Topic 7 answers

Steve Paterson

2024-06-12

Chlorophyll

> *TASK*
>
> The file chlorophyll.txt contains data on chlorophyll levels (mg/L) on a number of lakes, plus information on pH, water alkalinity (mg/L carbonate levels) and calcium concentration (mg/L).
>
> Generate and present a minimal model that identifies the main driver(s) of chlorophyll levels.

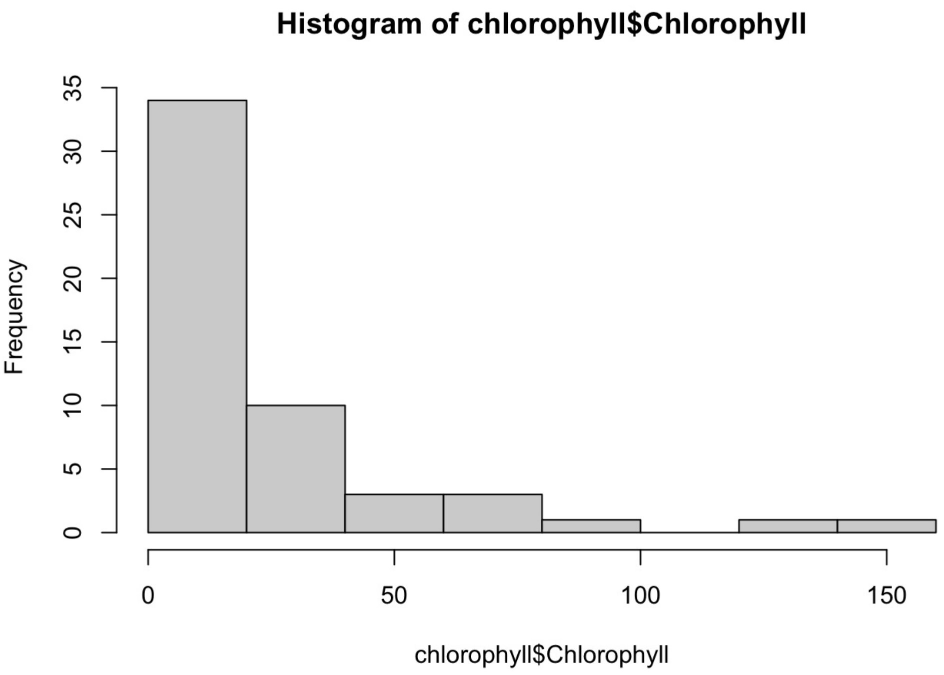
Read in the data

```
chlorophyll <- read.table("chlorophyll.txt",header=TRUE)
summary(chlorophyll)
```

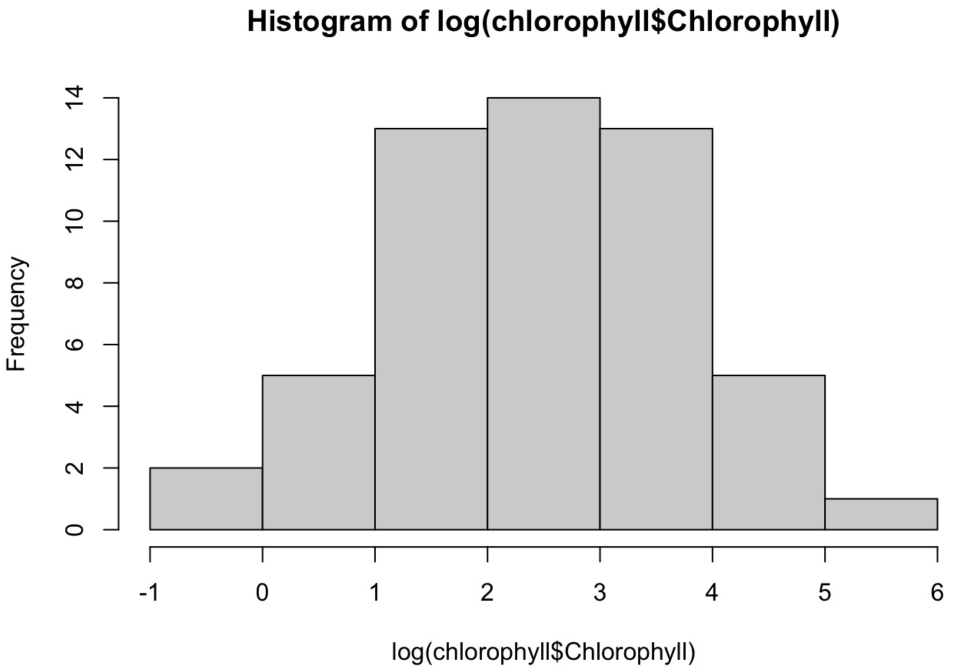
```
##      Lake            Chlorophyll           pH           Alkalinity    
##  Length:53          Min.   :  0.70   Min.   :3.600   Min.   :  1.20  
##  Class :character   1st Qu.:  4.60   1st Qu.:5.800   1st Qu.:  6.60  
##  Mode  :character   Median : 12.80   Median :6.800   Median : 19.60  
##                     Mean   : 23.12   Mean   :6.591   Mean   : 37.53  
##                     3rd Qu.: 24.70   3rd Qu.:7.400   3rd Qu.: 66.50  
##                     Max.   :152.40   Max.   :9.100   Max.   :128.00  
##     Calcium    
##  Min.   : 1.1  
##  1st Qu.: 3.3  
##  Median :12.6  
##  Mean   :22.2  
##  3rd Qu.:35.6  
##  Max.   :90.7  
```

