BIO782P ANOVA and Regression cribsheet

Rob Knell / Joe Parker

7 November 2018

ANOVA

Understanding how animals respond in the face of changing temperature is of obvious importance nowadays. One of the important questions that we need to understand is whether unpredictable changes in temperature have different effects from predictable, constant ones. This question was examined by Manenti et al. (2014) who described an experiment in which 25 isofemale lines of ***Drosophila simulans*** were reared in one of three treatments: a constant temperature (C) of 23°, a predictably fluctuating temperature (F) which rose to 28° during the day and declined to 13° at night, and an unpredictably fluctuating temperature (U) which rose to a randomly determined value between 23° and 28° during the day and fell to a randomly determined value between 23° and 13° at night. A variety of measurements were taken of flies from each treatment.

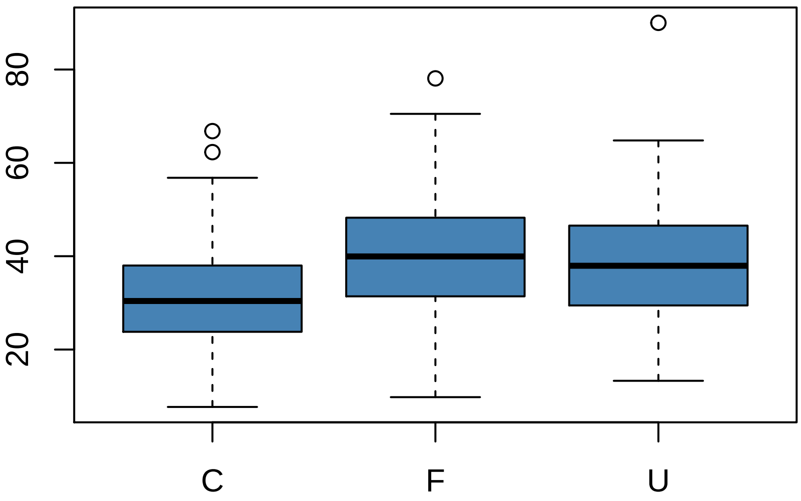
The file manenti_heatshock.txt contains data for one of these measurements: time to heat knock-down, which is the period for which the flies could withstand a temperature of 37.5° before becoming incapacitated. Each measurement is a mean value from about 10 measurements, each from a separate isofemale line.

1) Set up an object in R called "Heatshock" and read the data from the manenti_heatshock.txt text file into it using the `read.table()` function. You might find the chapter on importing data in Introductory R useful here.
2) Check that the data have been imported properly using the str() function. You should have a data frame with two variables: treatment, which indicates the temperature treatment that each group of insects was exposed to, and heatshock.time which is the mean time that insects from a particular line were able to withstand a high temperature for. Treatment should be a factor with three levels, heatshock.time should be a numeric variable.
3) To get an idea of what your data look like, draw a boxplot with the factor levels on the x-axis and longevity on the y-axis. NB to refer to the variables within the data frame you'll either need to attach the data frame or refer to them using the name of the data frame and then the name of the variable with a dollar sign between them, e.g. Heatshock$treatment.

```r
### Import data and save to object called "Heatshock"

Heatshock <- read.table("Manenti_heat_shock.txt", header=TRUE)

### Check data

str(Heatshock)
```

```
## 'data.frame':    630 obs. of  2 variables:
##  $ treatment     : Factor w/ 3 levels "C","F","U": 1 1 1 1 1 1 1 1 1 1 ...
##  $ heatshock.time: num  32.3 12.2 21.8 40 36 37 25.3 28.2 18.4 22.7 ...
```

```r
### Draw boxplot

boxplot(Heatshock$heatshock.time~Heatshock$treatment, col="steelblue")
```



4) What do you see? Is there anything that might make an ANOVA unsuitable for analysing these data as they are?

Boxplots are nicely symmetrical and there seems to be little difference in variance between treatments. Perhaps a few slight outlying data points but nothing to worry about at this point.

5) We're going to carry out an ANOVA on these data in three ways: by calculating it, by using the aov() function and by using the lm() function. Let's start by doing it the hard way, by hand.

   Firstly you need to calculate the total sums of squares for heatshock.time. Remember that this is calculated as each number in the vector, minus the overall mean value for the vector and squared. Make sure you square the numbers before you sum them: the sum of the squared differences is not the same as the sum of the differences, squared. Set up an object called SStotal to store this number in.

