Lab 6: Regression

Complete all of the following questions, adding your inputs as code chunks (enclose within triple accent marks) within Rmarkdown.

The exercises are not marked and will not be factored into your course grade, but it is important to complete them to make sure you have the skills to answer assessment questions. You may consult any resource, including other students and the instructor. Please upload your work via Canvas at the end of the session. Solutions will be posted for you to check your own answers.

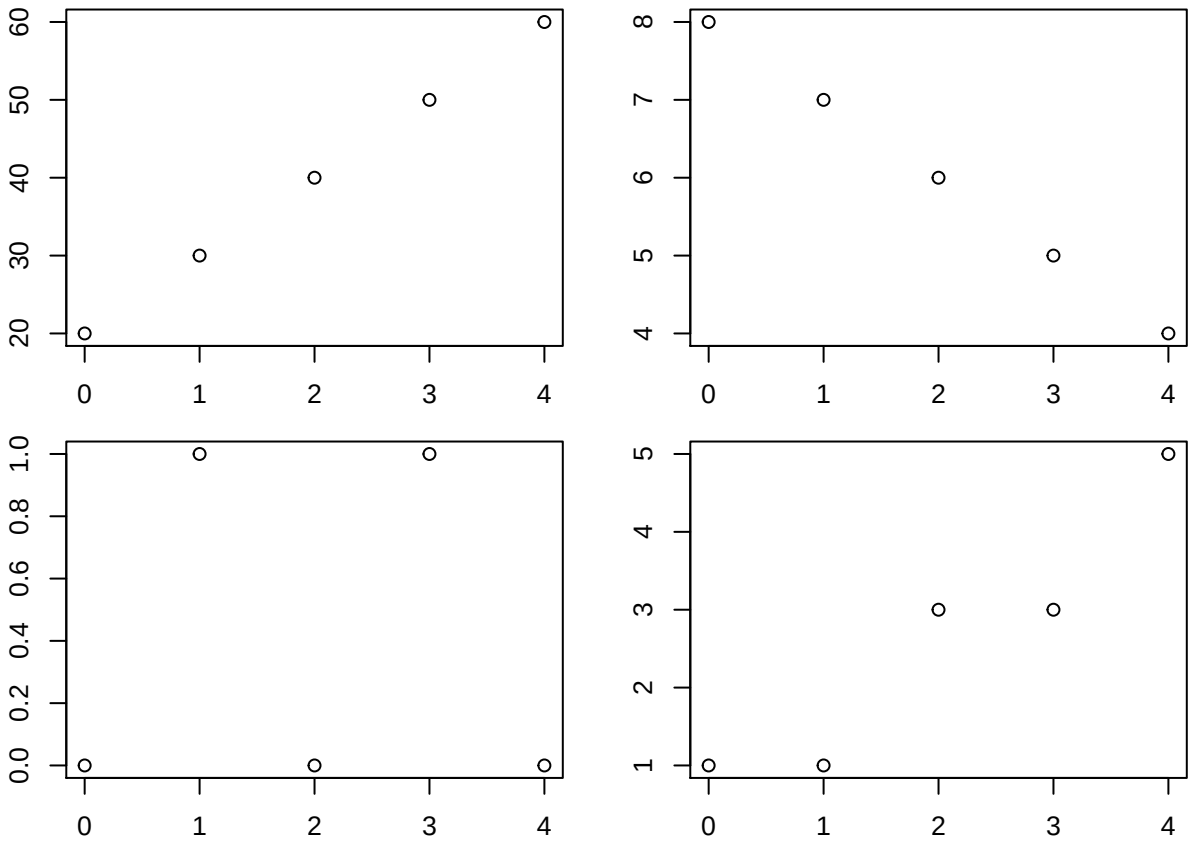
Correlation and Covariance

Consider the following vectors (expressed as rows of a table):

name	values
x	0, 1, 2, 3, 4
y1	20, 30, 40, 50, 60
y2	8, 7, 6, 5, 4
y3	0, 1, 0, 1, 0
y4	1, 1, 3, 3, 5

1. Make a four-panel plot (use `par(mfrow=c(2,2))`) of x versus y1, x versus y2, x versus y3, and x versus y4. Take a guess of what the correlation coefficient for each will be. (Be sure to set the plot back to single-panel when you're done.)

```
x = c(0, 1, 2, 3, 4)
y1 = c(20, 30, 40, 50, 60)
y2 = c(8, 7, 6, 5, 4)
y3 = c(0, 1, 0, 1, 0)
y4 = c(1, 1, 3, 3, 5)
par(mfrow=c(2,2))
par(mar=c(2,3,1,1))
plot(x,y1)
plot(x,y2)
plot(x,y3)
plot(x,y4)
```




```
par(mfrow=c(1,1))
par(mar=c(5,4,4,2))
```

2. Calculate the actual (Pearson) correlation coefficient for each of the four vector pairs above (use `cov(x,y)/(sd(x)*sd(y))` or just `cor(x,y)`), and compare it to your by-eye guess.

```
cor(x,y1)
```

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

```
cor(x,y2)
```

```
## [1] -1
```

```
cor(x,y3)
```

```
## [1] 0
```

```
cor(x,y4)
```

```
## [1] 0.9449112
```

Writing a Regression Function

3. Write your own simple linear regression parameter solver by implementing the equations from lecture. The solver should be implemented as a custom R function which accepts two vectors: x and y (which should be of equal length). The function should calculate both means, then calculate SSX and SSXY, then calculate b, then calculate a. Then, it should create a list containing a and b and return that list.

   The equations are:

$$SSXY = \sum_i x_i y_i - n\bar{x}\bar{y} = \sum_i ((x_i - \bar{x})(y_i - \bar{y}))$$

$$SSX = \sum_i x_i^2 - n\bar{x}^2 = \sum_i ((x_i - \bar{x})^2)$$

$$b = \frac{SSXY}{SSX}$$

$$a = \bar{y} - \frac{SSXY}{SSX}\bar{x}$$

```
regression = function(x, y) {
   meanx = mean(x)
   meany = mean(y)
   ssx = sum( (x-meanx)^2 )
   ssxy = sum( (x-meanx)*(y-meany) )
   b = ssxy / ssx
   a = meany-b*meanx
   return(list(a=a,b=b))
}
```

4. Test out your function by giving it the same four vector pairs from #1 and #2 (x as the independent variable and each y as the dependent variable) and confirming that the intercept and slope are sensible.

```
regression(x, y1)
```

```
## $a
## [1] 20
##
## $b
## [1] 10
```

```
regression(x, y2)
```

```
## $a
## [1] 8
##
## $b
## [1] -1
```

```
regression(x, y3)
```

```
## $a
## [1] 0.4
##
## $b
## [1] 0
```

```
regression(x, y4)
```

```
## $a
## [1] 0.6
##
## $b
## [1] 1
```