Getting a feel for the data, and the distribution of the variables is worthwhile

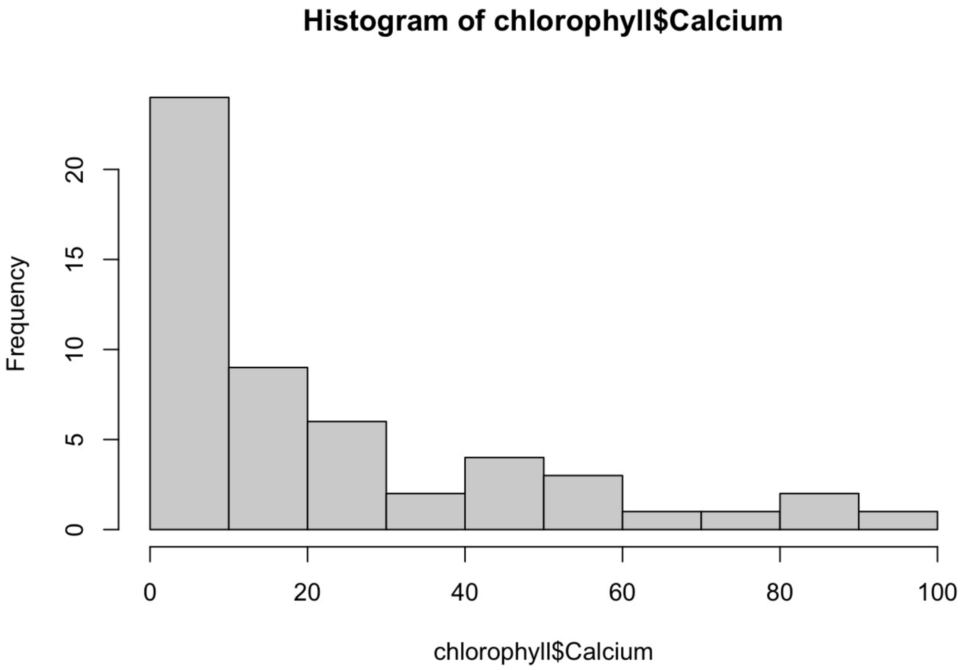
```
hist(chlorophyll$Chlorophyll)
```



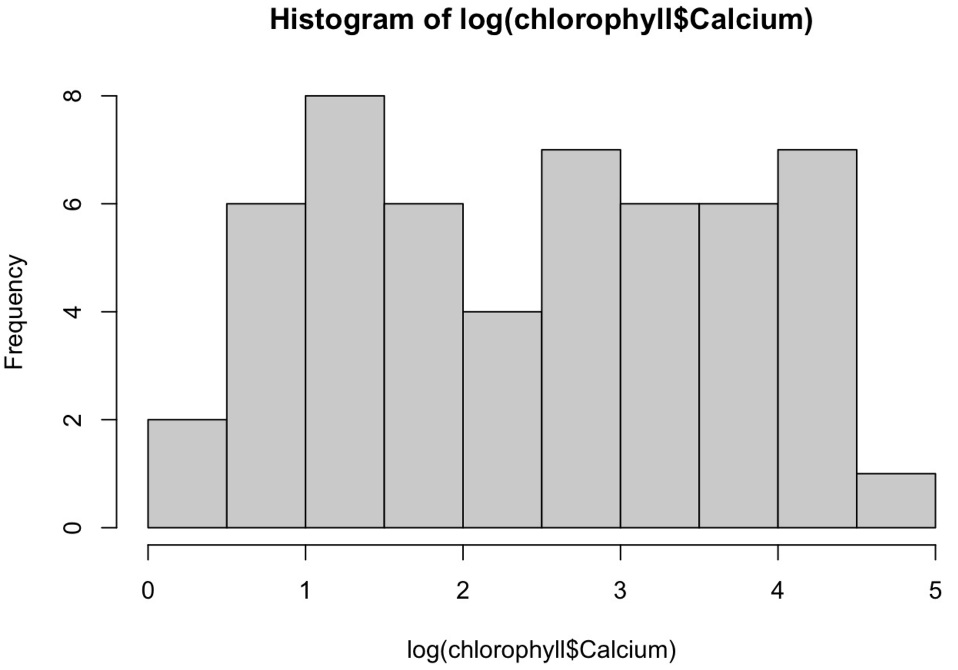
```
hist(log(chlorophyll$Chlorophyll))
```



