Stats with Sparrows - 15

Julia Schroeder and David Orme

15

This practical will look at fitting models in R that use multiple variables to explain the response variable.

Housekeeping!

```
rm(list=ls())
setwd("H:/StatsWithSparrows")
```

Daphnia growth

We will stray from the sparrows and use a dataset on the growth of Daphnia populations, looking at the rate of growth in water containing four different detergents and using individuals of three different clones. There is also a variable on the water source used but we will not be adding it to the analysis here. We will load the data and then look at the data distribution using two box and whisker plots (Figure 1)

```
daphnia <- read.delim("daphnia.txt")
summary(daphnia)
```

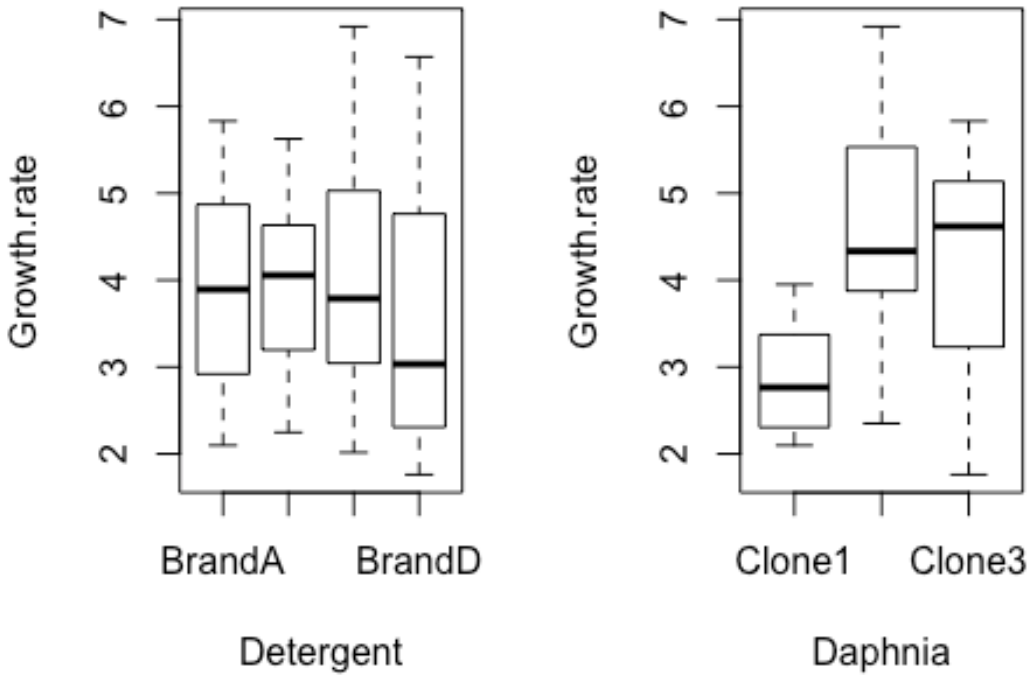
```
##   Growth.rate     Water     Detergent    Daphnia
##  Min.   :1.762   Tyne:36   BrandA:18   Clone1:24
##  1st Qu.:2.797   Wear:36   BrandB:18   Clone2:24
##  Median :3.788             BrandC:18   Clone3:24
##  Mean   :3.852             BrandD:18
##  3rd Qu.:4.807
##  Max.   :6.918
```

We go through the list we've gone through in the lecture. First things first!

1: Outliers

We check for potential outliers in x and y? From the summary data, we can see that the categories have sufficient samples size - this is a homogeneous dataset!

```
par(mfrow = c(1, 2))
plot(Growth.rate ~ Detergent, data = daphnia)
plot(Growth.rate ~ Daphnia, data = daphnia)
```



Outliers in boxplots come up as circles. So, no there are none.

2: Homogeneitry of variances

This is an important assumption for ANOVAS and regression analysis in general. To run the mdel explaining growth rate with Detergent brant and genotype, we have to assume that the variances within each brand, and within each genotype are similar. Looking at the plot, they are *sort of* similar. A rule of thumb (for ANOVA) is that the ratio between the largest and smallest variance should not be much more than 4 (this is a conservative estimate).

```
require(dplyr)
```

```
## Loading required package: dplyr
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
daphnia %>%
  group_by(Detergent) %>%
  summarise (variance=var(Growth.rate))
```

```
## # A tibble: 4 × 2
##   Detergent variance
##      <fctr>    <dbl>
## 1    BrandA 1.511245
## 2    BrandB 1.089727
## 3    BrandC 1.779843
## 4    BrandD 2.380693
```

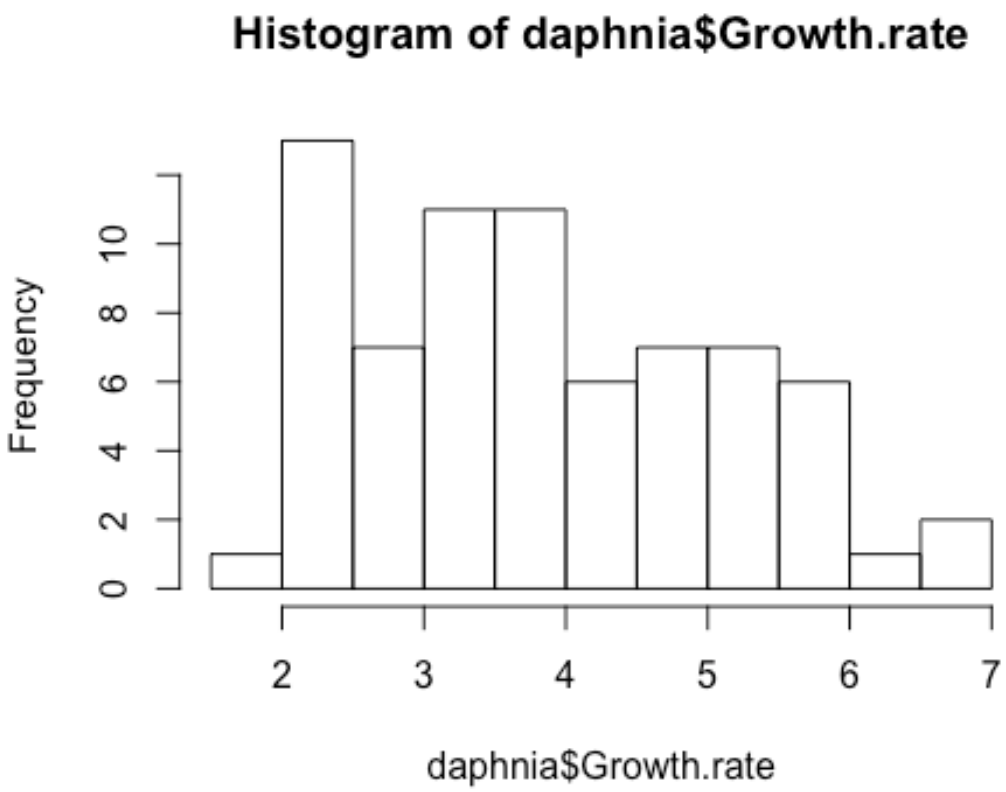
```
daphnia %>%
  group_by(Daphnia) %>%
  summarise (variance=var(Growth.rate))
```

```
## # A tibble: 3 × 2
##   Daphnia  variance
##    <fctr>     <dbl>
## 1  Clone1 0.3313181
## 2  Clone2 1.5300977
## 3  Clone3 1.5289960
```