   Now calculate it a different way using the `var()` function and multiplying the variance by the degrees of freedom. The two numbers should be the same.

```
### Calculate SSTotal

attach(Heatshock)

SStotal<-sum((heatshock.time-mean(heatshock.time))^2)

SStotal2<-var(heatshock.time)*(length(heatshock.time)-1)

SStotal
```

```
## [1] 98214.16
```

```
SStotal2
```

```
## [1] 98214.16
```

Now to calculate the error sums of squares. This is the residual variance left after the effect of the factor (in this case the temperature treatment) has been removed. You need to start by calculating the mean value for heatshock.time for each of the three factor levels: the easy way to do this is by using the `tapply()` function. Look up the help file and try to get it to work, and once you're there store the output as another object.

```
heatshock.means <- tapply(heatshock.time, treatment, mean)

heatshock.means
```

```
##        C        F        U
## 31.13073 39.64773 38.55833
```

Once you have the means for each factor level then you can calculate the error sum of squares by working out the sum of the squared deviations from the factor mean for each factor level, and adding the sums of squares for each factor level together. Store this value as an object called SSerror.

```
### Calculate SSerror
SSerrorC <- sum((heatshock.time[treatment =="C"] - heatshock.means[1])^2)
SSerrorF <- sum((heatshock.time[treatment =="F"] - heatshock.means[2])^2)
```

```
SSerrorU <- sum((heatshock.time[treatment =="U"] - heatshock.means[3])^2)

SSerror <- SSerrorC + SSerrorF + SSerrorU

SSerror
```

```
## [1] 88947.06
```

The total sums of squares is equal to the error sums of square plus the treatment sums of squares, so we can work this last value out by subtracting SSerror from SStotal.

```
### Calculate SStreatment

SStreatment <- SStotal - SSerror

SStreatment
```

```
## [1] 9267.098
```

Having calculated these values we can fill in an ANOVA table.

Source of Variation	df	Sum of squares	Mean squares	F	p
Treatment	2	9267	4633.55	32.663	
Error	627	88947	141.86		
Total	629	98214			

Now that we know the calculated test statistic, F, we want to know if there is a significant effect. In other words, we want to know what the probability is of getting the observed value of F given that our degrees of freedom are Treatment df and Error df. R has all sorts of statistical distributions built in and we can calculate out probability using the pf() function. This gives us the probability density of F, in other words the probability of observing that value or a smaller one given the degrees of freedom. We want the probability of observing our F value or bigger, so we subtract the value pf() from one to give us our p-value:

```
1-pf(32.663, 2, 627)
```

```
## [1] 3.208545e-14
```

6) Now use R to carry out an ANOVA on the same data using the lm() function. You'll need to set up a new object and allocate the output of the lm() function to it, and then you can get an ANOVA table using the anova() function. It should be the same as the one you've constructed above.

```
model1<-lm(heatshock.time ~ treatment)

anova(model1)
```

```
## Analysis of Variance Table
##
## Response: heatshock.time
##            Df Sum Sq Mean Sq F value    Pr(>F)
## treatment   2   9267  4633.5  32.663 3.208e-14 ***
## Residuals 627  88947   141.9
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- What is your null hypothesis?

The three treatment groups came from populations with the same means

- What is the alternative hypothesis?

The three treatment groups came from populations with different means

- What do you conclude from the results of your ANOVA?

Reject H0, accept H1. At least one of the treatment group means is significantly different from at least one other

7) Use `summary()` to produce a summary table for your fitted ANOVA model which you produced using `lm()`.