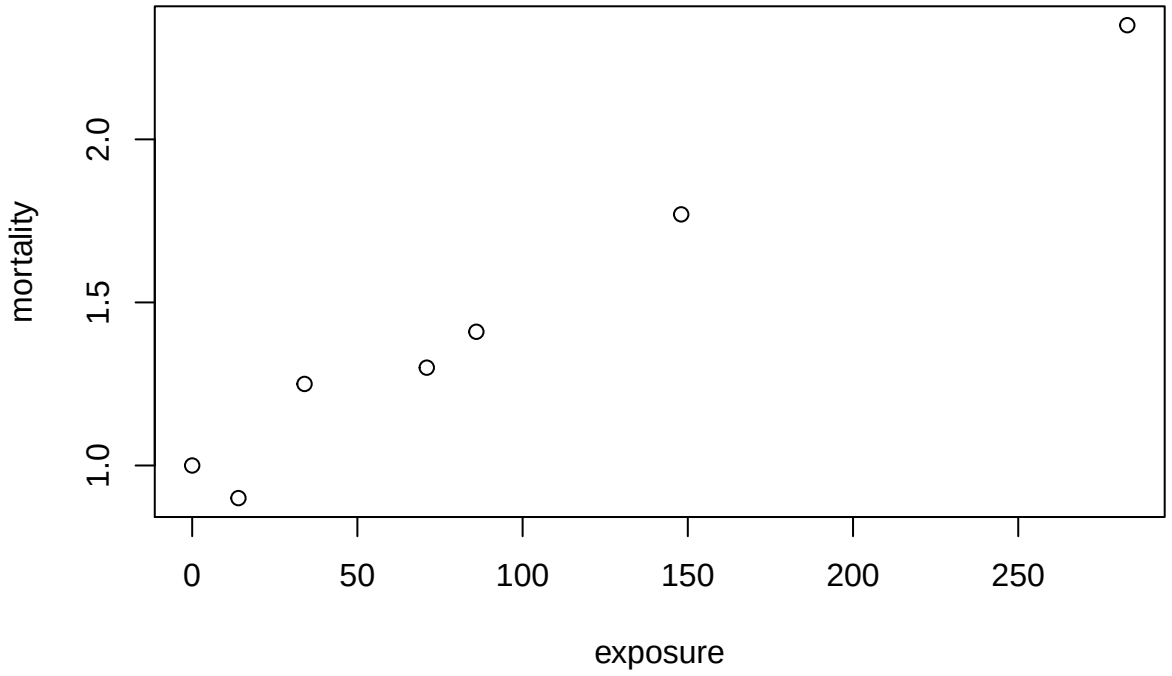
Simple Linear Regression

The following data table compares the radon exposure levels (in WLM) and lung cancer mortality rates (relative to the general population) for several groups of underground miners at various sites. (From Lubin et al., 1995)

Exposure	Mortality
0	1
14	0.90
34	1.25
71	1.30
86	1.41
148	1.77
283	2.35

5. Load these data into two vectors and make a scatter plot of exposure (x-axis) versus mortality (y-axis). Do you think there is a trend? Does it appear to be linear?

```
exposure = c(0,14,34,71,86,148,283)
mortality = c(1,0.9,1.25,1.30,1.41,1.77,2.35)
plot(exposure,mortality)
```



6. Calculate the Pearson correlation coefficient (r) between exposure and mortality using `cor()`, and determine the statistical significance (p-value) of the correlation using `cor.test()`.

```
cor(exposure,mortality)
```

```
## [1] 0.9870158
```

```
cor.test(exposure,mortality)
```

```
##
##  Pearson's product-moment correlation
##
## data:  exposure and mortality
```

```
## t = 13.74, df = 5, p-value = 3.664e-05
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9113347 0.9981607
## sample estimates:
##       cor
## 0.9870158
```

7. Also calculate the Spearman (ρ) and Kendall (τ) correlation coefficients for these variables, and the statistical significance of the correlation using these two methods. Compare these numbers to the Pearson equivalents.

```
cor.test(exposure,mortality, method='spearman')
```

```
##
##  Spearman's rank correlation rho
##
## data:  exposure and mortality
## S = 2, p-value = 0.002778
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.9642857
```

```
cor.test(exposure,mortality, method='kendall')
```

```
##
##  Kendall's rank correlation tau
##
## data:  exposure and mortality
## T = 20, p-value = 0.002778
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## 0.9047619
```

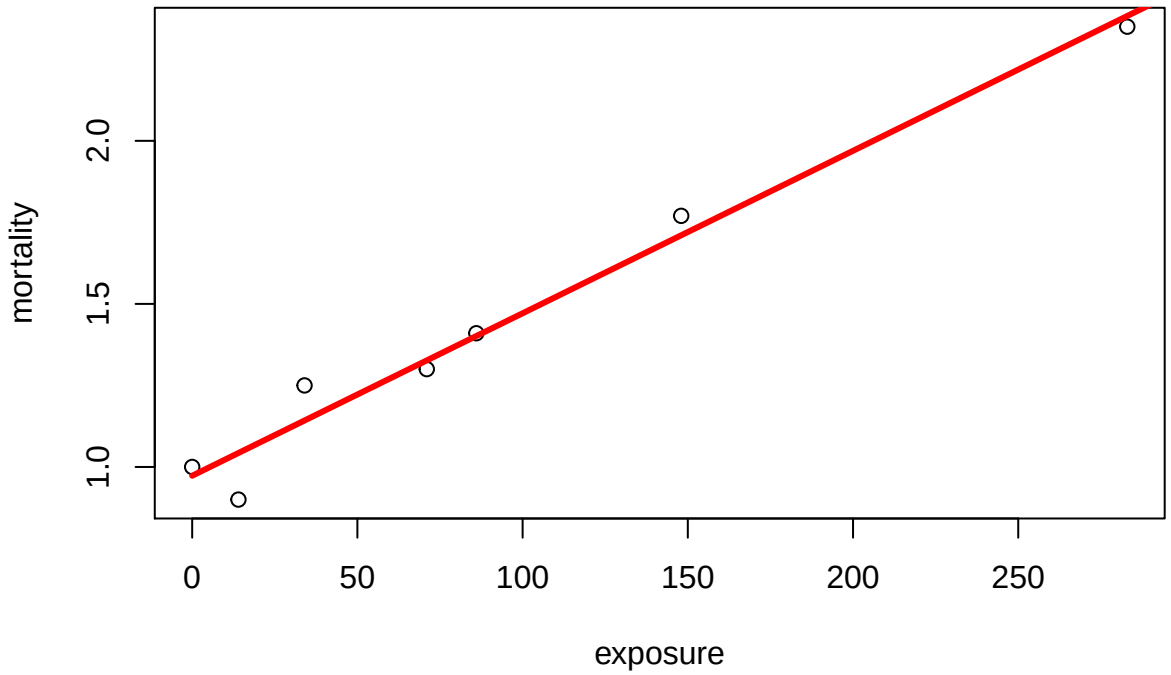
8. Use your regression function from #3 to measure the maximum-likelihood best-fit parameters for a and b.

```
rmpar = regression(exposure,mortality)
rmpar
```

```
## $a
## [1] 0.9731026
##
## $b
## [1] 0.004981575
```

9. Redraw your exposure vs. mortality scatter plot from #4 and add the best-fit line. (Reminder: to add a line/curve to an existing plot, use `lines`.) Does it look like a good fit?

```
xplot = c(0,300)
plot(exposure,mortality)
lines(xplot,rmpar$a+rmpar$b*xplot,col='red',lwd=3)
```



10. Use R's `lm()` function to confirm your estimates of the best-fit parameters a and b.

```
lm(mortality~exposure)
```

```
##
## Call:
## lm(formula = mortality ~ exposure)
##
## Coefficients:
## (Intercept)     exposure
##    0.973103     0.004982
```

11. What are the standard errors on the parameters? What is the significance of the conclusion that b is not zero?

```
summary(lm(mortality~exposure))
```

```
##
## Call:
## lm(formula = mortality ~ exposure)
##
## Residuals:
##         1         2         3         4         5         6         7
##  0.026897 -0.142845  0.107524 -0.026794  0.008482  0.059624 -0.032888
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.9731026  0.0466271   20.87 4.68e-06 ***
## exposure    0.0049816  0.0003625   13.74 3.66e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08731 on 5 degrees of freedom
## Multiple R-squared:  0.9742, Adjusted R-squared:  0.969
```

```
## F-statistic: 188.8 on 1 and 5 DF,  p-value: 3.664e-05
```

The standard errors are SE_a = 0.0466 and SE_b = 0.00036. The significance (p-value) that b is not zero is 3.66e-5.