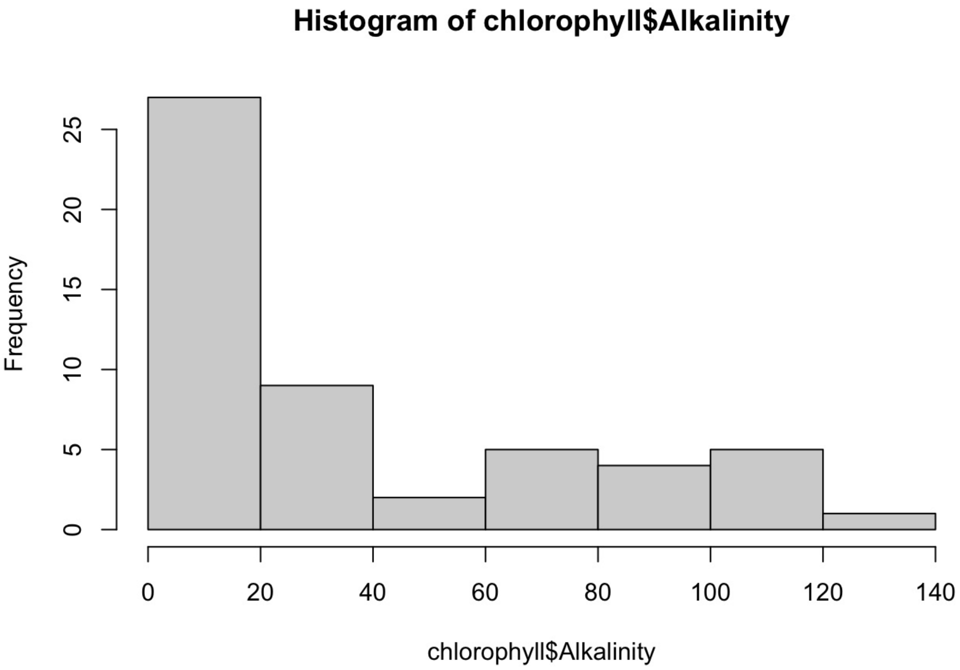
```
hist(chlorophyll$Calcium)
```



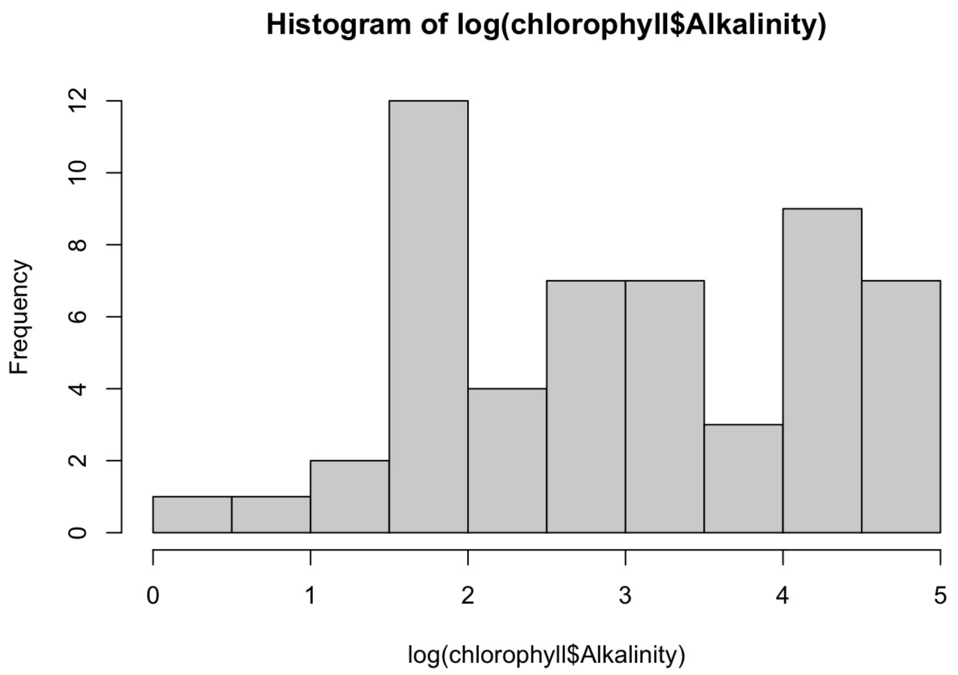
```
hist(log(chlorophyll$Calcium))
```



```
hist(chlorophyll$Alkalinity)
```



```
hist(log(chlorophyll$Alkalinity))
```