```
summary(model1)
```

```
##
## Call:
## lm(formula = heatshock.time ~ treatment)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.848  -8.056  -0.345   7.569  51.442
##
## Coefficients:
##             Estimate Std. Error t value          Pr(>|t|)
## (Intercept)  31.1307     0.8067  38.591           < 2e-16 ***
## treatmentF    8.5170     1.1382   7.483 0.000000000000247 ***
## treatmentU    7.4276     1.1788   6.301 0.000000000557538 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.91 on 627 degrees of freedom
## Multiple R-squared:  0.09436,    Adjusted R-squared:  0.09147
## F-statistic: 32.66 on 2 and 627 DF,  p-value: 3.208e-14
```

Try to work out what the coefficients mean: you might find the section on sexual signalling in yeast in Introductory R (the latest version) helpful.

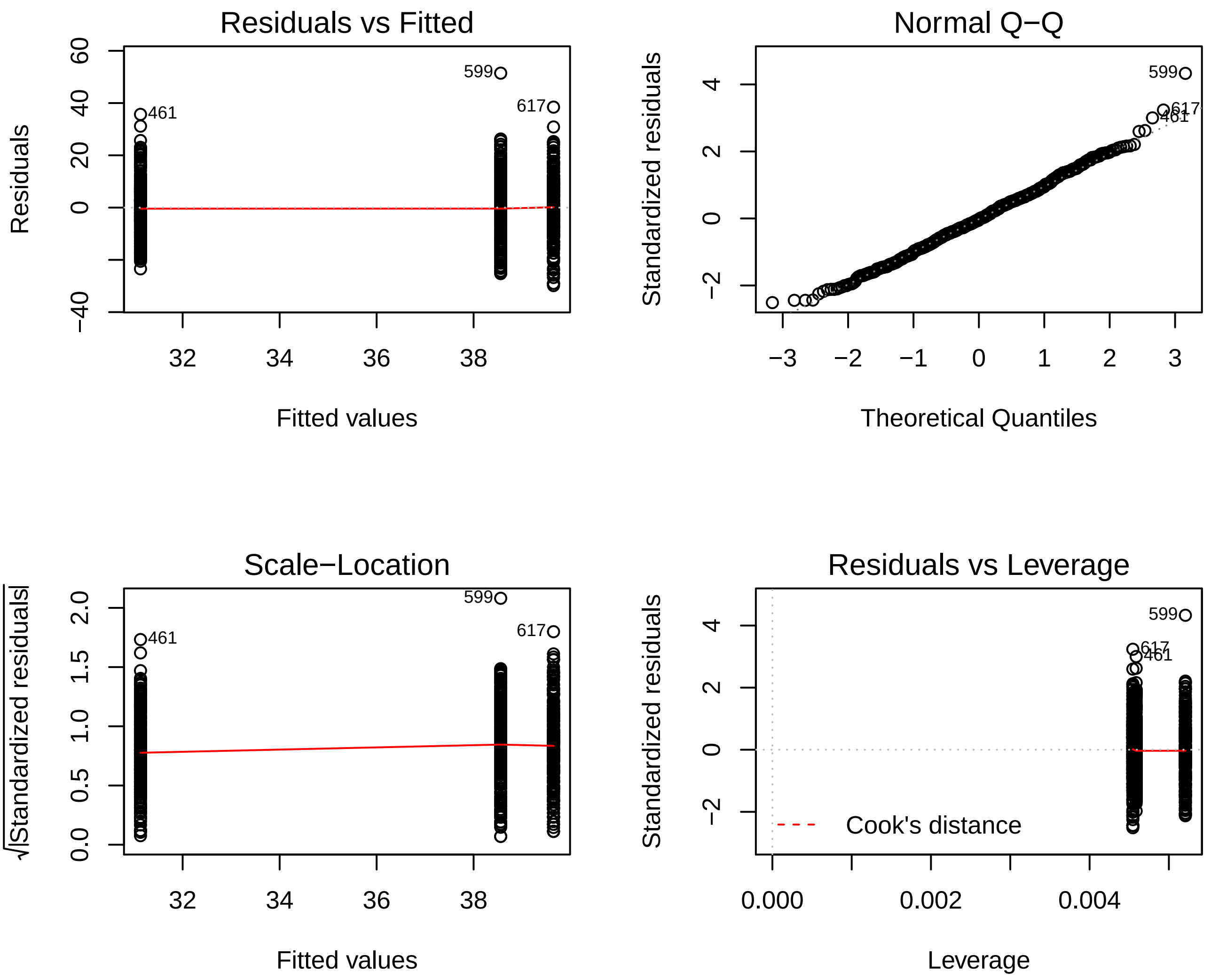
The value for “intercept” is the estimated mean value for treatment “C” which comes first because it is first alphabetically. The p-value etc. associated with it simply means that it is different from zero.

The value for “F” is the difference between the mean of “C” and the mean of “F”. This value is a lot bigger than its standard error and the associated low p-value tells us that the mean for “F” is significantly different from the mean for “C”.

The value for “U” is also different from the value for “C” - however, it is quite similar to the value for “F” and the estimated mean for “F” is within one SE of that for “U”. This tells us that the means for “F” and “U” are very unlikely to be signficantly different from each other.

8) Use plot() to produce some diagnostic plots for your fitted ANOVA. Have a look: does your model have an acceptable fit?

