F-statistic: 59.06 on 23 and 36 DF,  p-value: < 2.2e-16
```

```
# model simplification
m2 <- step(m1)
```

```
## Start:  AIC=-155.01
## Weight ~ Sex * Age * Genotype
##
##           Df Sum of Sq  RSS    AIC
## - Sex:Age:Genotype  5   0.34912 2.3849 -155.51
## <none>                2.0358 -155.01
##
```

```
## Step: AIC=-155.51
## Weight ~ Sex + Age + Genotype + Sex:Age + Sex:Genotype + Age:Genotype
##
##           Df Sum of Sq   RSS   AIC
## - Sex:Genotype  5  0.146901 2.5318 -161.92
## - Age:Genotype  5  0.168136 2.5531 -161.42
## - Sex:Age       1  0.048937 2.4339 -156.29
## <none>                  2.3849 -155.51
##
## Step: AIC=-161.92
## Weight ~ Sex + Age + Genotype + Sex:Age + Age:Genotype
##
##           Df Sum of Sq   RSS   AIC
## - Age:Genotype  5  0.168136 2.7000 -168.07
## - Sex:Age       1  0.048937 2.5808 -162.78
## <none>                  2.5318 -161.92
##
## Step: AIC=-168.07
## Weight ~ Sex + Age + Genotype + Sex:Age
##
##           Df Sum of Sq   RSS   AIC
## - Sex:Age      1      0.049   2.749 -168.989
## <none>                  2.700 -168.066
## - Genotype     5   54.958 57.658   5.612
##
## Step: AIC=-168.99
## Weight ~ Sex + Age + Genotype
##
##           Df Sum of Sq   RSS   AIC
## <none>                  2.749 -168.989
## - Sex          1   10.374 13.122  -77.201
## - Age          1   10.770 13.519  -75.415
## - Genotype     5   54.958 57.707   3.662
```

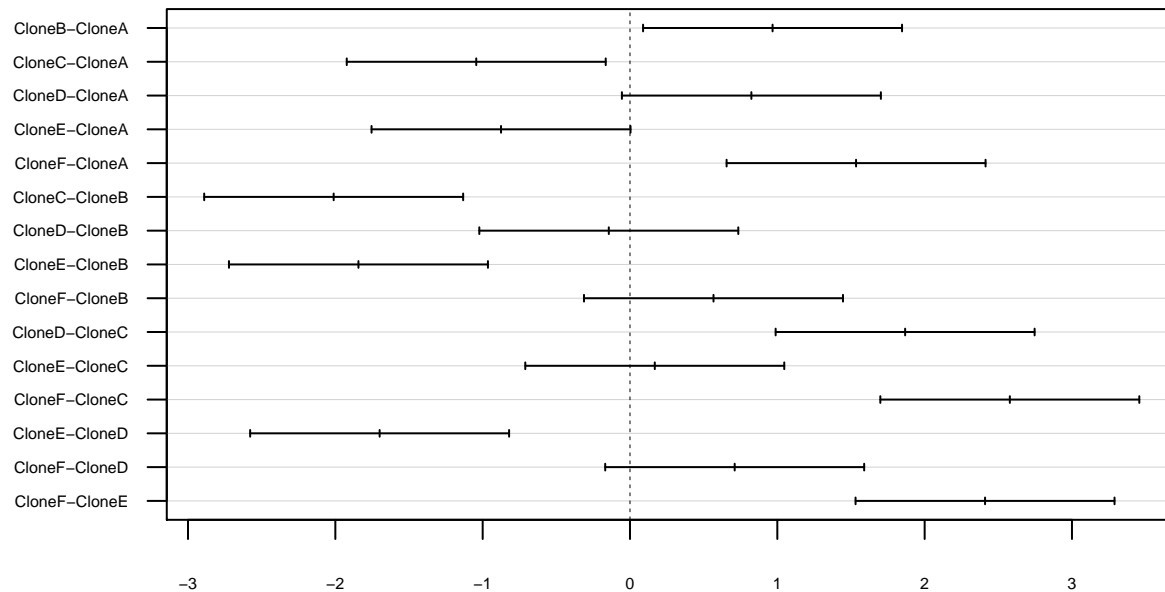
```
summary(m2)
```