Possibly more investigation of transforms could be done, but what’s here suggests that probably log chlorophyll and log Calcium may be best, Alkanity not sure

```
ch1 <- lm(log(Chlorophyll) ~ (pH + log(Calcium)+Alkalinity)^2,data= chlorophyll)
ch2 <- lm(log(Chlorophyll) ~ (pH + log(Calcium)+log(Alkalinity))^2,data=chlorophyll)

summary(ch1)
```

```
## 
## Call:
## lm(formula = log(Chlorophyll) ~ (pH + log(Calcium) + Alkalinity)^2, 
##     data = chlorophyll)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -2.48342 -0.64062  0.02989  0.73236  1.66737 
## 
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)  
## (Intercept)              0.605975   1.613770   0.376   0.7090  
## pH                       0.320412   0.315495   1.016   0.3151  
## log(Calcium)            -2.247570   1.146440  -1.960   0.0560 .
## Alkalinity              -0.006846   0.056252  -0.122   0.9037  
## pH:log(Calcium)          0.323343   0.172240   1.877   0.0668 .
## pH:Alkalinity           -0.003789   0.006317  -0.600   0.5516  
## log(Calcium):Alkalinity  0.006990   0.006485   1.078   0.2867  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.9543 on 46 degrees of freedom
## Multiple R-squared:  0.5066, Adjusted R-squared:  0.4422 
## F-statistic: 7.872 on 6 and 46 DF,  p-value: 7.332e-06
```

```
summary(ch2)
```

```
##
## Call:
## lm(formula = log(Chlorophyll) ~ (pH + log(Calcium) + log(Alkalinity))^2,
##     data = chlorophyll)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4309 -0.5505  0.1106  0.7229  1.5758
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  -0.392992   2.148877  -0.183   0.8557
## pH                            0.477319   0.491699   0.971   0.3367
## log(Calcium)                 -2.555070   1.302456  -1.962   0.0559 .
## log(Alkalinity)               0.930170   1.153554   0.806   0.4242
## pH:log(Calcium)               0.399430   0.203198   1.966   0.0554 .
## pH:log(Alkalinity)           -0.181939   0.197447  -0.921   0.3616
## log(Calcium):log(Alkalinity)  0.002428   0.163509   0.015   0.9882
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9666 on 46 degrees of freedom
## Multiple R-squared:  0.4938, Adjusted R-squared:  0.4277
## F-statistic: 7.478 on 6 and 46 DF,  p-value: 1.262e-05
```

The r2 for ch1 slightly better than ch2

Deletion test down from this, as a sortcut use step(), but this may not get all the way to the minimal model so we still have to check what it produces

```
ch3 <- step(ch1)
```

```
## Start:  AIC=1.53
## log(Chlorophyll) ~ (pH + log(Calcium) + Alkalinity)^2
##
##                           Df Sum of Sq    RSS     AIC
## - pH:Alkalinity            1    0.3275 42.215 -0.0579
## - log(Calcium):Alkalinity  1    1.0580 42.946  0.8513
## <none>                                 41.888  1.5293
## - pH:log(Calcium)          1    3.2091 45.097  3.4417
##
## Step:  AIC=-0.06
## log(Chlorophyll) ~ pH + log(Calcium) + Alkalinity + pH:log(Calcium) +
##     log(Calcium):Alkalinity
##
##                           Df Sum of Sq    RSS      AIC
## - log(Calcium):Alkalinity  1    1.2687 43.484 -0.48856
## <none>                                 42.215 -0.05793
## - pH:log(Calcium)          1    4.1294 46.345  2.88821
##
## Step:  AIC=-0.49
## log(Chlorophyll) ~ pH + log(Calcium) + Alkalinity + pH:log(Calcium)
##
##                   Df Sum of Sq    RSS     AIC
## - Alkalinity       1    0.7164 44.200 -1.6225
## <none>                         43.484 -0.4886
## - pH:log(Calcium)  1    5.3697 48.854  3.6826
##
## Step:  AIC=-1.62
## log(Chlorophyll) ~ pH + log(Calcium) + pH:log(Calcium)
##
##                   Df Sum of Sq   RSS     AIC
## <none>                         44.20 -1.6225
## - pH:log(Calcium)  1    4.9199 49.12  1.9710
```