```
par(mfrow = c(2,2))  ### Write plots into a 2x2 matrix

plot(model1) ### Diagnostic plots for model 1
```



```
par(mfrow=c(1,1))  ### Revert back to normal plotting
```

Have a look: does your model have an acceptable fit?

It's basically fine

9) What do you conclude?

Flies exposed to either a predictably or an unpredictably fluctuating temperature can resist a temperature of 37.5 degrees for roughly 7-8 minutes longer than flies kept in a constant 23 degree treatment.

Manenti, T., Sørensen, J.G., Moghadam, N.N., and Loeschcke, V. (2014). Predictability rather than amplitude of temperature fluctuations determines stress resistance in a natural population of Drosophila simulans. J. Evol. Biol. 27, 2113–2122.

Update: Populating a table of observed, expected ('predicted', or 'fitted') values and residuals (or 'error')

Some of you wanted to repeat filling out our table of values as I showed in labs. Here's how to do it. First let's look at the basic data:

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

```
##   treatment heatshock.time
## 1         C           32.3
## 2         C           12.2
## 3         C           21.8
## 4         C           40.0
## 5         C           36.0
## 6         C           37.0
```

Now we can calculate the grand mean (overall mean), treatment means, and from them the total, treatment and error SS for the F-ratio in the ANOVA table:

```
# set up a column with the mean for all the data. It will be the same in each row
Heatshock$grand_mean=mean(Heatshock$heatshock.time)

# calculate the total variance as observed - expected (where 'expected' is the grand mean)
Heatshock$deviations=Heatshock$heatshock.time-Heatshock$grand_mean

# fit separate means for each treatment; for rows within each treatment this will be the same.
heatshock.means <- tapply(heatshock.time, treatment, mean)

# set up another column holding the expected value by treatment and fill it (depending on which treatment an obser
Heatshock$treatment_means=heatshock.means[1]
Heatshock$treatment_means[Heatshock$treatment=="F"]=heatshock.means[2]
Heatshock$treatment_means[Heatshock$treatment=="U"]=heatshock.means[3]

# calculate the proportion of variation explained by the 'treatment' prediction in each observation. This is the g
Heatshock$treatment = Heatshock$treatment_means - Heatshock$grand_mean

# calculte the proportion of variance remaining in each point, *after* treatment means have been fitted. We might
Heatshock$error = Heatshock$heatshock.time-Heatshock$treatment_means

# calculate SS for each of the quantities above by squaring each row's value and them summing down the column.
totalSS = sum(Heatshock$deviates^2)
treatmentSS = sum(Heatshock$treatment^2)
totalSS = sum(Heatshock$deviations^2)
errorSS = sum(Heatshock$error^2)