```
##
## Call:
## lm(formula = Weight ~ Sex + Age + Genotype)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.40005 -0.15120 -0.01668  0.16953  0.49227
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    7.93701    0.10066   78.851 < 2e-16 ***
## Sexmale       -0.83161    0.05937  -14.008 < 2e-16 ***
## Age           0.29958    0.02099   14.273 < 2e-16 ***
## GenotypeCloneB 0.96778    0.10282    9.412 8.07e-13 ***
## GenotypeCloneC -1.04361    0.10282  -10.149 6.21e-14 ***
## GenotypeCloneD 0.82396    0.10282    8.013 1.21e-10 ***
## GenotypeCloneE -0.87540    0.10282   -8.514 1.98e-11 ***
## GenotypeCloneF 1.53460    0.10282   14.925 < 2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2299 on 52 degrees of freedom
## Multiple R-squared:  0.9651, Adjusted R-squared:  0.9604
## F-statistic: 205.7 on 7 and 52 DF,  p-value: < 2.2e-16
```

```
test <- aov(Weight ~ Genotype)
plot(TukeyHSD(test), las = 1, cex.axis = 0.5)
```

95% family-wise confidence level



Differences in mean levels of Genotype

```
# further simplification by combining factors levels
newGenotype <- Genotype
levels(newGenotype)
```

```
## [1] "CloneA" "CloneB" "CloneC" "CloneD" "CloneE" "CloneF"
```

```
# overwrite the original levels
levels(newGenotype)[c(3, 5)] <- "ClonesCandE"
levels(newGenotype)[c(2, 4)] <- "ClonesBandD"
levels(newGenotype)
```

```
## [1] "CloneA"      "ClonesBandD" "ClonesCandE" "CloneF"
```

```
# four levels for genotype now
```

```
# fit the new model
m3 <- lm(Weight ~ Sex + Age + newGenotype)
anova(m2, m3)
```

```
## Analysis of Variance Table
##
```

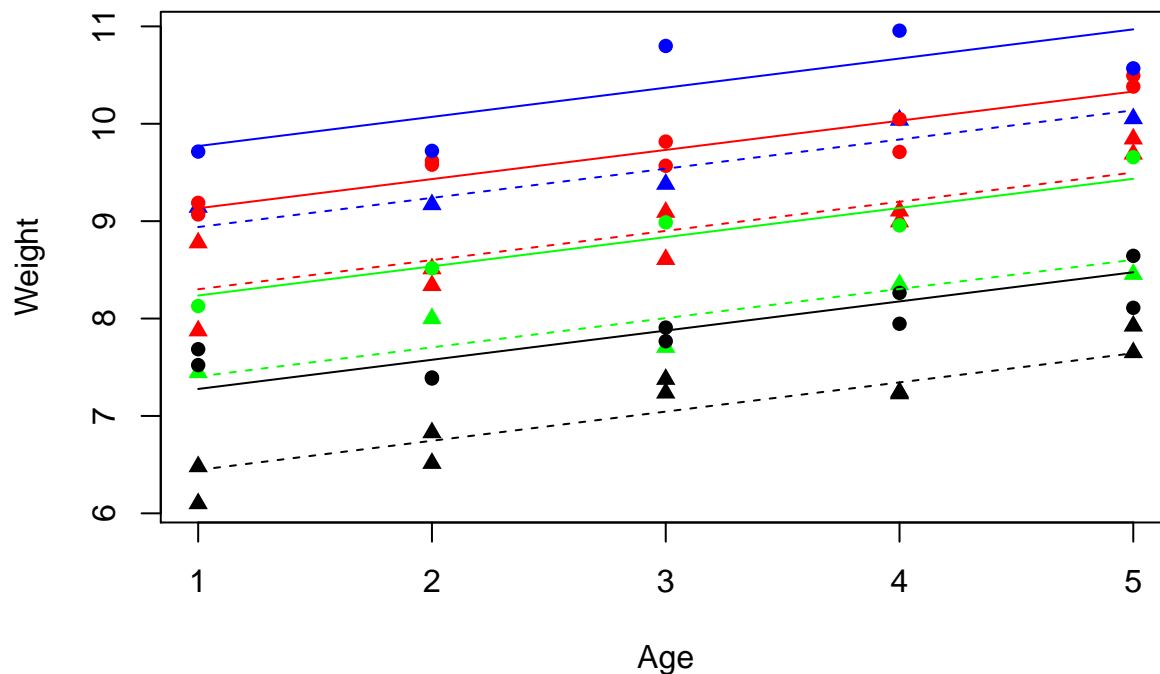
```
## Model 1: Weight ~ Sex + Age + Genotype
## Model 2: Weight ~ Sex + Age + newGenotype
##   Res.Df    RSS Df Sum of Sq    F Pr(>F)
## 1      52 2.7489
## 2      54 2.9938 -2   -0.24489 2.3163 0.1087
```