Well. The ratio of variances for Clone 1 with the other two is larger than 4, it's about 5, actually. Well, this is not *much* more. This is for ANOVA, and for regression, we really want to look at the variances of the residuals. Hmm. It's borderline. Our goal is a regression analysis and not the ANOVA. Also, note that ecological data is always very messy, and you can get stuck if you try to follow all these rules - they can stop you from doing anything! Thus, instead of being intimidated, we'll plough ahead, but we will keep this in mind when we *interpret our model and draw conclusions*! That's the important bit - you can do all statistical analysis you like, you can validate the assumptions, and that's all ok *be explicitely clear about it in your report* ("the assumption of normality were violated - the ratio between the largest and smalles variance was 5, which is slightly too much and might bias the least square estimators") and *consider the consequences of this for when you draw your conclusions* (also explicitely in your report).

3: Are the data normally distributed?

```
hist(daphnia$Growth.rate)
```



Errr. Well. This is a good one. WHAT exactly needs to be normally distributed? And how close to normal should it be? Linear regression assumes normality, but *is reasonably robust against violations*. However, it assumes that the observations for each x are normal. So, if you measure something at x-2 10 times, you'd expect the resulting y to be normally distributed. Zuur et al. 2010 nicely shows this. Really, we are interested in the residuals. So we'll look at that later in more detail and for now hope that the growth rate is ok-ish normally distributed.

4: Are there excessively many zeroes?

Looking at the histogram, no.

5: Is there collinearity among the covariates?

Well, we only have categories here. So it doesn't apply. What we could do is check if all combinations are represented, and if so if there is any correlation. But as we know the dataset if homogeneous, that will suffice.

6: Visually inspect relationships

Done that before with the boxplots. No continous covariates. It loks like Clone 1 has an effect, but that's about it.

7: Consider interactions?

Not this time around!

Ok, let's get the the meat:

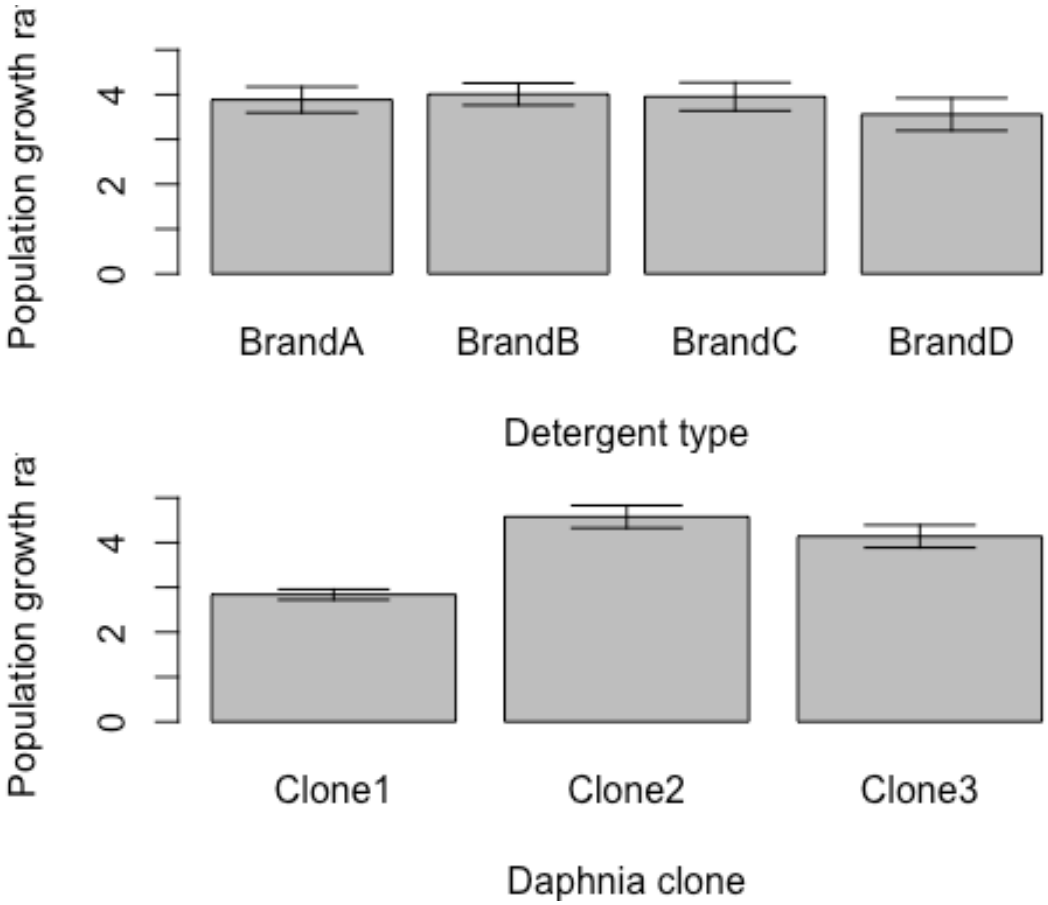
Model daphnia:

We can also use our superior plotting skills from last week to create barplots showing the means and standard errors of th emean for both clonal genotype and detergent presence. We will first use the function tapply to get the means and standard deviations for each of the two explanatory variables. Tapply is about as useful as dpyr. Often, you can reach the same goal in R with different means.

```
seFun <- function(x) {
    sqrt(var(x)/length(x))
}
detergentMean <- with(daphnia, tapply(Growth.rate, INDEX = Detergent,
    FUN = mean))
detergentSEM <- with(daphnia, tapply(Growth.rate, INDEX = Detergent,
    FUN = seFun))
cloneMean <- with(daphnia, tapply(Growth.rate, INDEX = Daphnia, FUN = mean))
cloneSEM <- with(daphnia, tapply(Growth.rate, INDEX = Daphnia, FUN = seFun))
```