#see what we've got
totalSS
```

```
## [1] 98214.16
```

```
treatmentSS
```

```
## [1] 9267.098
```

```
errorSS
```

```
## [1] 88947.06
```

```
head(Heatshock)
```

```
##   treatment heatshock.time grand_mean  deviations treatment_means
## 1 -5.237837           32.3   36.36857  -4.0685714        31.13073
## 2 -5.237837           12.2   36.36857 -24.1685714        31.13073
## 3 -5.237837           21.8   36.36857 -14.5685714        31.13073
## 4 -5.237837           40.0   36.36857   3.6314286        31.13073
## 5 -5.237837           36.0   36.36857  -0.3685714        31.13073
## 6 -5.237837           37.0   36.36857   0.6314286        31.13073
##        error
## 1   1.169266
## 2 -18.930734
## 3  -9.330734
## 4   8.869266
## 5   4.869266
## 6   5.869266
```

Regression

Cope's rule refers to the tendency of animal body sizes to increase in time with increasing age of the lineage. De Souza and Santucci (2014) tested this using a set of data from the dinosaur clade called the Titanosaurs. This is a group of sauropods that included some of the largest animals known in the history of the Earth (e.g. ***Argentinosaurus***) and which were widespread and diverse during the Cretaceous period. Most Titanosaurs are known from only a few bones, and the massive limb bones are particularly common. De Souza and Santucci assembled measurements of limb bones from 46 species of Titanosaur. Measurements of both humerus and femur were available for 20 of these, with 12 being represented only by the humerus and 14 by the femur only. In order to carry out an analysis of how the body size of these animals varied with time, it is necessary to estimate the femur sizes for those animals that are only represented by the humerus, and to do this we can use a linear regression to work out the expected values for each of these missing data points.

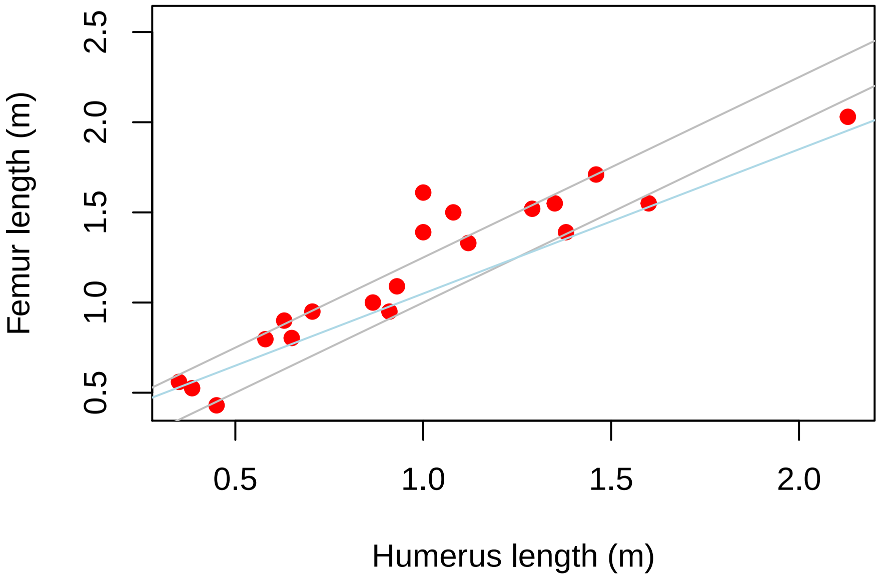
1. Load the dataset called `Titanosaurs.txt` into R – save it as an object called (for example) Titanosaurs. Use str() to check the structure of the new data frame – there should be a variable called "Taxa" which is the species name, the one called "Mean.time.MA" which is the age of the fossil in millions of years, then "Femur" and "Humerus" which are the lengths of the respective bones in metres.

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

str(Titanosaurs)
```

```
## 'data.frame':    46 obs. of  4 variables:
##  $ Taxa        : Factor w/ 46 levels "Aeolosaurus.maximus",..: 13 5 31 3 42 23 35 30 22 6 ...
##  $ Mean.time.Ma: num  151.2 69 113.4 69 74.8 ...
##  $ Femur       : num  2.03 0.803 0.95 1.61 0.9 ...
##  $ Humerus     : num  2.13 0.65 0.705 1 0.63 0.45 0.58 0.385 0.91 1.35 ...
```

2. Plot a scatterplot to show how the length of the two limb bones relate to each other. Remember we are going to try to predict femur length from humerus length so decide which measurement should go on the y- and x- axes on this basis.

```
plot(Titanosaurs$Humerus,Titanosaurs$Femur, xlab = "Humerus length (m)", ylab = "Femur length (m)",col="red",pch=1

# We can try a few lines out. The R fuction for this is abline() - e.g lines of the form y = a + bx
abline(0,1,col="grey")
# Not quite - try translating it up a bit
abline(0.25,1,col="grey")
# Maybe a bit shallower?
abline(0.25,0.8,col="light blue")
```



```
# Hmm..
```

3. Have a look at the data. Can you see anything that might mean that you should be cautious in using a linear regression on these data? If so, can you fix it in any way?

Possible hint of curvature but not enough to be sure without actually fitting the line

4. Fit a linear regression to the data using the lm() function and save the fitted model to an object in the R workspace.

```
attach(Titanosaurs)
model2<-lm(Femur~Humerus)
```

You can bring up a summary of the fitted model using the `summary()` function as before. What information does this give you that you didn't have before?

```
summary(model2)
```

```
##
## Call:
## lm(formula = Femur ~ Humerus)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.25714 -0.11233 -0.01991  0.05536  0.42467
##
## Coefficients:
##             Estimate Std. Error t value      Pr(>|t|)
## (Intercept)  0.27952    0.09192   3.041       0.00703 **
## Humerus      0.90581    0.08460  10.707 0.00000000309 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1666 on 18 degrees of freedom
##   (26 observations deleted due to missingness)
## Multiple R-squared:  0.8643, Adjusted R-squared:  0.8567
## F-statistic: 114.6 on 1 and 18 DF,  p-value: 0.000000003092
```

lots - r-squared value, significance test etc.