12. For every 1 WLM increase in radon exposure, by what *percent* does the risk of fatal lung cancer increase over the general population, according to the model? (Hint: this is related to the slope of the model line.)

```
100*0.0049816
```

```
## [1] 0.49816
```

13. Based on the regression, how much radon exposure (in WLM) is necessary to double the risk of fatal lung cancer, compared to someone with zero exposure? (That is, at what level of exposure does the model say that mortality will be 2)?

Invert the relation: y = a + bx becomes x = (y-a)/b.

```
(2-rmpar$a)/rmpar$b
```

```
## [1] 206.1391
```

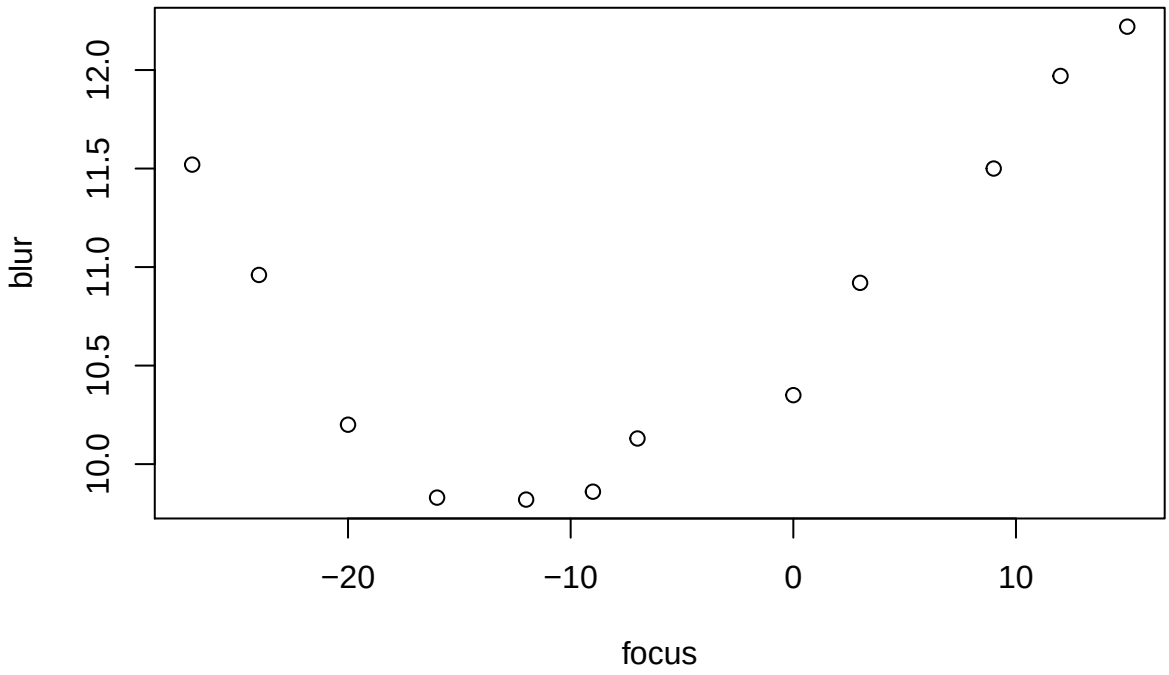
Linear Regression with Polynomials

Consider the following data, which correspond to measurements of the amount a camera image becomes blurred (response variable "blur") as the focus position is adjusted (explanatory variable "focus").

```
focus:  -27,   -24,  -20,   -16,   -12,    -9,    -7,     0,     3,    9,    12,    15
blur:   11.52, 10.96, 10.2,  9.83,  9.82,  9.86, 10.13, 10.35, 10.92, 11.5, 11.97, 12.22
```

14. Enter these data into R as vectors and plot as a scatter plot. Do you think a simple linear model will be appropriate?

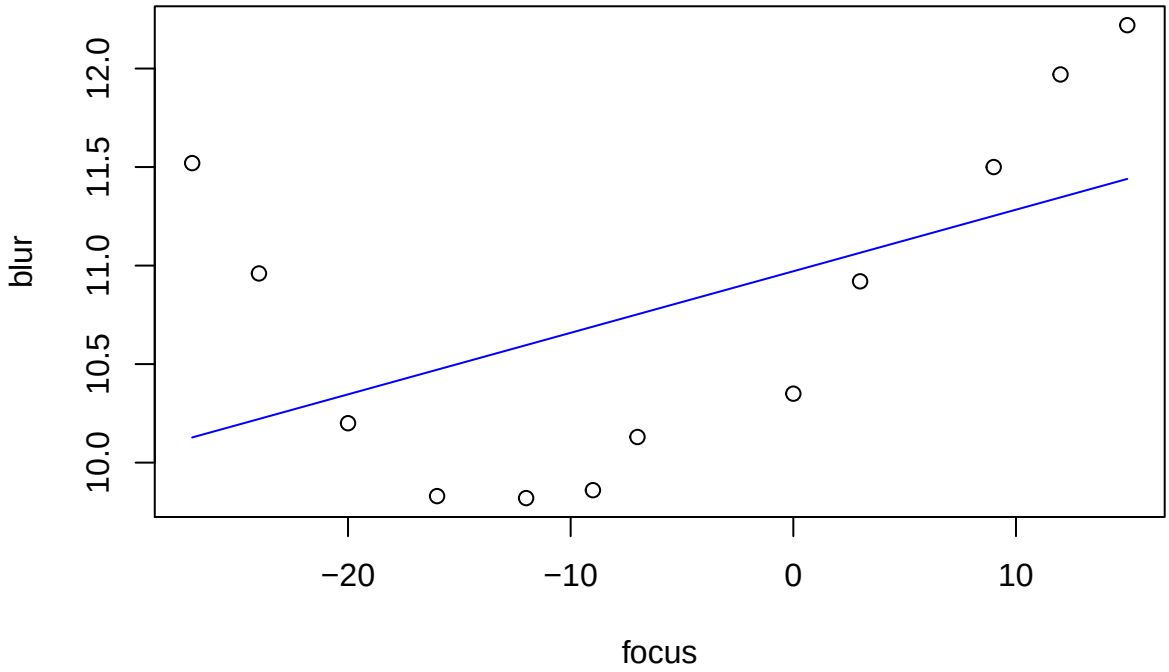
```
focus = c(-27, -24, -20,-16, -12, -9, -7, 0, 3, 9, 12, 15)
blur = c(11.52, 10.96, 10.2, 9.83, 9.82, 9.86, 10.13, 10.35, 10.92, 11.5, 11.97, 12.22)
plot(focus,blur)
```



The data shows a clear dip; a simple linear model is very unlikely to fit the data well.