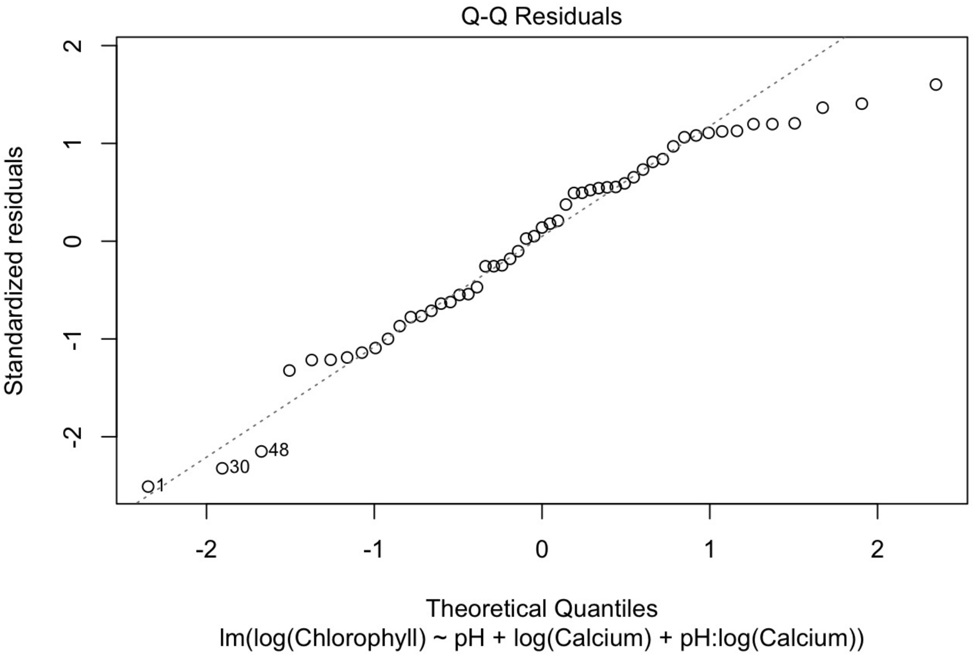
```
drop1(ch3,test="F")
```

```
## Single term deletions
## 
## Model:
## log(Chlorophyll) ~ pH + log(Calcium) + pH:log(Calcium)
##                 Df Sum of Sq   RSS     AIC F value  Pr(>F)  
## <none>                       44.20 -1.6225                  
## pH:log(Calcium)  1    4.9199 49.12  1.9710  5.4541 0.02366 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(ch3) #present a tidied up version of this in a report
```

```
## 
## Call:
## lm(formula = log(Chlorophyll) ~ pH + log(Calcium) + pH:log(Calcium), 
##     data = chlorophyll)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -2.3194 -0.6603  0.1187  0.7501  1.4843 
## 
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)  
## (Intercept)       0.7877     1.3952   0.565   0.5750  
## pH                0.2181     0.2480   0.880   0.3834  
## log(Calcium)     -1.4836     0.6521  -2.275   0.0273 *
## pH:log(Calcium)   0.2200     0.0942   2.335   0.0237 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.9498 on 49 degrees of freedom
## Multiple R-squared:  0.4794, Adjusted R-squared:  0.4475 
## F-statistic: 15.04 on 3 and 49 DF,  p-value: 4.551e-07
```

```
plot(ch3,which=2)
```



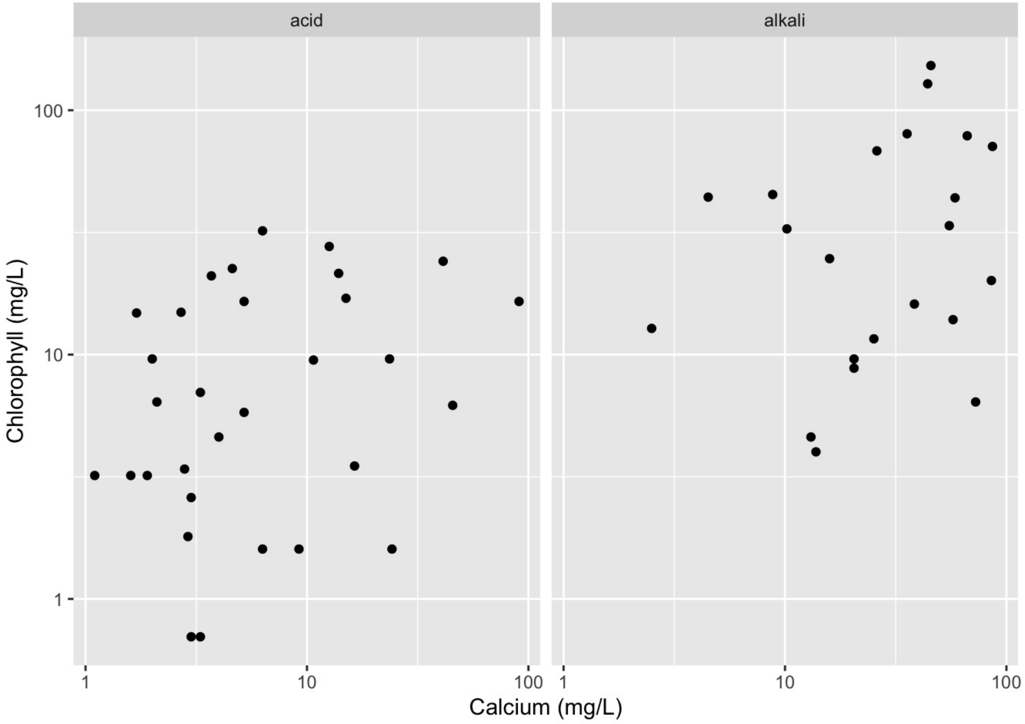
ch3 is the (or a) minimal adequate model. Some residuals a bit odd up the top, but can live with this