Now we can use par(mfrow=(2,1), mar=c(4,4,1,1)) to plot them one above the other on the same graphics device and to reduce the size of the margins (Figure 2).

```
par(mfrow=c(2,1),mar=c(4,4,1,1))
barMids <- barplot(detergentMean, xlab = "Detergent type", ylab = "Population
growth rate",
    ylim = c(0, 5))
arrows(barMids, detergentMean - detergentSEM, barMids, detergentMean +
    detergentSEM, code = 3, angle = 90)
barMids <- barplot(cloneMean, xlab = "Daphnia clone", ylab = "Population
growth rate",
    ylim = c(0, 5))
arrows(barMids, cloneMean - cloneSEM, barMids, cloneMean + cloneSEM,
    code = 3, angle = 90)
```



The differences in the means for the detergents don't look like they matter but we should test whether they have any explanatory power. We can do this by adding both variables into the formula describing the model. So far we have only seen this in simple situations where there is one variable describing another (y ~ x). We can use the + sign to add extra variables into the right hand side of the formula: y ~ x + z means model y using both the x and z variables. So now we can fit the model and look at the analysis of variance table:

```
daphniaMod <- lm(Growth.rate ~ Detergent + Daphnia, data = daphnia)
anova(daphniaMod)

## Analysis of Variance Table
##
## Response: Growth.rate
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Detergent  3  2.212  0.7372  0.6422    0.5906
## Daphnia    2 39.178 19.5889 17.0635 1.064e-06 ***
## Residuals 66 75.768  1.1480
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We now have the ANOVA table with line for each variable. In each case, we fol- low exactly the same procedure of using an F test on the mean squares — does this variable explain a significant amount of variation in the data? In each case, we compare the mean square

variation for the line ('Mean Sq.') to the residual mean square variation: 0.737/1.148 and 19.589/1.148.

From this, we conclude that genotype is important in determining the population growth rate measured in the Daphnia but that the detergents do not have any effect.. We can now use the same techniques as before to see the differences in the means between each detergent and each genotype.

```
summary(daphniaMod)
```

```
##
## Call:
## lm(formula = Growth.rate ~ Detergent + Daphnia, data = daphnia)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.25917 -0.72208 -0.06135  0.71041  2.28597
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      2.87280    0.30930   9.288 1.34e-13 ***
## DetergentBrandB  0.12521    0.35715   0.351    0.727
## DetergentBrandC  0.06968    0.35715   0.195    0.846
## DetergentBrandD -0.32660    0.35715  -0.914    0.364
## DaphniaClone2    1.73725    0.30930   5.617 4.21e-07 ***
## DaphniaClone3    1.29884    0.30930   4.199 8.19e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.071 on 66 degrees of freedom
## Multiple R-squared:  0.3533, Adjusted R-squared:  0.3043
## F-statistic: 7.211 on 5 and 66 DF,  p-value: 1.944e-05
```

This table confirms that none of the estimated mean differences for the detergent types differ from the first and shows that both clones 2 and 3 differ from clone 1. We also see that the model is fairly poor at explaining the data — the adjusted r^2 is 0.30, so over half the variation in the data is not explained by the model. Though, note that biology deals with messy, complex systems where there is often massive variation in data. An r^2 of 0.3 is therefore often cause for rejoicing. Anyways, let's move on: The coefficient table tests whether one mean is different from zero and then tests whether the other means are different from the first. The first value it uses this time is the mean of the data from the first detergent brand for the first clonal genotype. The next lines are then the difference from this 'Brand A, Clone 1' reference to each of the other means. We know these differences already because they are the same as the differences in the means we calculated for the barplot:

```
detergentMean - detergentMean[1]
```

```
##      BrandA      BrandB      BrandC      BrandD
##  0.00000000  0.12521198  0.06968013 -0.32660105

cloneMean - cloneMean[1]

##   Clone1   Clone2   Clone3
## 0.000000 1.737246 1.298845
```

So to get the mean for 'Brand A, Clone 2', we add 1.74 on to the 'Brand A, Clone 1' reference value of 2.87. If we want to get 'Brand B, Clone 3', we need to combine coefficients and add the 0.125 to go from Brand A to Brand B and the 1.30 to go from Clone 1 to Clone 2. We can use the Tukey HSD test to test all the pairwise differences. This is a very useful test to test across multiple categories. But to do that we first need to run the model (slightly) differently (because it doesn't accept the lm input).

```
daphniaANOVAMod <- aov(Growth.rate ~ Detergent + Daphnia, data = daphnia)
summary(daphniaANOVAMod)

##             Df Sum Sq Mean Sq F value   Pr(>F)
## Detergent    3   2.21   0.737   0.642    0.591
## Daphnia      2  39.18  19.589  17.063 1.06e-06 ***
## Residuals   66  75.77   1.148
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

?aov
```

You can see that the aov method is similar to lm, but less powerful as it only fits the ANOVA. Yet, when we want to do multiple comparisons, it can be quite useful.

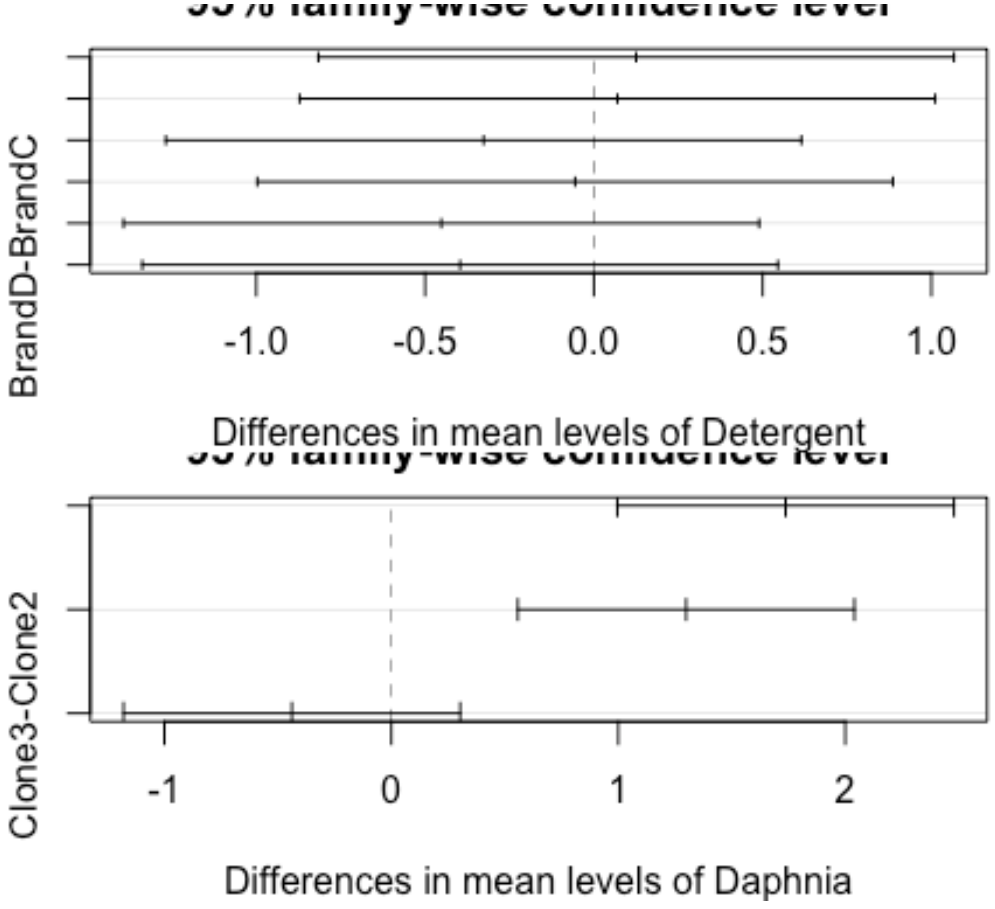
```
daphniaModHSD <- TukeyHSD(daphniaANOVAMod)
daphniaModHSD