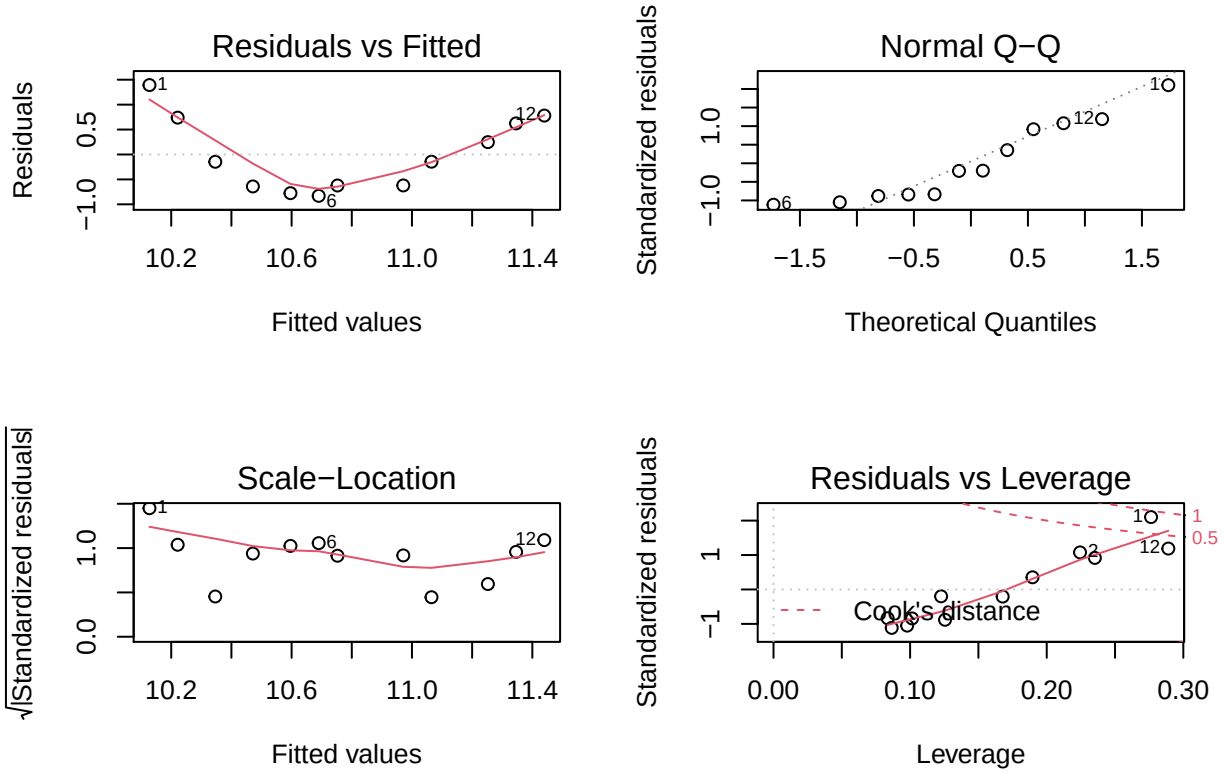
15. Try fitting a simple linear regression model to these data anyway. Overplot the best-fit model line on the data scatter plot.

```
m1 = lm(blur~focus)
plot(focus,blur)
lines(focus,predict(m1),col='blue')
```



16. Confirm that this is a poor fit by investigating the R diagnostic plots. (Use `par(mfrow=c(2,2))` to see all the plots at once, setting `par(mfrow=c(1,1))` when you're done.) Make sure you understand (in qualitative terms) what each panel means.

```
par(mfrow=c(2,2))
plot(m1)
```




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

17. Next, try fitting a quadratic model instead of a simple linear one. What is the t-statistic on the quadratic parameter, and is it significantly different from zero?

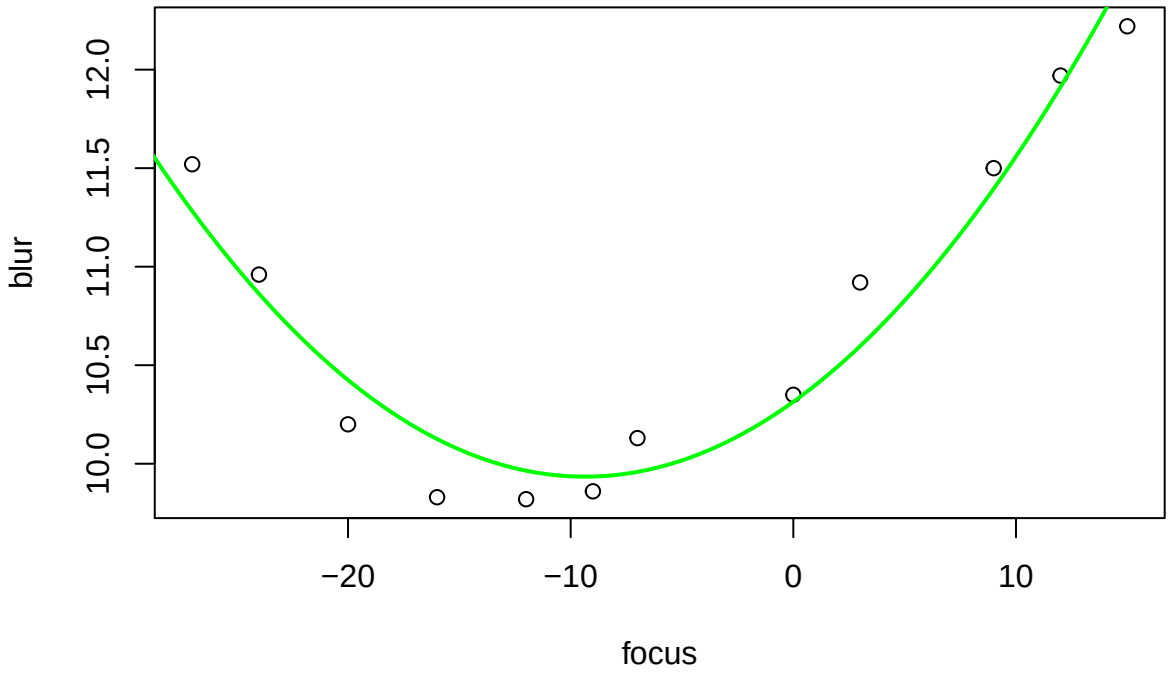
```
m2 = lm(blur~focus+I(focus^2))
summary(m2)
```

```
##
## Call:
## lm(formula = blur ~ focus + I(focus^2))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.29540 -0.16468  0.04672  0.12104  0.32320
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.031e+01  9.568e-02  107.80 2.58e-15 ***
## focus       8.115e-02  6.782e-03   11.97 7.89e-07 ***
## I(focus^2)  4.334e-03  4.147e-04   10.45 2.48e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2265 on 9 degrees of freedom
## Multiple R-squared:  0.9439, Adjusted R-squared:  0.9314
## F-statistic: 75.65 on 2 and 9 DF,  p-value: 2.355e-06
```