```
summary(m3) # m3 preferred
```

```
##
## Call:
## lm(formula = Weight ~ Sex + Age + newGenotype)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.42651 -0.16687  0.01211  0.18776  0.47736
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      7.93701     0.10308  76.996 < 2e-16 ***
## Sexmale          -0.83161     0.06080 -13.679 < 2e-16 ***
## Age              0.29958     0.02149  13.938 < 2e-16 ***
## newGenotypeClonesBandD 0.89587     0.09119   9.824 1.28e-13 ***
## newGenotypeClonesCandE -0.95950     0.09119 -10.522 1.10e-14 ***
## newGenotypeCloneF      1.53460     0.10530  14.574 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2355 on 54 degrees of freedom
## Multiple R-squared:  0.962, Adjusted R-squared:  0.9585
## F-statistic: 273.7 on 5 and 54 DF,  p-value: < 2.2e-16
```

```
# draw fitted lines through a scatterplot
```

```
plot(Age, Weight, type = "n")
colours <- c("green", "red", "black", "blue")
lines <- c(1, 2)
symbols <- c(16, 17)
points(Age, Weight, pch = symbols[as.numeric(Sex)], col=colours[as.numeric(newGenotype)])
xv <- c(1, 5)
for (i in 1:2) {
  for (j in 1:4) {
    a <- coef(m3)[1] + (i > 1) * coef(m3)[2] + (j > 1)*coef(m3)[j + 2]
    b <- coef(m3)[3]
    yv <- a + b*xv
    lines(xv, yv, lty = lines[i], col = colours[j]) }
}
```



```
detach(Gain)
```

Contrasts and the parameters of ANCOVA models

```
Ancovacontrasts <- read.table("Ancovacontrasts.txt", header = TRUE)
attach(Ancovacontrasts)
names(Ancovacontrasts)
```

```
## [1] "weight" "sex" "age"
```

```
tapply(weight, list(sex, age), mean)
```

```
##           1           2           3           4           5           6           7
## female 3.720102 3.386025 4.883829 6.670705 5.791259 8.167242 8.487023
## male   5.311580 7.340586 6.561460 10.151011 9.976946 12.076808 13.341954
##           8           9          10
## female 10.17159 10.64643 12.52977
## male   14.39329 17.34668 20.49589
```

```
# fit two separate models conditioning on sex
```

```
lm(weight[sex == "male"] ~ age[sex == "male"])
```

```
##
```

```
## Call:
```

```
## lm(formula = weight[sex == "male"] ~ age[sex == "male"])
```

```
##
```

```
## Coefficients:
```

```
##      (Intercept)  age[sex == "male"]
```

```
##           3.115           1.561
```

```
lm(weight ~ age, subset = (sex == "female"))
```

```
##
```

```
## Call:
## lm(formula = weight ~ age, subset = (sex == "female"))
##
## Coefficients:
## (Intercept)      age
##      1.9663      0.9962

# fit overall model
lm(weight ~ age)

##
## Call:
## lm(formula = weight ~ age)
##
## Coefficients:
## (Intercept)      age
##      2.541      1.279

# default contrasts in R
options(contrasts = c("contr.treatment", "contr.poly"))
model1 <- lm(weight ~ age * sex)
summary(model1)

##
## Call:
## lm(formula = weight ~ age * sex)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.23614 -0.60421  0.05658  0.65660  1.77263
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.9663     0.6268   3.137  0.00636 **
## age           0.9962     0.1010   9.862 3.33e-08 ***
## sexmale       1.1489     0.8864   1.296  0.21331
## age:sexmale   0.5646     0.1429   3.952  0.00114 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9175 on 16 degrees of freedom
## Multiple R-squared:  0.9652, Adjusted R-squared:  0.9587
## F-statistic: 147.8 on 3 and 16 DF,  p-value: 7.101e-12

#
options(contrasts = c("contr.helmert", "contr.poly"))
model2 <- lm(weight ~ age*sex)
summary(model2)

##
## Call:
## lm(formula = weight ~ age * sex)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.23614 -0.60421  0.05658  0.65660  1.77263
```

```
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.54073    0.44319   5.733 3.08e-05 ***
## age          1.27851    0.07143  17.899 5.26e-12 ***
## sex1         0.57445    0.44319   1.296 0.21331
## age:sex1     0.28230    0.07143   3.952 0.00114 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9175 on 16 degrees of freedom
## Multiple R-squared:  0.9652, Adjusted R-squared:  0.9587
## F-statistic: 147.8 on 3 and 16 DF,  p-value: 7.101e-12
```