##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Growth.rate ~ Detergent + Daphnia, data = daphnia)
##
## $Detergent
##                       diff        lwr       upr     p adj
## BrandB-BrandA  0.12521198 -0.8161307 1.0665547 0.9850797
## BrandC-BrandA  0.06968013 -0.8716625 1.0110228 0.9973423
## BrandD-BrandA -0.32660105 -1.2679437 0.6147416 0.7972087
## BrandC-BrandB -0.05553185 -0.9968745 0.8858108 0.9986474
## BrandD-BrandB -0.45181303 -1.3931557 0.4895296 0.5881893
## BrandD-BrandC -0.39628118 -1.3376239 0.5450615 0.6849619
##
## $Daphnia
##                     diff        lwr       upr     p adj
## Clone2-Clone1  1.737246  0.9956362 2.4788555 0.0000013
```

```
## Clone3-Clone1  1.298845  0.5572351 2.0404544 0.0002393
## Clone3-Clone2 -0.438401 -1.1800107 0.3032086 0.3378930
```

Cool! This Tukey test even gives us the upper and lower 95 confidence intervals. That's brilliant!

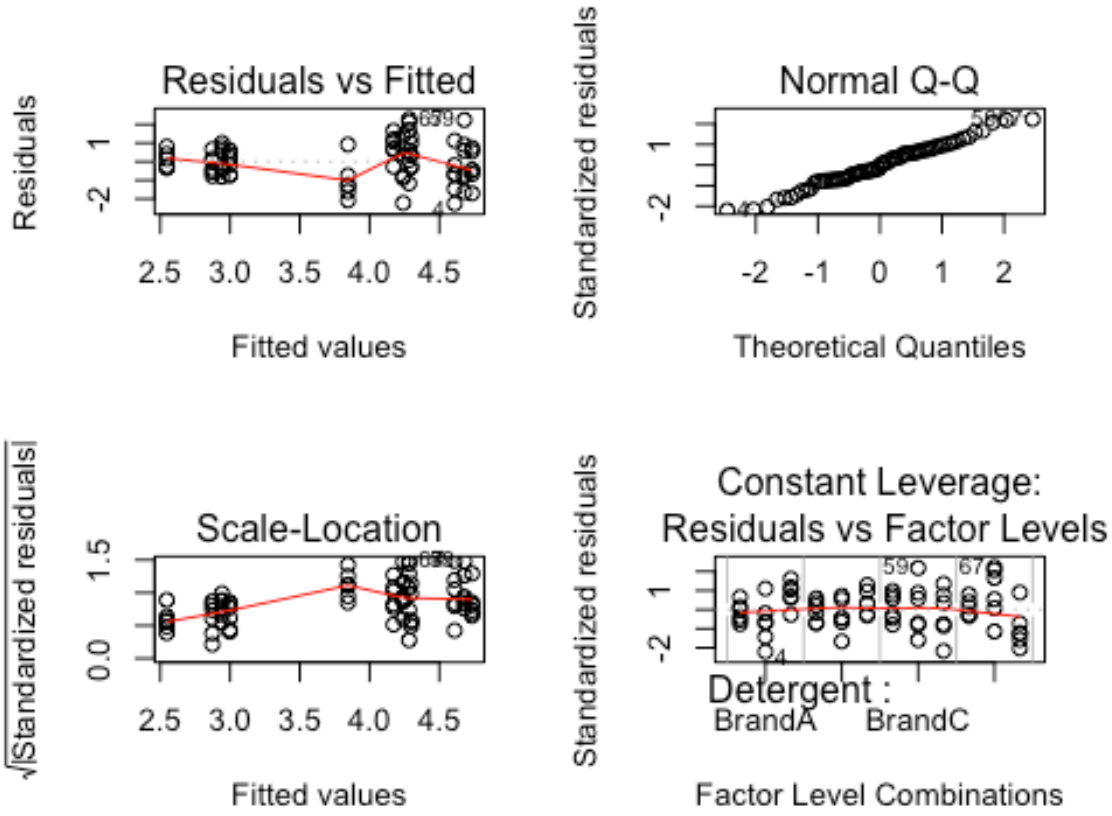
The Tukey test now gives us two tables: the first shows which pairs of detergents differ in their effect on population growth rate (spoiler: none!) and the second shows which of the differences in genotype matter. It shows that the extra comparison between Clone 2 and 3 is not significant. Plotting daphniaModHSD (Figure 3) will also give us two plots, so we will use par(mfrow=c(1,2)) again to put them on the same page. We are also going to need a wide margin on the left for all the long pairwise labels. One new par() option: the setting las=1 changes the orientation of the axis labels so that they are all horizontal.