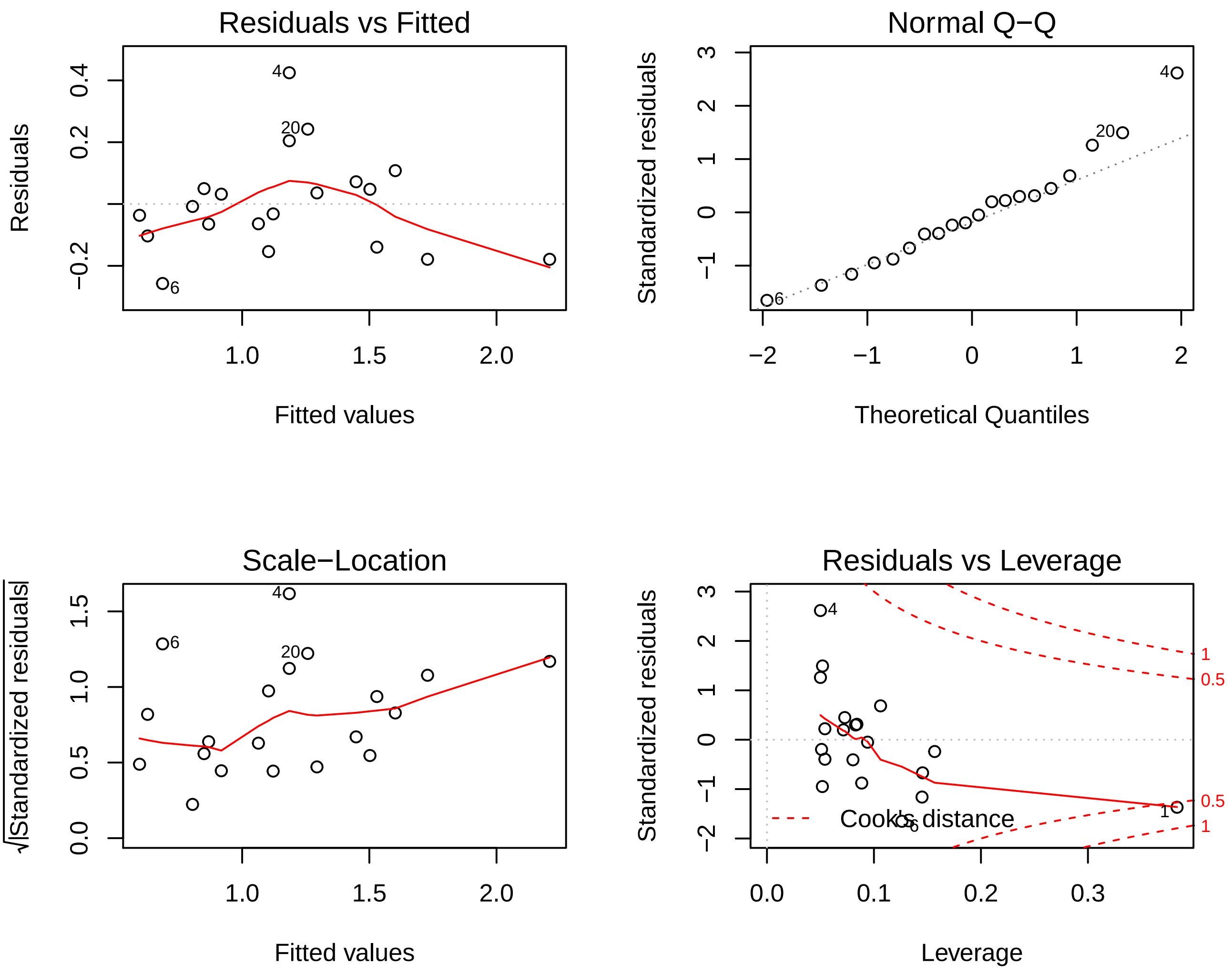
5. An alternative way to check the statistical significance of a regression is to calculate how much of the variance in the data is explained by the regression line and compare that to the error variance in exactly the same way as we do for an ANOVA. You can do this by using the anova() function.

```
anova(model2)
```

```
## Analysis of Variance Table
##
## Response: Femur
##           Df Sum Sq Mean Sq F value         Pr(>F)
## Humerus    1 3.1806  3.1806  114.63 0.000000003092 ***
## Residuals 18 0.4994  0.0277
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

6. Check the diagnostic plots for the regression using plot() as before – in particular pay attention to the plot of residual versus predicted values. Do you see anything that causes concern?

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

plot(model2)
```



```
par(mfrow = c(1,1))
```