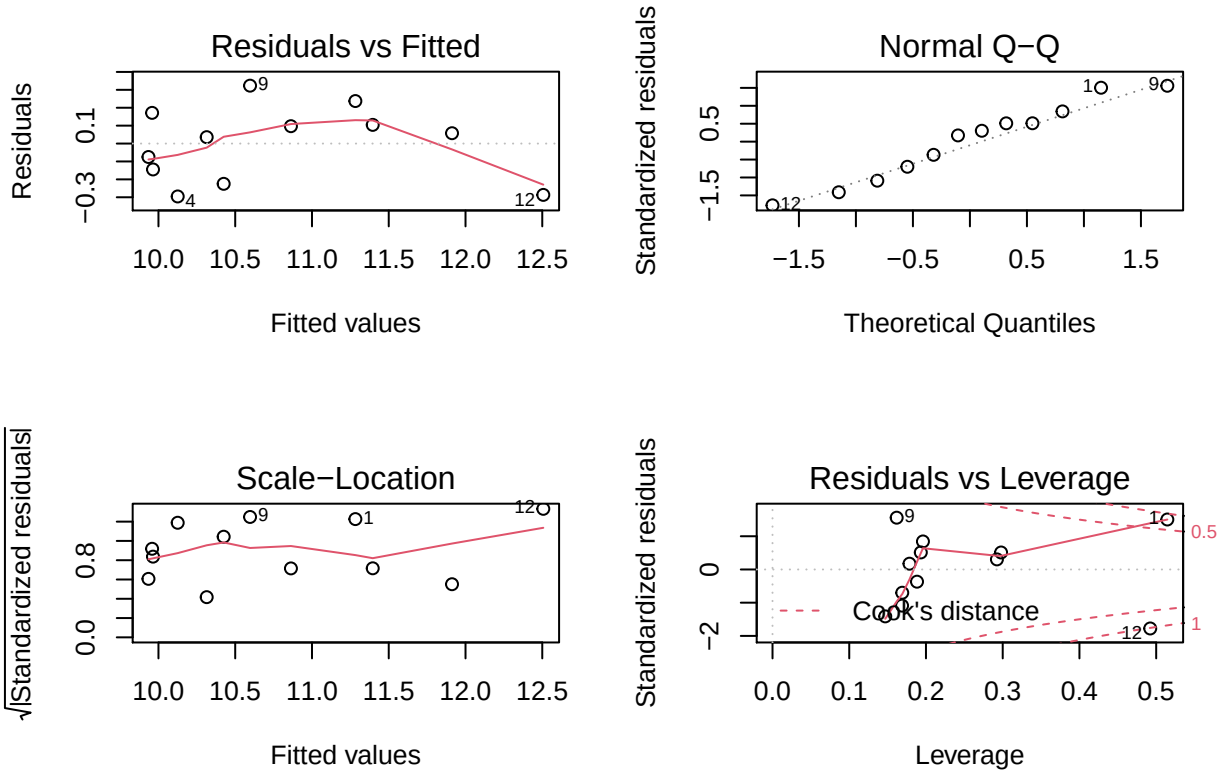
The quadratic parameter has a t-statistic of 10.45 and a significance p-value of 2.48e-6, so it is definitely not consistent with zero.

18. Overplot the curve for the new model against the data (try to make the curve smooth by using a well-sampled plotting variable for the x values!) and check the diagnostic plots. Is there still evidence for a trend?

```
plot(focus,blur)
xplot = seq(-30,20,0.1)
lines(xplot,predict(m2,list(focus=xplot)),col='green',lwd=2)
```




```
par(mfrow=c(2,2))
plot(m2)
```



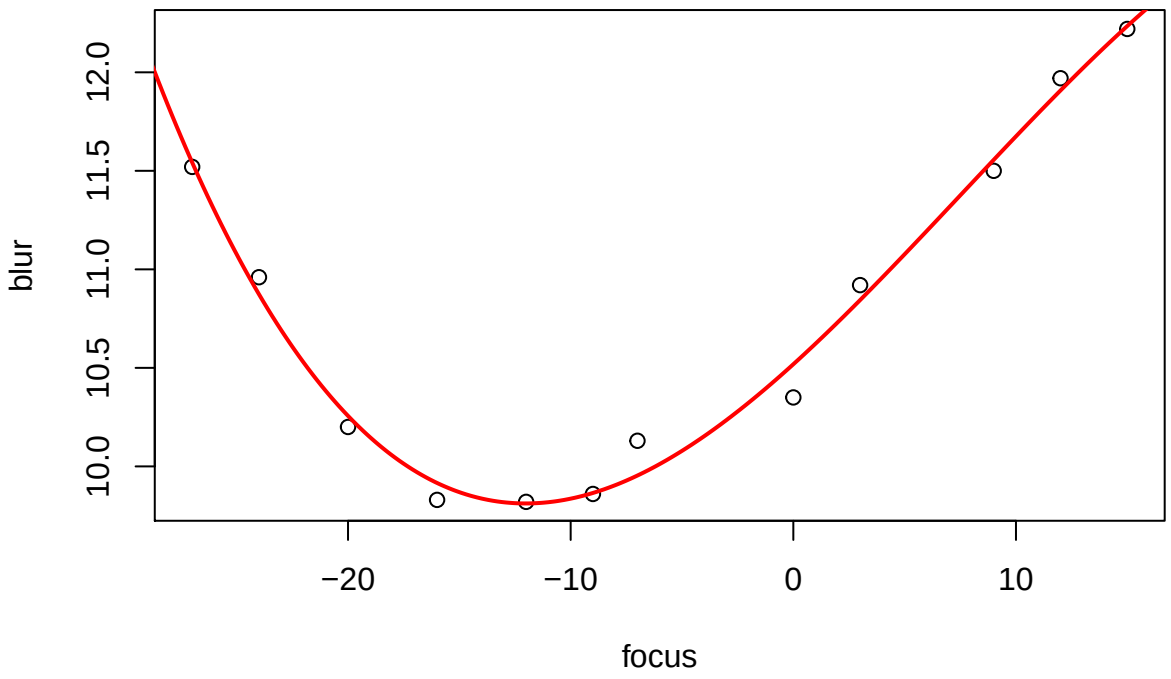

```
par(mfrow=c(1,1))
```

19. Now try fitting a cubic model. (Don't drop the linear and quadratic components: these are still part of a cubic model!) Is the cubic term significantly different from zero? Overplot this curve on the data and check the diagnostic plots.

```
m3 = lm(blur~focus+I(focus^2)+I(focus^3))
summary(m3)
```

```
##
## Call:
## lm(formula = blur ~ focus + I(focus^2) + I(focus^3))
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.168399 -0.055928 -0.009965  0.065581  0.176313
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.052e+01  5.751e-02 182.894 8.94e-16 ***
## focus        1.024e-01  4.905e-03  20.878 2.91e-08 ***
## I(focus^2)   2.371e-03  3.953e-04   5.999 0.000324 ***
## I(focus^3)  -1.047e-04  1.833e-05  -5.709 0.000450 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1066 on 8 degrees of freedom
## Multiple R-squared:  0.9889, Adjusted R-squared:  0.9848
## F-statistic: 238.3 on 3 and 8 DF,  p-value: 3.674e-08
```

```
plot(focus,blur)
ypred = predict(m3,list(focus=xplot))
lines(xplot,ypred,col='red',lwd=2)
```



The cubic term is highly significantly different from zero.

20. Using the cubic model curve, estimate the value of "focus" at which "blur" is minimized and thus the best focus is obtained (numerically is fine, but you can solve the equation if you prefer.) Allow for the possibility that the best focus value is in between the actual measurements.

```
# This is the approximate, numerical method.
xplot[ypred==min(ypred)]
```

```
## [1] -12
```

```
xplot[which.min(ypred)] # a handy shortcut for the above
```