```
par(mfrow=c(2,1),mar=c(4,4,1,1))
plot(daphniaModHSD)
```



Now some model validation:

```
par(mfrow=c(2,2))
plot(daphniaMod)
```



Ok. So much for stars in the sky. I'm sure this left bit comes from clone 3. The QQ plot at least looks good, and there is no outliner indeed. While this is not nice, it's also not the end of the world. This is publishable - given that you openly explain all the things that may affect how one interprets the data. With experience, you'll get a better "feel" for what is ok and what not. But you need to do it to get that feel, so don't be frustrated when your data doesn't fit perfectly! It never will.

Good. Now we move on to some more real bits!

Multiple regression

We will use an example dataset that looks at the volume of usable timber harvested from trees of known height and girth (diameter). We want to see whether both height and girth are important in predicting the yield from a tree. We will use the built in pairs() function to look at the distribution of the data: this generates a neat layout from a data frame that plots each variable against each other variable in the data frame (Figure 4).

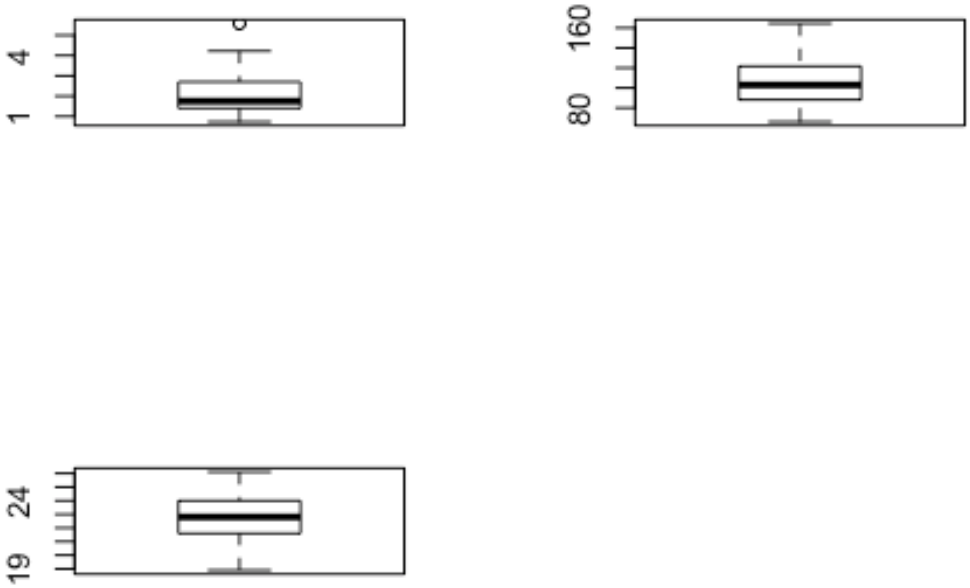
```
timber <- read.delim("timber.txt")
summary(timber)

##      volume           girth            height
##  Min.   :0.7386   Min.   : 66.23   Min.   :18.9
##  1st Qu.:1.4048   1st Qu.: 88.17   1st Qu.:21.6
```

```
##  Median :1.7524   Median :102.94   Median :22.8
##  Mean   :2.1847   Mean   :105.72   Mean   :22.8
##  3rd Qu.:2.7010   3rd Qu.:121.69   3rd Qu.:24.0
##  Max.   :5.5757   Max.   :164.38   Max.   :26.1
```

Let's get through our list:

1: Outliers

```
par(mfrow = c(2, 2))
boxplot(timber$volume)
boxplot(timber$girth)
boxplot(timber$height)
```




```
## null device
##           1
```

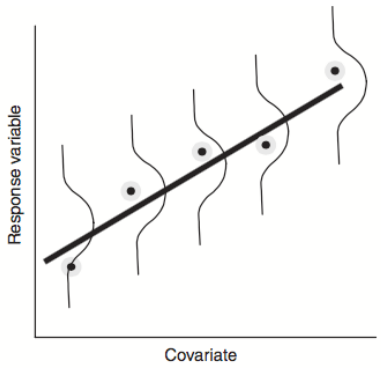
Ok. So there is one outlier in volume. It's rather large, but not excessively so - nothing suggeststst that it's a measurement error or typo or so. It seem biologically true, so we keep it in but remember it's there for when we look at leverage. If this point would, say, be very important in determining some relationship we might want to have a look whether the model would turn out differently if we'd take it out. Generally, it's always better to keep stuff in! If you decide to take outliers out, *always disclose that in your report and JUSTIFY why you did it on biological grounds*. Stating "the analysis was odd with this data point in "

is not a good justification. Saying "I think it's a typo because there is no tree in this world with a diameter of 5km" is biologically reasonable.

2: Homogeneitry of variances

```
var(timber$volume)

## [1] 1.416803

var(timber$girth)

## [1] 627.0461

var(timber$height)

## [1] 3.654
```

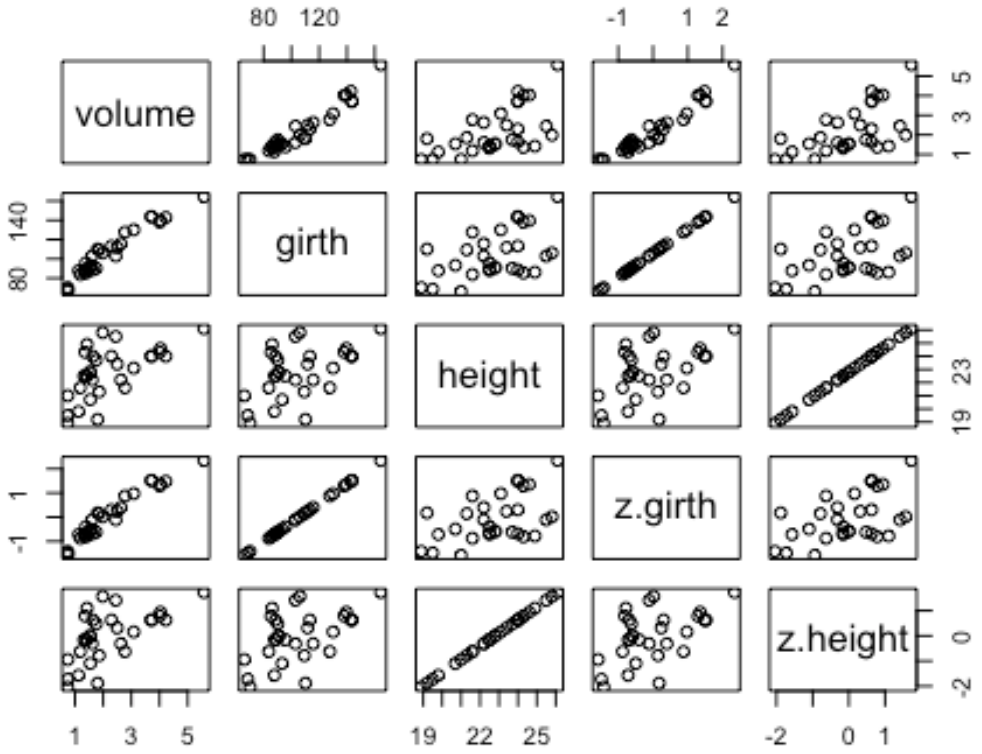
Ouf. Wait a moment. What does this thing "homogeneity of variances" really mean? It talks about the variance of all the ys of one single x. That's easy for categorical variables as in the daphnia dataset.



(I copied this Figure from Zuur et al. 2010).

So, well, we can't do much here, can we? Well, there is one thing that might be good to do. Since we're interested in volume (y), we could standardize our x'es for the analysis.

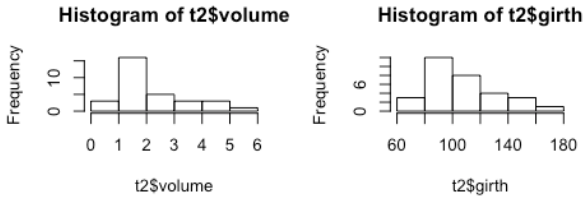
```
t2<-as.data.frame(subset(timber, timber$volume!="NA"))
t2$z.girth<-scale(timber$girth)
t2$z.height<-scale(timber$height)
var(t2$z.girth)