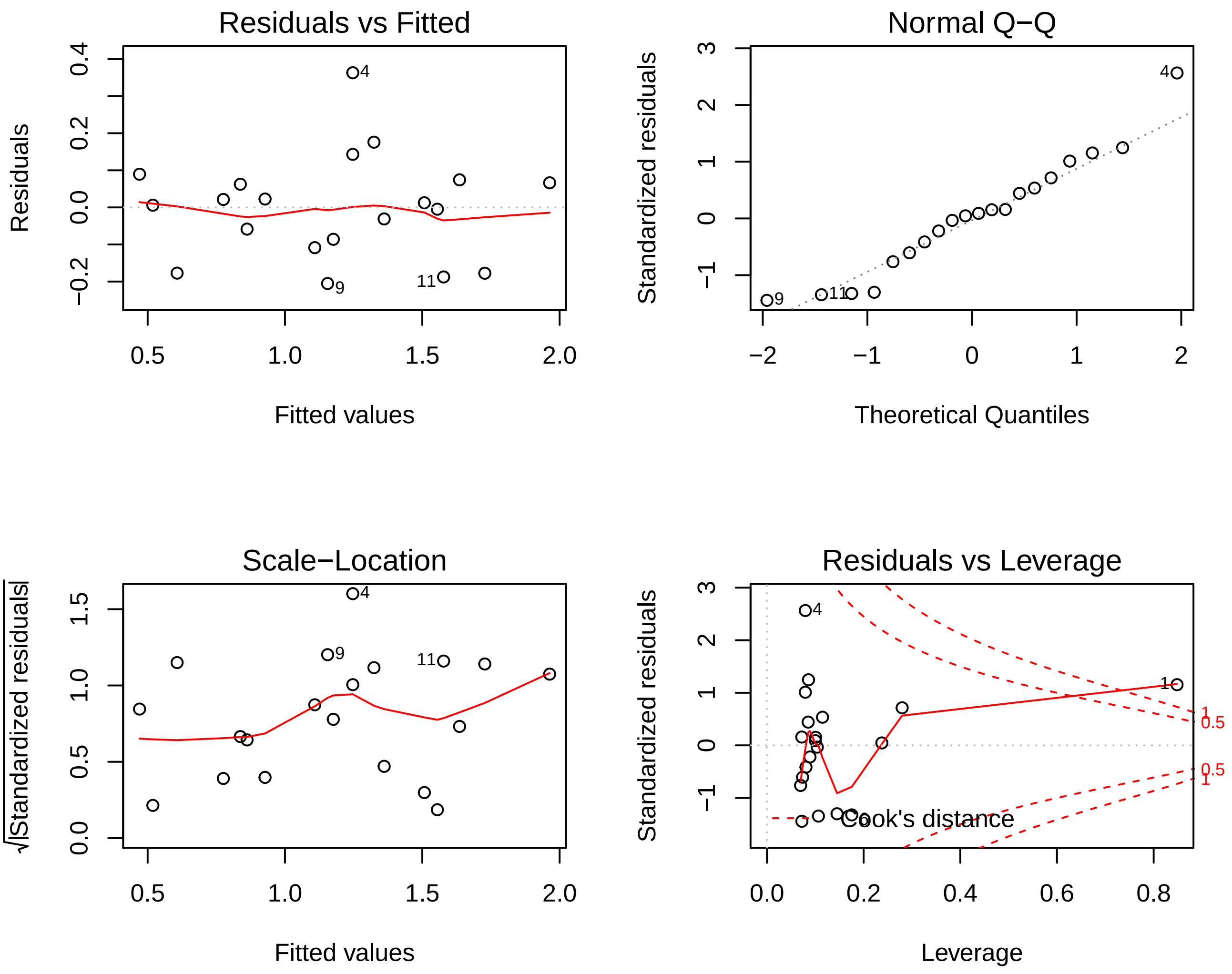
Residual versus fitted values plot shows some indication of curvature

7. Instead of fitting a straight line to your data, try fitting a curve. I've given you the code to fit a second order polynomial:

```
model3 <- lm(Femur~Humerus + I(Humerus^2))
```

8. Check your diagnostic plots and compare them with the plots for the simple linear model. Also have a look at the table produced by summary() – is the addition of the quadratic term justified?

```
par(mfrow = c(2,2))
plot(model3)
```