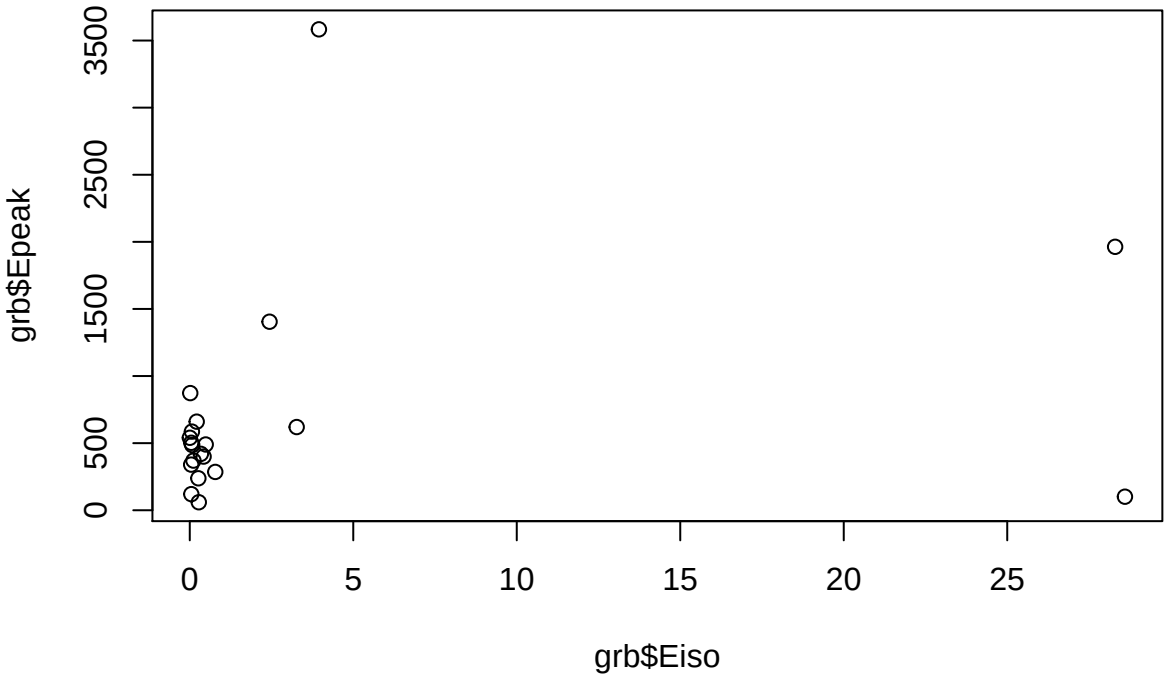
```
## [1] -12
```

Transforms

A recent paper submitted by a group of prominent theoretical astrophysicists (Zou et al., arXiv:/1710.07436) examines the relationship between two parameters observed for gamma-ray bursts, the peak photon energy ("Epeak") and the isotropic-equivalent energy release ("Eiso"). A data table from their paper is given in grbenergy.txt.

21. Load this data table into R (use `read.table` with `header=TRUE` for space-delimited files), and make a simple scatterplot of Eiso (x-axis) versus Epeak (y-axis).

```
grb = read.table('grbenergy.txt',header=TRUE)
plot(grb$Eiso,grb$Epeak)
```



The "Amati relation" purports an empirical relationship between these two quantities described by the following equation:

$$E_{\rm peak} = A \times E_{\rm iso}^{P}$$

This is a nonlinear function, since one parameter (P) is in an exponent and therefore not linear with respect to the dependent variable Epeak.

22. Transform the relation by taking the logarithm of both sides of this equation to produce a linear equation (do this by hand, not in R!). Then, transform your data variables (Epeak and/or Eiso) to match this new equation.

The transformation steps are:
$Epeak = A \times E_{iso}^{P}$
$log(Epeak) = log(A \times E_{iso}^{P})$
$log(Epeak) = log(A) + P \times log(Eiso)$
This is a new linear relation of the form $y = a + bx$, with variables y=log(Epeak) and x=log(Eiso). The intercept is a=log(A) and the slope is b=P.

```
x = log(grb$Eiso)
y = log(grb$Epeak)
```

23. Perform a linear regression with `lm()`in R on the transformed variables and examine the output (use `summary`).

```
grbmodel = lm(y~x)
summary(grbmodel)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

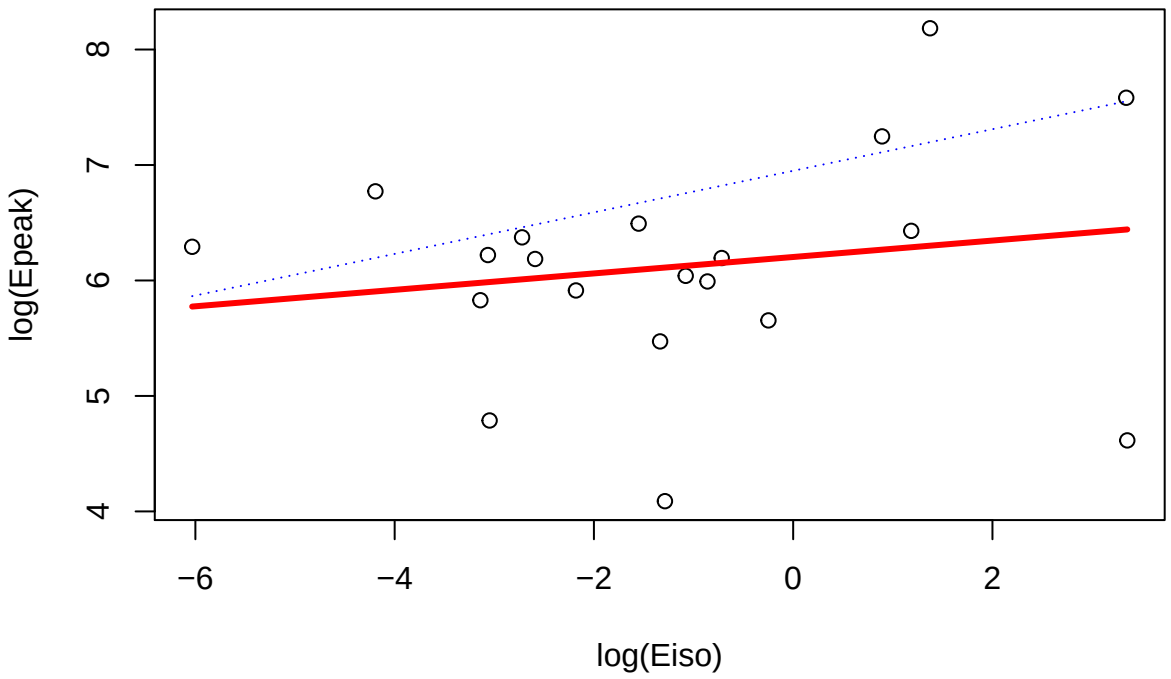
```
## -2.02289 -0.24609  0.09213  0.42883  1.88303
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.20331    0.24337  25.489 1.41e-15 ***
## x            0.07115    0.09326   0.763    0.455
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9676 on 18 degrees of freedom
## Multiple R-squared:  0.03132,    Adjusted R-squared:  -0.0225
## F-statistic: 0.582 on 1 and 18 DF,  p-value: 0.4554
```

24. In the original version of their paper, these authors declared the correlation between the transformed variables to be significant (p-value of p=0.03) and reported coefficients of log(A)=6.95 and P=0.18. Do you agree with their conclusions?

No, linear regression gives different values (6.20 +/- 0.24 for log(A) instead of 6.95, and 0.07 +/- 0.09 for P instead of 0.18) and states that the correlation is not at all significant (p=0.455). The authors did not do their statistics properly (they seem to have ignored sampling error).

25. Plot the transformed variables (as points) and your fitted relation (as a line) on a scatter plot. (Optional: also add the authors' relation.)