```
#
options(contrasts = c("contr.sum", "contr.poly"))
model3 <- lm(weight ~ age * sex)
summary(model3)
```

```
##
## Call:
## lm(formula = weight ~ age * sex)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.23614 -0.60421  0.05658  0.65660  1.77263
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.54073    0.44319   5.733 3.08e-05 ***
## age          1.27851    0.07143  17.899 5.26e-12 ***
## sex1        -0.57445    0.44319  -1.296 0.21331
## age:sex1    -0.28230    0.07143  -3.952 0.00114 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9175 on 16 degrees of freedom
## Multiple R-squared:  0.9652, Adjusted R-squared:  0.9587
## F-statistic: 147.8 on 3 and 16 DF,  p-value: 7.101e-12
```

```
# reset
options(contrasts = c("contr.sum", "contr.poly"))

# orders matter in summary.aov
summary.aov(lm(weight ~ sex * age))
```

```
##           Df Sum Sq Mean Sq F value    Pr(>F)
## sex         1  90.49   90.49   107.50 1.66e-08 ***
## age         1 269.71  269.71   320.39 5.26e-12 ***
## sex:age     1  13.15   13.15    15.62 0.00114 **
## Residuals   16  13.47    0.84
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



```
summary.aov(lm(weight ~ age * sex))
```

```
##              Df Sum Sq Mean Sq F value    Pr(>F)
## age           1 269.71   269.71   320.39 5.26e-12 ***
## sex           1  90.49    90.49   107.50 1.66e-08 ***
## age:sex       1  13.15    13.15    15.62 0.00114 **
## Residuals    16  13.47     0.84
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# the above two results are the same
```

```
# however, the order matters here
```

```
attach(regrowth)
```

```
summary.aov(lm(Fruit ~ Grazing * Root))
```

```
##              Df Sum Sq Mean Sq F value    Pr(>F)
## Grazing       1   2910     2910   62.380 2.26e-09 ***
## Root          1 19149    19149  410.420 < 2e-16 ***
## Grazing:Root   1     5         5    0.103    0.75
## Residuals     36   1680         47
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary.aov(lm(Fruit ~ Root * Grazing))
```

```
##              Df Sum Sq Mean Sq F value    Pr(>F)
## Root          1  16795    16795  359.968 < 2e-16 ***
## Grazing       1   5264     5264  112.832 1.21e-12 ***
## Root:Grazing   1     5         5    0.103    0.75
## Residuals     36   1680         47
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# the above two have different sum of squares:
```

```
# it's because that in the first example, the x values for the continuous variable(age) were identical
# for both sexes, however, the x values (root size) in the second example is different in the two
# treatments and mean root size was greater for the grazed plants as shown below
```

```
tapply(Root, Grazing, mean)
```

```
##      Grazed Ungrazed
```

```
##      8.3094   6.0529
```

```
# however, the effect sizes and standard errors in the summary.lm table are completely unaffected.
```

```
summary(lm(Fruit ~ Root * Grazing))
```

```
##
## Call:
## lm(formula = Fruit ~ Root * Grazing)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.3177  -2.8320   0.1247   3.8511  17.1313
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)   -109.770      8.421 -13.035 3.36e-15 ***
## Root          23.618      1.177  20.066 < 2e-16 ***
## Grazing1     -15.403      8.421  -1.829  0.0757 .
## Root:Grazing1 -0.378      1.177  -0.321  0.7500
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.831 on 36 degrees of freedom
## Multiple R-squared:  0.9293, Adjusted R-squared:  0.9234
## F-statistic: 157.6 on 3 and 36 DF,  p-value: < 2.2e-16
```

```
summary(lm(Fruit ~ Grazing * Root))
```

```
##
## Call:
## lm(formula = Fruit ~ Grazing * Root)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.3177  -2.8320   0.1247   3.8511  17.1313
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -109.770      8.421 -13.035 3.36e-15 ***
## Grazing1      -15.403      8.421  -1.829  0.0757 .
## Root          23.618      1.177  20.066 < 2e-16 ***
## Grazing1:Root  -0.378      1.177  -0.321  0.7500
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.831 on 36 degrees of freedom
## Multiple R-squared:  0.9293, Adjusted R-squared:  0.9234
## F-statistic: 157.6 on 3 and 36 DF,  p-value: < 2.2e-16
```

```
detach(regrowth)
```

```
detach(Ancovacontrasts)
```

Summary: Whenever the x values are different in different factor levels, and/or there is different replication in different factor levels, then SSX, SSY will vary from level to level, and this will affect the way the sum of squares is distributed across the main effects. **However**, it's of no consequence in terms of the interpretation of the model.