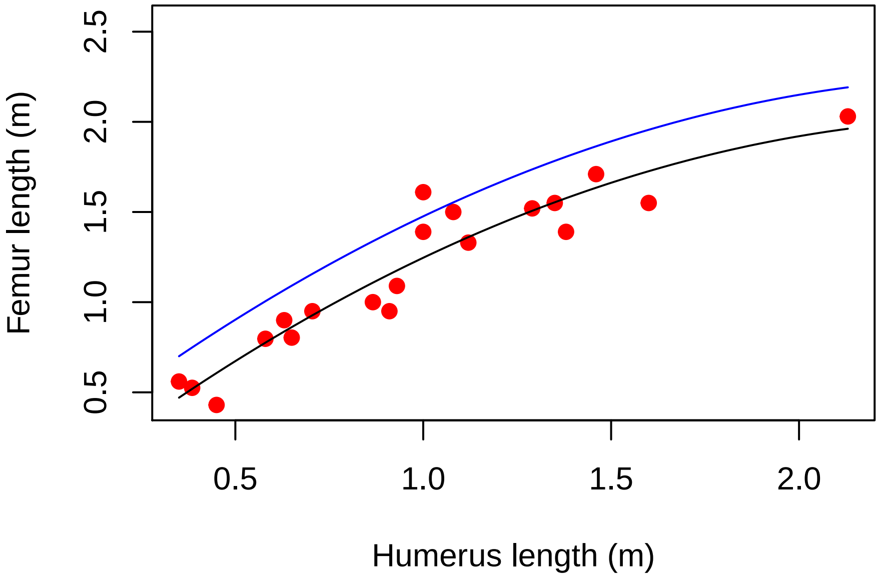
```
par(mfrow = c(1,1))

summary(model3)
```

```
##
## Call:
## lm(formula = Femur ~ Humerus + I(Humerus^2))
##
## Residuals:
##      Min       1Q  Median      3Q     Max
## -0.2053 -0.0917  0.0092  0.0683  0.3631
##
## Coefficients:
##              Estimate Std. Error t value  Pr(>|t|)
## (Intercept)  -0.05772    0.16059  -0.359    0.7237
## Humerus       1.61931    0.30228   5.357 0.0000523 ***
## I(Humerus^2) -0.31469    0.12916  -2.436    0.0261 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1476 on 17 degrees of freedom
##   (26 observations deleted due to missingness)
## Multiple R-squared:  0.8994, Adjusted R-squared:  0.8876
## F-statistic:    76 on 2 and 17 DF,  p-value: 0.000000003324
```

Diagnostic plots now look better. In the summary table you can see that the estimated coefficient for the quadratic term is significantly different from zero. Both of these lend support to using the model with the curve as a description of the relationship between Humerus and Femur rather than the simple straight line model. However, the residuals versus leverage plot now shows one data point with rather high leverage - this is the very large individual. An alternative to the curve fit would be to remove this data point and just use a *straight line. This second option is problematic in this particular case however because we want to predict femur length from individuals including two with humerus lengths of 1.6m or more - removing this large data point would mean that we'd essentially be extrapolating outside our data range for these two and we'd have reduced confidence in our estimates for them.*

9. Plot your data with the fitted curve by using the curve() function to add a curve to a scatterplot.

```
plot(Titanosaurs$Humerus,Titanosaurs$Femur, xlab = "Humerus length (m)", ylab = "Femur length (m)",col="red",pch=1

curve(-0.05772 + 1.619*x - 0.315*x^2, add=TRUE) # code the function for the line eqn' as we go

# code the line equation more explicitly
line_func=function(x){
  0.172 + 1.619*x - 0.315*x^2 #not quite the fitted values; I've translated it up a bit for clarity
}

curve(line_func,add=T,col="blue") # plot curve with function explicitly defined.
```



10. Now that you have calculated your slope and intercept you can use the equation for the curve that relates femur length to humerus length to workout what the predicted values are for the femurs of the titanosaur species that are only represented in this dataset by their humerus measurements.

equation of the curve is y=-0.05772 + 1.619*x - 0.315*x^2 *so just plug the humerus values into this.*

We can (and probably should) see whether the curve is actually any good:

```
anova(model2,model3,test="LRT")
```

```
## Analysis of Variance Table
##
## Model 1: Femur ~ Humerus
## Model 2: Femur ~ Humerus + I(Humerus^2)
##   Res.Df     RSS Df Sum of Sq Pr(>Chi)
## 1     18 0.49942
## 2     17 0.37016  1   0.12926  0.01483 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

De Souza, L.M., and Santucci, R.M. (2014). Body size evolution in Titanosauriformes (Sauropoda, Macronaria). J. Evol. Biol. 27, 2001–2012.