```
plot(x,y, xlab='log(Eiso)', ylab='log(Epeak)')
px = sort(x)
lines(px, predict(grbmodel,list(x=px)), col='red', lwd=3) # our relation
lines(px, 6.95+0.18*px, col='blue', lty=3)                # authors' relation
```



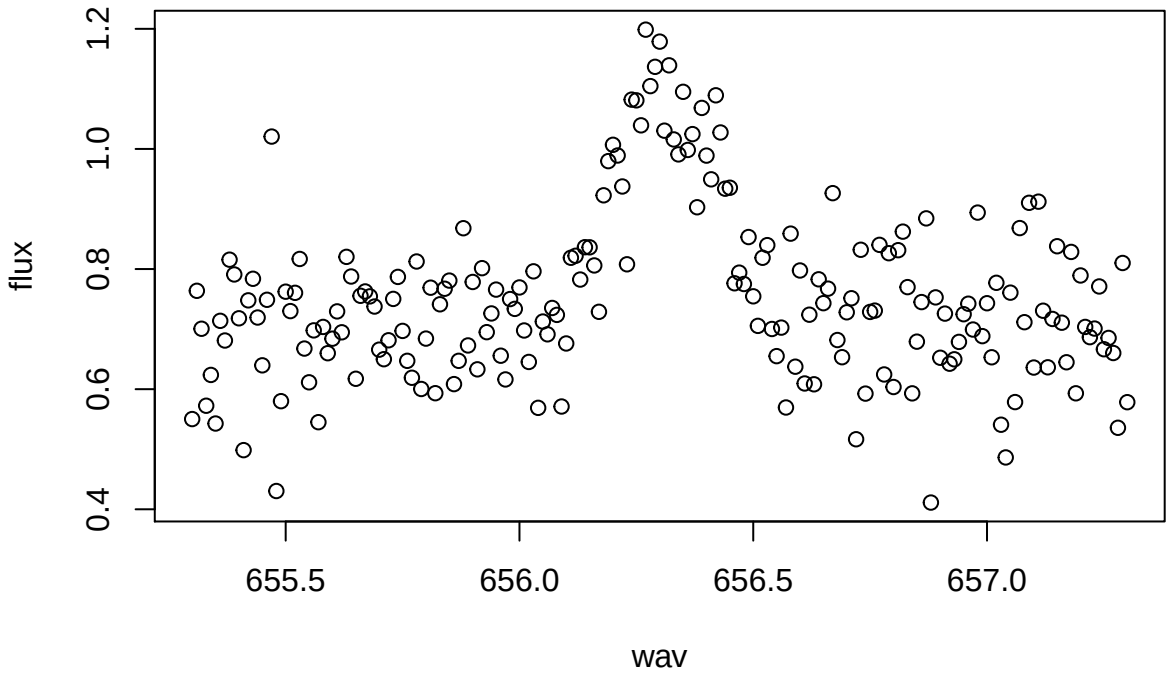
Nonlinear Regression

The data file in specline.csv contains simulated observations of an emission spectrum of an object. The columns are the wavelength in nanometers and the spectral energy flux at that wavelength (in arbitrary

units).

26. Plot the data as a scatter plot (flux is the response variable). Would a linear model (or polynomial model) be appropriate here?

```
spec=read.csv('specline.csv')
plot(spec)
```



27. Fit a non-linear model to the data using `nls()`. Specifically, fit a normal function (of unknown centre C, width W, and height H), plus a constant component A:

$$y = A + H \times \exp(\frac{-(x-C)^2}{2W^2})$$

You'll have to specify an initial guess for the model, and it's essential that you guess this reasonably accurately: narrowly peaked functions such as this one can easily get "lost" in parameter space!

```
linemodel = nls(flux~A+H*exp(-(wav-C)^2/(2*W^2)),start=list(A=0.7,H=1,C=656.3,W=0.2), data=spec)
```

28. What is the significance of the detection of this spectral line? (A "detection" in this context means that the height of the line H is significantly greater than zero.)

```
summary(linemodel)
```

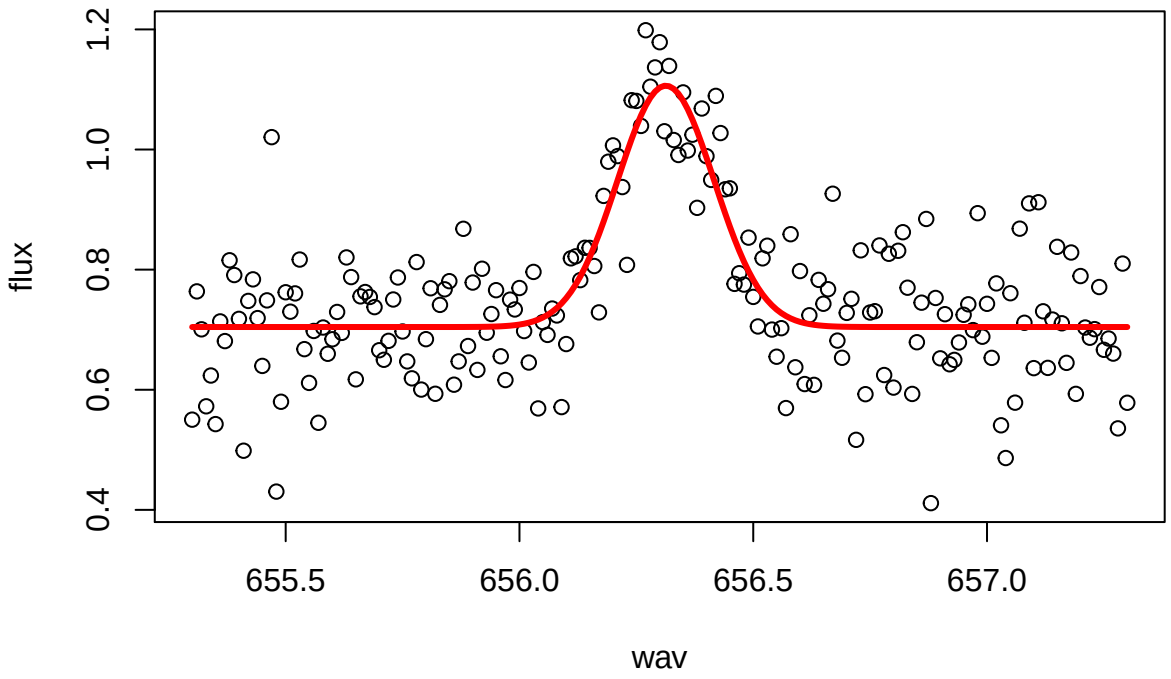
```
##
## Formula: flux ~ A + H * exp(-(wav - C)^2/(2 * W^2))
##
## Parameters:
##    Estimate Std. Error  t value Pr(>|t|)
## A 7.044e-01  7.767e-03    90.69   <2e-16 ***
## H 4.019e-01  2.740e-02    14.66   <2e-16 ***
## C 6.563e+02  7.977e-03 82271.35   <2e-16 ***
## W 1.034e-01  8.463e-03    12.22   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.09386 on 197 degrees of freedom
##
## Number of iterations to convergence: 9
## Achieved convergence tolerance: 3.362e-06
```

Despite the noisy data the significance of H is very high: the p-value is less than 2e-16.

29. Plot the model curve on top of the data.

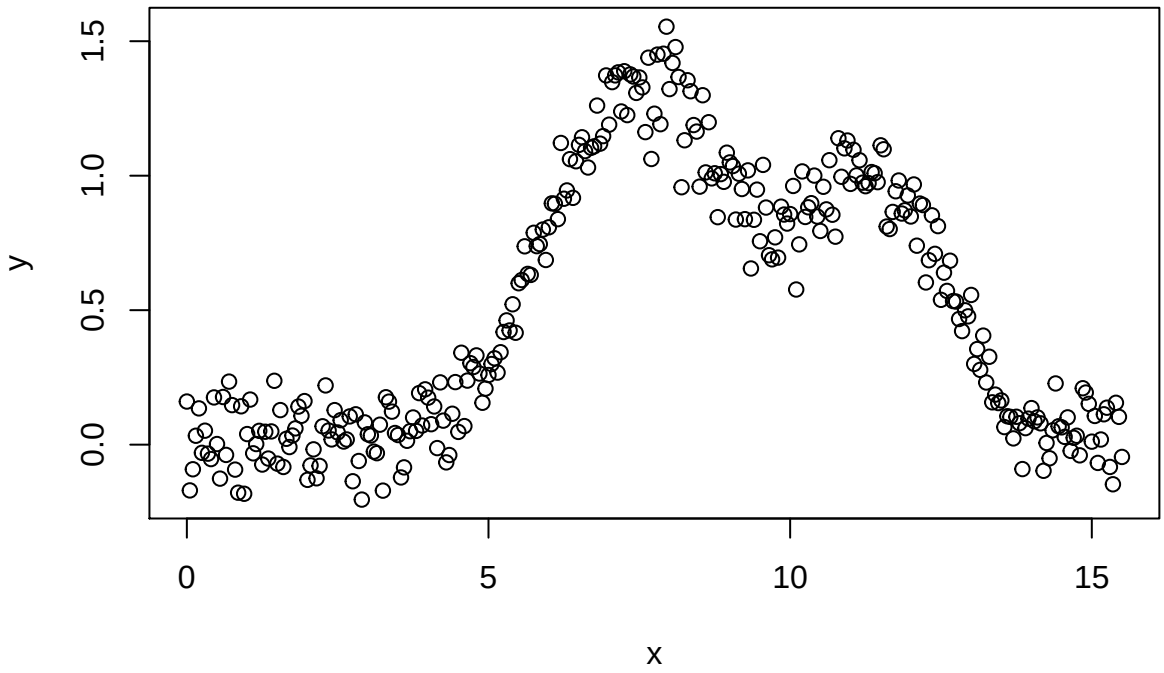
```
plot(spec)
lines(spec$wav,predict(linemodel),col='red',lwd=3)
```



Nonparametric Regression and Smoothing

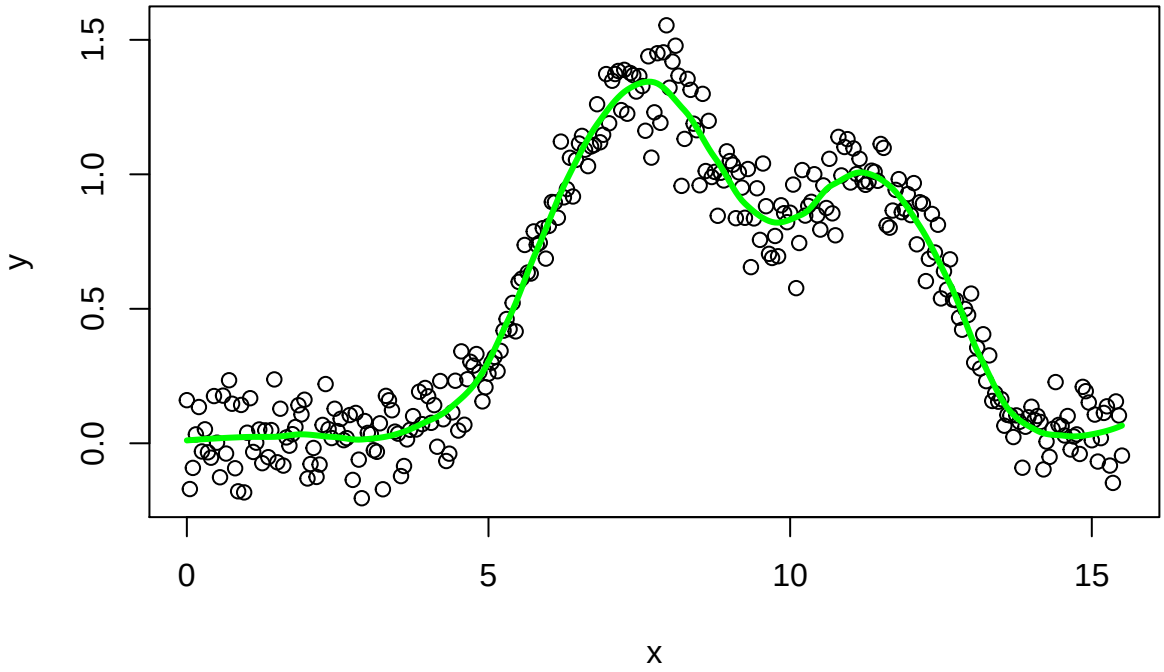
30. Load in the data in bumpy.csv. Make a scatter plot of x versus y from this data. Is there an obvious functional model that fits this curve?