Perhaps not surprising that alkalinity drops out since related to both pH and Ca concentration

The interaction here means that the effect of clacium concentration on chlorophyll is only evidenced in alkaline lakes

One way to plot this is to plot for pH <7 and pH ≥ 7. In alkali lakes the amount of chlorophyl is higher.

```
# a column for whether acid or alkali (there are a few ways to do this, eg in tidyverse case_when)

chlorophyll$acid <- factor("acid",levels=c("acid","alkali"))
chlorophyll$acid[chlorophyll$pH >= 7] <- "alkali"

chloro.plot1 <- ggplot(data = chlorophyll, aes(x=Calcium,y=Chlorophyll))

chloro.plot1 + geom_point() +
  xlab("Calcium (mg/L)") + ylab("Chlorophyll (mg/L)") +
  scale_x_log10() + scale_y_log10() +
  facet_wrap(~acid)
```



We’re sometimes asked how to present this in an assessment. In a sentence say that you investigated using logs, give the maximal model you started with and then a sentence on model deletion to arrive at a minimal adequate model. In addition to the figure above (to which you’ll need a suitable figure legend below the figure), you can present the minimal adequate model as a table, something like:

Term	Coefficient (± s.e.)	F statistic	p-value
Intercept	0.788 (± 1.395)		
pH	0.218 (± 0.248)		
log(Calcium)	-1.484 (± 0.652)		
pH:log(Calcium)	0.220 (± 0.094)	$F_{1,49}$=5.454	0.024

Possibly giving the adjusted R^2 as well.

Turtles

> **TASK**
>
> Sex determination in turtles is dependent on the temperature at which eggs are incubated at. Consider the data in the turtles.txt file and describe this relationship.

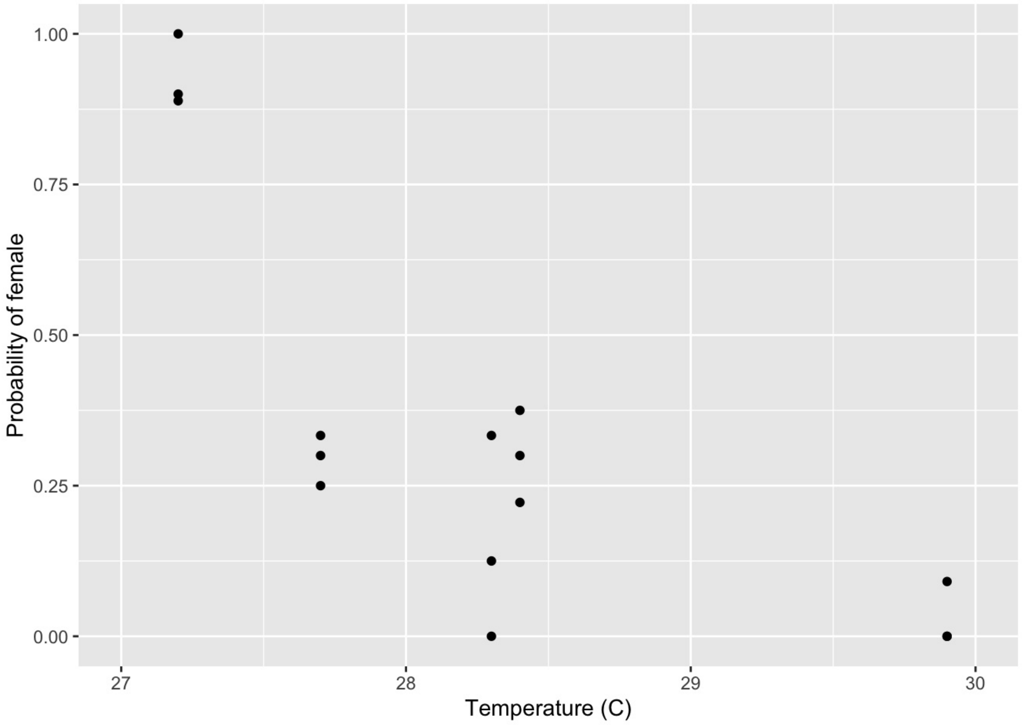
Read in the data, view it and plot.

```
turtle <- read.table("turtle.txt",header=TRUE)

head(turtle)
```

```
##   temp male female
## 1 27.2    1      9
## 2 27.2    0      8
## 3 27.2    1      8
## 4 27.7    7      3
## 5 27.7    4      2
## 6 27.7    6      2
```