##      [,1]
## [1,]    1

var(t2$z.height)

##      [,1]
## [1,]    1

plot(t2)
```

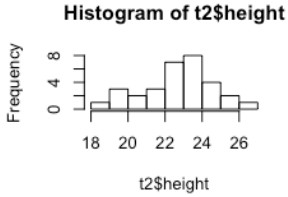


Ok. It's probably a good idea to run the analysis with z-scores.

3: Are the data normally distributed?

```
par(mfrow = c(2, 2))
hist(t2$volume)
hist(t2$girth)
hist(t2$height)
```





```
## null device
##           1
```

We've learned and won't freak out this time. This is as good as it gets in EEC.

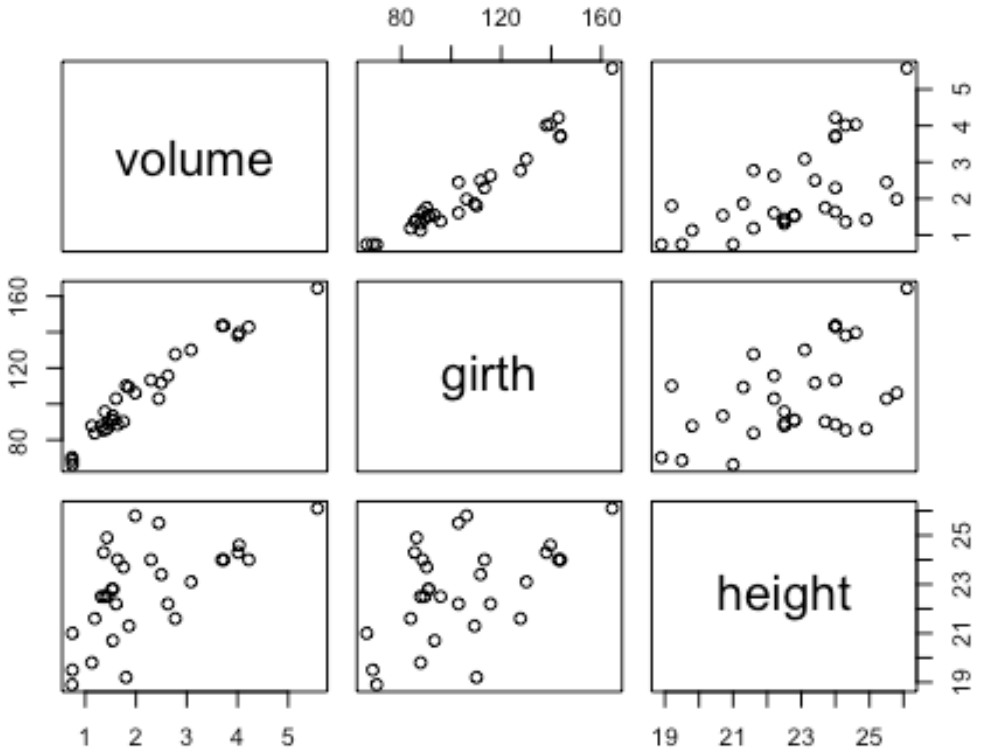
4: Are there excessively many zeroes?

Nope.

5: Is there collinearity among the covariates?

Uh. This is a nice one.

```
pairs(timber)
```




```
cor(timber)
```

```
##            volume     girth    height
## volume 1.0000000 0.9671176 0.5982517
## girth  0.9671176 1.0000000 0.5192873
## height 0.5982517 0.5192873 1.0000000
```

So these variables are all positively correlated, with tree diameter being a better predictor of timber yield than height. Too much correlation among the predictors (girth and height)

is not a good thing. This is called "collinearity". What it does is it inflates variation. So, when you hvae a lot of collinearity in your covariates, you'll get larger standart errors of those correlated variables than you'd get if there was no collinearity. That means that it is more difficult to detect an effect, that *you are likely to not get a significant result even though there might be one*. This means that *if there is lots of collinearity any normally evaluation of a model is super conservative*. Also, dropping covariates can affect the estimates of other covariates if there is collinearity around. That can be super confusing. The standard errors are inflated with the square root of the *Variance Inflation Factor*. This VIF we can use to find out what amoung of collinearity is too much. VIF can be calculated by running an extra linear model in which the covariate of focus (here, girth) is y, and all other covariates of the model (here only one, height) are the covariates. Then you can calculate the VIF as follows:

$$VIF = \frac{1}{1 - R^2}$$

```
summary(lm(girth ~ height, data = timber))
```

```
##
## Call:
## lm(formula = girth ~ height, data = timber)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -33.82 -15.32  -0.57  21.90  36.21
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -49.382     47.559  -1.038  0.30770
## height         6.803      2.079   3.272  0.00276 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 21.77 on 29 degrees of freedom
## Multiple R-squared:  0.2697, Adjusted R-squared:  0.2445
## F-statistic: 10.71 on 1 and 29 DF,  p-value: 0.002757
```