```
bumpy = read.csv('bumpy.csv')
plot(bumpy)
```



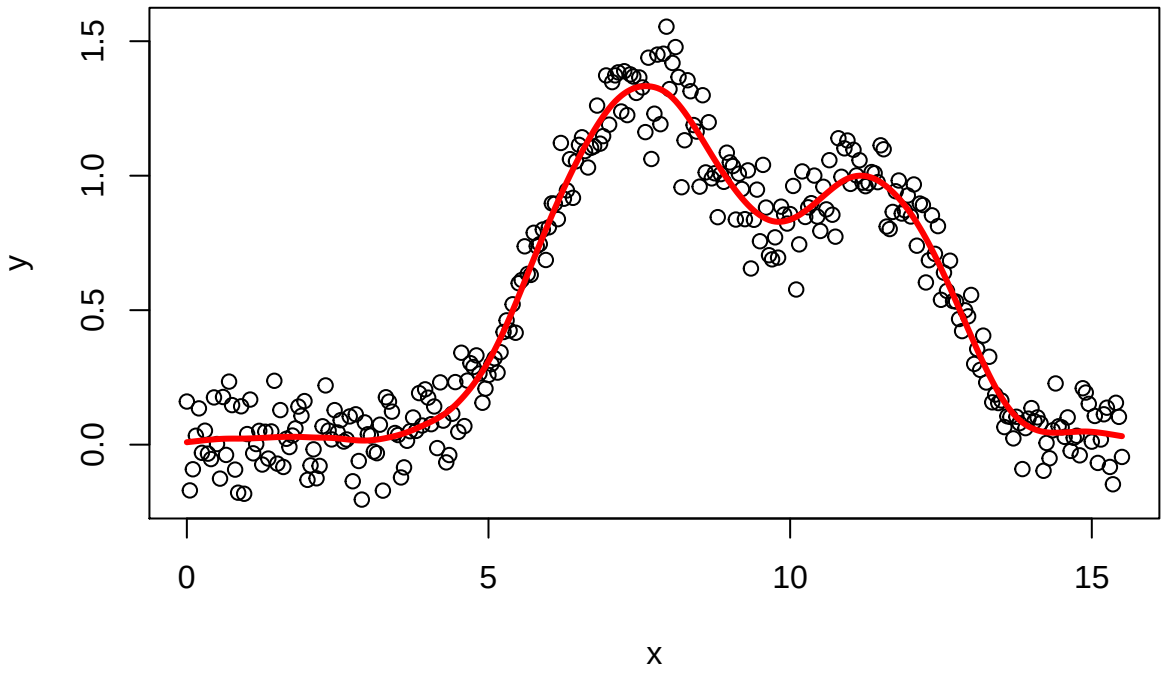
31. Fit a local regression (loess) model, then replot the data and add the loess prediction curve. If the fit is not good, try reducing `span`.

```
lbump = loess(y~x,span=0.2, data=bumpy)
plot(bumpy)
lines(bumpy$x,predict(lbump),col='green',lwd=3)
```



32. Fit a smoothed spline model, then replot the data and add the new prediction curve.

```
sbump = smooth.spline(bumpy$y~bumpy$x) # data argument doesn't work here
plot(bumpy)
lines(bumpy$x,predict(sbump)$y,col='red',lwd=3)
```



33. Fit a generalized additive model using `gam()`, then replot the data and add the new prediction curve.

```
library(mgcv)
```

```
## Loading required package: nlme
```

```
## This is mgcv 1.8-33. For overview type 'help("mgcv-package")'.
```

```
gbump = gam(bumpy$y~s(bumpy$x))
plot(bumpy)
lines(bumpy$x,predict(gbump),col='blue',lwd=3)
```