A simple plot, i.e. of proportion against temperature. You can also create another column in the dataframe for the proportion:

```
turtle.plot1 <- ggplot(data=turtle,aes(x=temp,y=female/(male+female)))

turtle.plot1 + geom_point() +
  xlab("Temperature (C)") + ylab("Probability of female") +
  xlim(c(27, 30)) + ylim(c(0, 1))
```



So the data are discrete; either male or female. This is similar to the o-ring example where data were binomial. Here it’s male/female instead of damaged/not damaged. The difference being that the total number of samples at each temperature can vary.

```
turtle.mod <- glm(cbind(female,male)~temp,data=turtle,family="binomial")
summary(turtle.mod)
```

```
## 
## Call:
## glm(formula = cbind(female, male) ~ temp, family = "binomial", 
##     data = turtle)
## 
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)    
## (Intercept)  61.3183    12.0224   5.100 3.39e-07 ***
## temp         -2.2110     0.4309  -5.132 2.87e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 74.508  on 14  degrees of freedom
## Residual deviance: 24.942  on 13  degrees of freedom
## AIC: 53.836
## 
## Number of Fisher Scoring iterations: 5
```

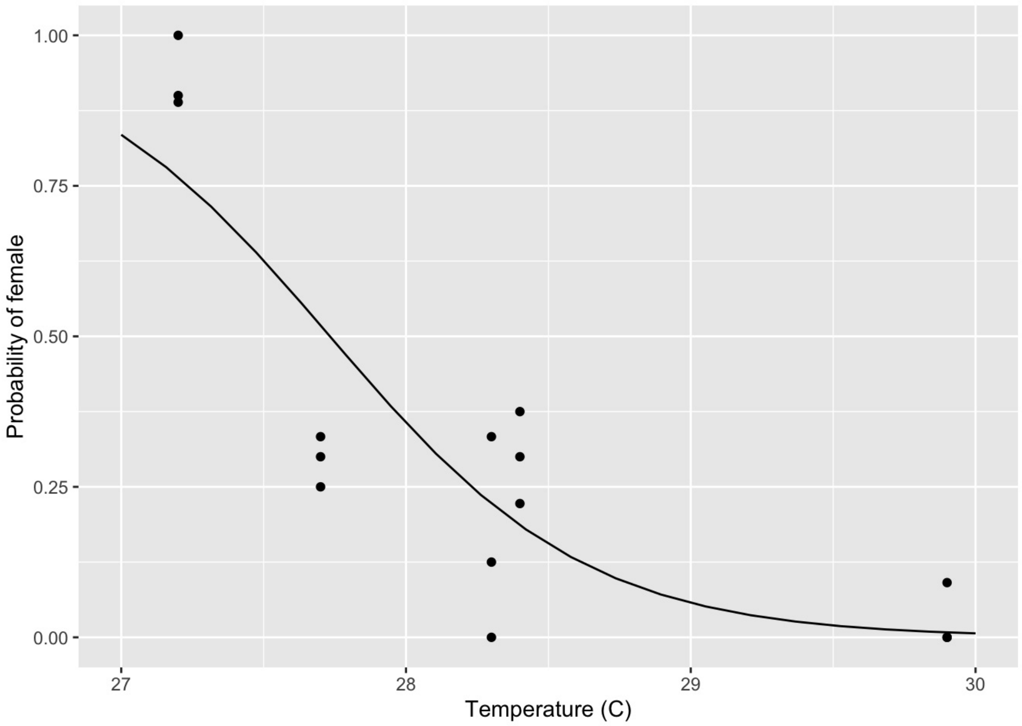
```
drop1(turtle.mod,test="Chisq")
```

```
## Single term deletions
##
## Model:
## cbind(female, male) ~ temp
##        Df Deviance     AIC    LRT  Pr(>Chi)
## <none>      24.942  53.836
## temp    1   74.508 101.402 49.566 1.919e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

So a strong effect of temperature

Bonus points for generating a plot (much like the orings sample code)

```
turtle.pred <- data.frame(temp = seq(27,30,length=20))
turtle.pred$pred.female <- predict(turtle.mod, type = "response", newdata = turtle.pred)


turtle.plot1 + geom_point() +
  xlab("Temperature (C)") + ylab("Probability of female") +
  xlim(c(27, 30)) + ylim(c(0, 1)) +
  geom_line(data=turtle.pred,aes(x=temp,y=pred.female))
```



One tricky thing with binomial data is to interpret the sign of the coefficient, ie do you get more males or females with higher temperatures? The plot helps to keep you straight. From the help page to `glm` it says the response can be specified as a two column matrix with ‘successes’ and ‘failures’; this is the `cbind(female,male)` bit of the formula. (Here I’ve put getting a female as a ‘success’, but if you’ve put `cbind(male,female)` the model fits a coefficient of the same magnitude but opposite sign.)

Phage

> **TASK**
>
> Wild-type and a mutant bacterial strain were tested for their ability to gain antibiotic resistance following infection by a lysogenic phage (which integrates into the host genome and hence alter the regulation of host genes). Counts of antibiotic resistant bacteria in wildtype and mutant strains following phage infection in replicate experiments were conducted.
>
> Consider the data in phage.txt and perform an analysis to test for a difference in antibiotic resistance between wildtype and mutant bacterial strains.

Read in the data and check it.