```
VIF<- 1/(1-0.27)
VIF
```

```
## [1] 1.369863
```

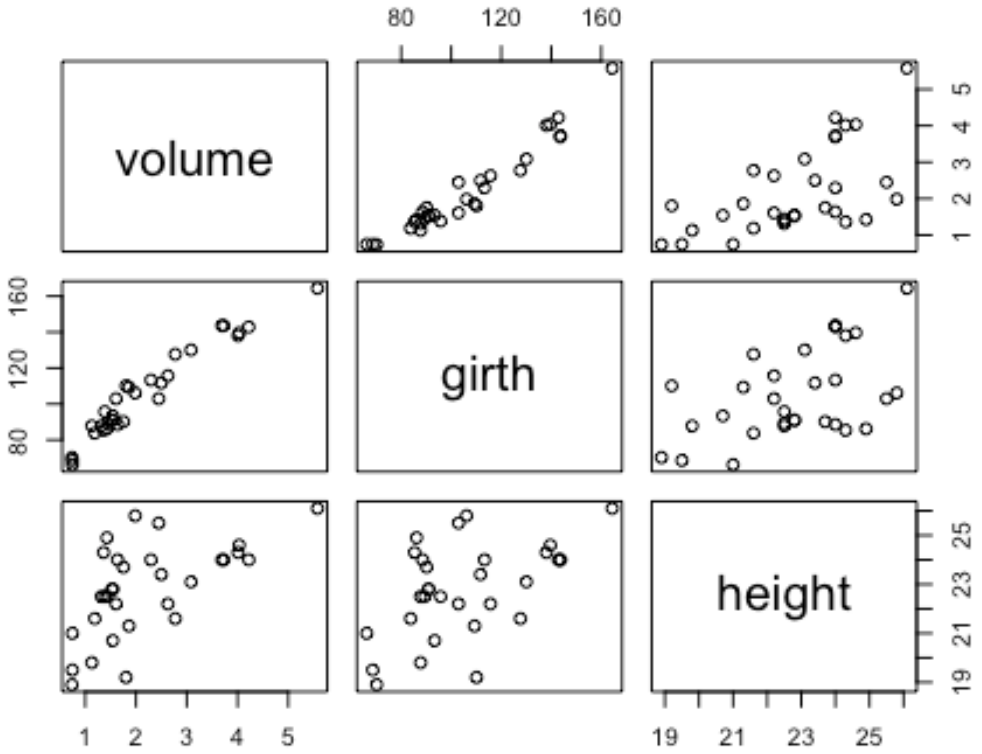
```
sqrt(VIF)
```

```
## [1] 1.170411
```

The standard errors of girth are thus inflated by 1.17, which is not a lot. A VIF of 1.4 is also ok. Some people are super strict and say throw out all covariates with VIF more than 3. Others say VIF more than 10. I say, it depends. I also say, test for it, and keep it in mind when you do your interpretation, and disclose it in your report! Think biology, always.

Back to the pairs plot:

```
pairs(timber)
```

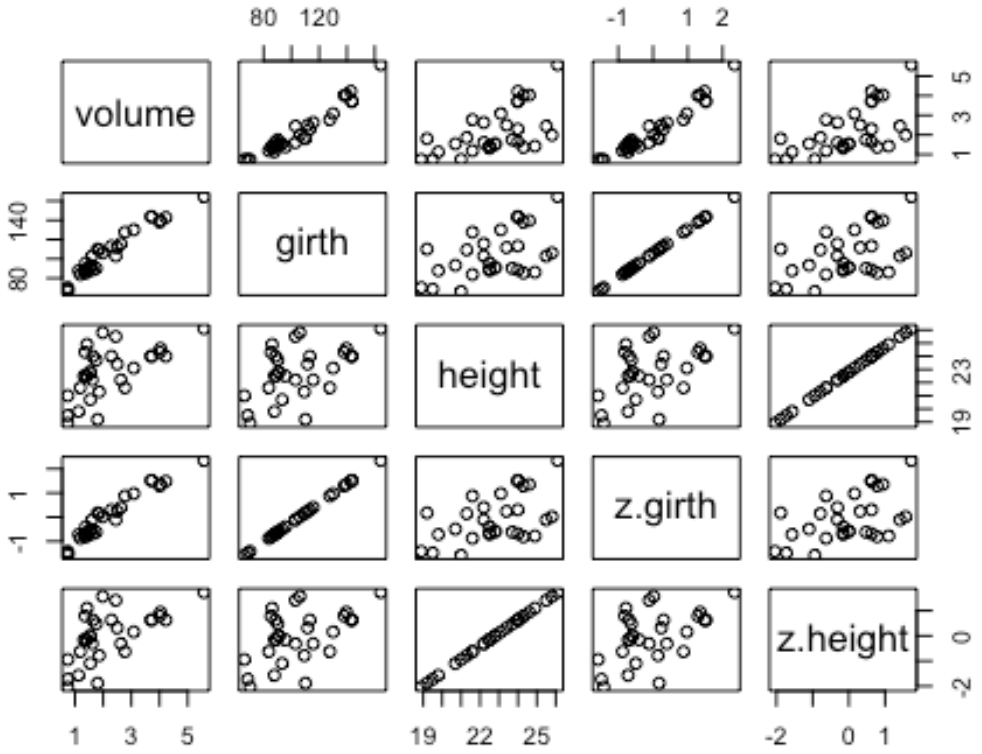



```
cor(timber)
```

```
##            volume     girth    height
## volume 1.0000000 0.9671176 0.5982517
## girth  0.9671176 1.0000000 0.5192873
## height 0.5982517 0.5192873 1.0000000
```

We also see the outlier in volume. It behaves as all other points behave, and doesn't influence the correlation much. But we'll see that better in the leverage plots later, but for now, we're happy. We can do the same thing with your scaled predictors, but that won't be much different:

```
pairs(t2)
```




```
cor(t2)

##               volume     girth    height   z.girth  z.height
## volume   1.0000000 0.9671176 0.5982517 0.9671176 0.5982517
## girth    0.9671176 1.0000000 0.5192873 1.0000000 0.5192873
## height   0.5982517 0.5192873 1.0000000 0.5192873 1.0000000
## z.girth  0.9671176 1.0000000 0.5192873 1.0000000 0.5192873
## z.height 0.5982517 0.5192873 1.0000000 0.5192873 1.0000000
```

You can see the correlations are the same (they should be).

6: Visually inspect relationships

Well, we've done that for the covariates, but same plot also shows the relationships with the response, volume. There seem to be some relationships.

7: Consider interactions?

Not now because we're still learning. We'll look at interactions with the next dataset.

Now on to our model:

If girth has such a high correlation with volume, do we actually need both variables?

```
timberMod <- lm(volume ~ girth + height, data = timber)
anova(timberMod)

## Analysis of Variance Table
##
## Response: volume
##           Df Sum Sq Mean Sq  F value Pr(>F)
## girth      1 39.755  39.755 503.1070 <2e-16 ***
## height     1  0.537   0.537   6.7933 0.0145 *
## Residuals 28  2.213   0.079
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