```
phage <- read.table("phage.txt",header=TRUE)
head(phage)
```

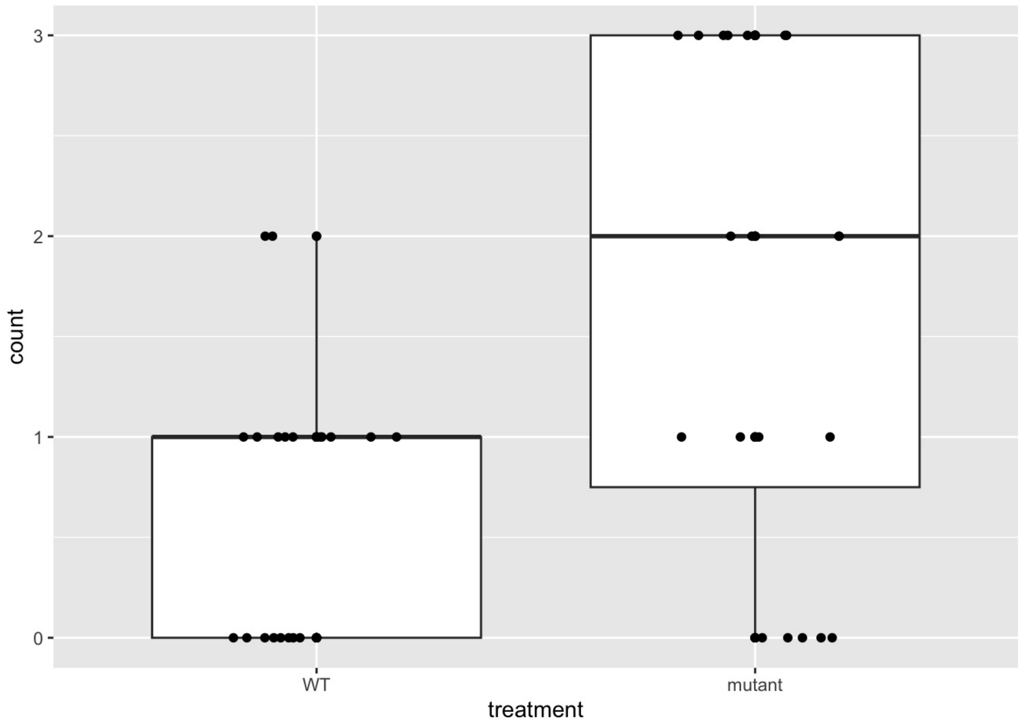
```
##   treatment count
## 1        WT     0
## 2        WT     1
## 3        WT     1
## 4        WT     0
## 5        WT     1
## 6        WT     1
```

```
summary(phage)
```

```
##   treatment             count      
##  Length:40          Min.   :0.000  
##  Class :character   1st Qu.:0.000  
##  Mode  :character   Median :1.000  
##                     Mean   :1.175  
##                     3rd Qu.:2.000  
##                     Max.   :3.000
```

Specify treatment as a factor and do a plot:

```
phage$treatment <- factor(phage$treatment,levels=c("WT","mutant"))

phage.plot1 <- ggplot(data=phage,aes(x=treatment,y=count))

#a boxplot
phage.plot1 + geom_boxplot() +
  # you can also plot the points but give a bit of horizontal 'jitter' to see the individual points
  geom_point() + geom_jitter(height=0,width=0.2)
```



You could also look at the data with `View` or plot histograms to see the data.

Here you should spot that the data are discrete integers, but are only bound on the left not the right. There is also likely to be some underlying probability of resistance emerging and you're counting the number of times it emerges. Hence these data are Poisson distributed.

```
phage.mod <- glm(count~treatment,data=phage,family="poisson")
summary(phage.mod)
```

```
##
## Call:
## glm(formula = count ~ treatment, family = "poisson", data = phage)
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -0.3567     0.2673  -1.335  0.18202
## treatmentmutant   0.8575     0.3190   2.688  0.00718 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 47.618  on 39  degrees of freedom
## Residual deviance: 39.713  on 38  degrees of freedom
## AIC: 108.34
##
## Number of Fisher Scoring iterations: 5
```

```
drop1(phage.mod,test="Chisq")
```

```
## Single term deletions
##
## Model:
## count ~ treatment
##           Df Deviance    AIC    LRT Pr(>Chi)
## <none>         39.713 108.34
## treatment  1   47.618 114.24 7.9051  0.00493 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note that `drop1` with the argument `test="Chisq"` gives a more accurate p-value than the output of `summary` .

Hence we'd conclude that antibiotic resistance emerges more often in the mutant than the wild-type strain.