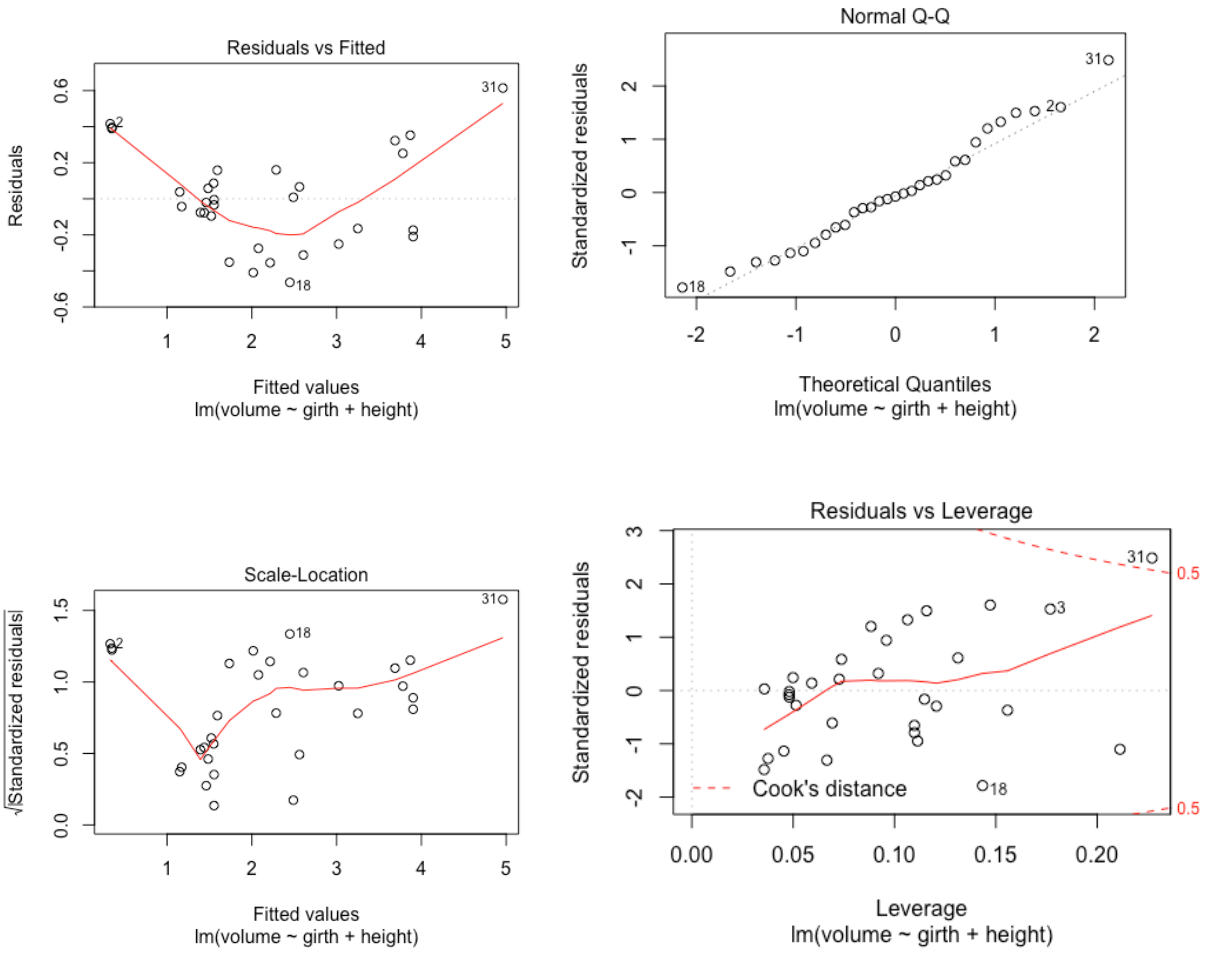
So yes, it looks like height is needed as well as the girth of the tree. If we look at the coefficients in the linear model summary, then we can get the estimates of the slopes and intercept.

```
summary(timberMod)

##
## Call:
## lm(formula = volume ~ girth + height, data = timber)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.46391 -0.19171 -0.02072  0.15929  0.61439
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.198997   0.625537  -6.713 2.75e-07 ***
## girth        0.042725   0.002398  17.815  < 2e-16 ***
## height       0.081883   0.031416   2.606   0.0145 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2811 on 28 degrees of freedom
## Multiple R-squared:  0.9479, Adjusted R-squared:  0.9442
## F-statistic:   255 on 2 and 28 DF,  p-value: < 2.2e-16
```

So, looking at the r^2, more than 90 percent of the variation in timber volume is explained by the following equation: *volume = −4.2 + 0.08 × height + 0.04 × girth*. One important thing to note is that this model makes stupid predictions if the tree is very small — a one metre sapling of diameter 10 cm will contain -3.72 tonnes of timber. It is not at all sensible to expect a statistical model to make good predictions outside of the range of the data used to fit it.

Time for model validation:

```
plot(timberMod)
```



Ouch. Again, not so much starry sky. QQ is ol. Leverage is not nice, and htere seems to be one point especially that stands out (31). If we'd want to publish this we'd run the whole thing without this point and see if we'd come to the same conclusions.

Exercises:

1) Run the timber model without the previously found outlier. See what your conclusions are. If you'd publish it, would you do it with outlier or without? Why?

2) Another example: The previous two examples looked at predicting a variable using two categorical variables and then two continuous variables. What if you have both continuous and categorical variables like sex and tarsus in the lecture? We will use the Ipopmopsis dataset, which records plant fruit production as a function of root stock diameter for plants that have been grazed by rabbit and those that have been protected from grazing. We want to know whether fruit production (response) is associated with both grazing (fixed factor) and root stock size (continous covariate). The model we are fitting here is describing fruit production as a function of root stock size — this is like in a normal regression, so we are after a slope and an intercept. In addition to this, we want to fit fruit production as a function of the two levels of

grazing — wer are testing whether there is a difference in intercept between the two lines.

```
plantGrowth <- read.delim("ipomopsis.txt")
summary(plantGrowth)
```

```
##       Root            Fruit            Grazing
##  Min.   : 4.426   Min.   : 14.73   Grazed  :20
##  1st Qu.: 6.083   1st Qu.: 41.15   Ungrazed:20
##  Median : 7.123   Median : 60.88
##  Mean   : 7.181   Mean   : 59.41
##  3rd Qu.: 8.510   3rd Qu.: 76.19
##  Max.   :10.253   Max.   :116.05
```

Use the checklist (1-11) on the plant growth dataset. Fruit is the response, Root the covariate and Grazing the fixed factor.You want to specify your null model, the maximum model, and find the best model that includes and interaction between Root and Grazing (indicated with a * in your model formula). You need to run through the checklist as we've done above first, this time at point 6 to also plot potential interactions. You can do that best by seperately plotting xy plots and/or cor.tests for each category. Then you want to do model validation.

3) Use the sparrow dataset to find out how much each structural measuement (tarsus, wing, bill) and sex affects body mass. Use multiple linear models to do that. Run the checklist first. Write down the null-model, the maximum model, and the final model that you select to be the best fit. Explain what it means.

On the nest page, you'll find the check list as a whole for reference.

1: Outliers

2: Homogeneitry of variances

3: Are the data normally distributed?

4: Are there excessively many zeroes?

5: Is there collinearity among the covariates?

6: Visually inspect relationships

7: Consider interactions?

8: Decide on maximal model based on biology and question

9: Simplify model

10: Decide on final model

11: Run model validation