ST2304 Exercises Week 7: Multiple Regression

Bob O'Hara

19 February 2018

Problem 1: Life Expectancy (again)

First, read in the data:

```
rawdata <- read.csv("../Data/LifeExpectancy.csv")
NoSA <- rawdata[rawdata$Country!="South Africa",]
```

For each model, find the R^2, and look at and describe how the R^2 changes with the order of the polynomial (hint: plot them).

First, fit the univariate models.

```
LElog <- lm(Life.exectancy ~ log(Health.Spending.per.capita), data=NoSA)

LE1 <- lm(Life.exectancy ~ Health.Spending.per.capita, data=NoSA)
LE2 <- lm(Life.exectancy ~ Health.Spending.per.capita + I(Health.Spending.per.capita^2),
          data=NoSA)
LE3 <- lm(Life.exectancy ~ Health.Spending.per.capita + I(Health.Spending.per.capita^2) +
          I(Health.Spending.per.capita^3), data=NoSA)
LE4 <- lm(Life.exectancy ~ Health.Spending.per.capita + I(Health.Spending.per.capita^2)+
          I(Health.Spending.per.capita^3) + I(Health.Spending.per.capita^4),
          data=NoSA)
LE5 <- lm(Life.exectancy ~ Health.Spending.per.capita + I(Health.Spending.per.capita^2)+
          I(Health.Spending.per.capita^3) + I(Health.Spending.per.capita^4) +
            I(Health.Spending.per.capita^5), data=NoSA)
# Oh yes, I should ue update()
LE6 <- update(LE5, .~. + I(Health.Spending.per.capita^6))
LE7 <- update(LE6, .~. + I(Health.Spending.per.capita^7))
LE8 <- update(LE7, .~. + I(Health.Spending.per.capita^8))
LE9 <- update(LE8, .~. + I(Health.Spending.per.capita^9))
LE10 <- update(LE9, .~. + I(Health.Spending.per.capita^10))
```

Looking at the R^2 values (Fig. 1), we can see that they increase with the order of the polynomial, but the rate of increase gets less.

```
Order <- 1:10
R2 <- c(summary(LE1)$r.square, summary(LE2)$r.square, summary(LE3)$r.square,
        summary(LE4)$r.square, summary(LE5)$r.square, summary(LE6)$r.square,
        summary(LE7)$r.square, summary(LE8)$r.square, summary(LE9)$r.square,
        summary(LE10)$r.square)
plot(Order, R2, ylab=expression(R^2))
abline(h=summary(LElog)$r.square, lty=2)
```

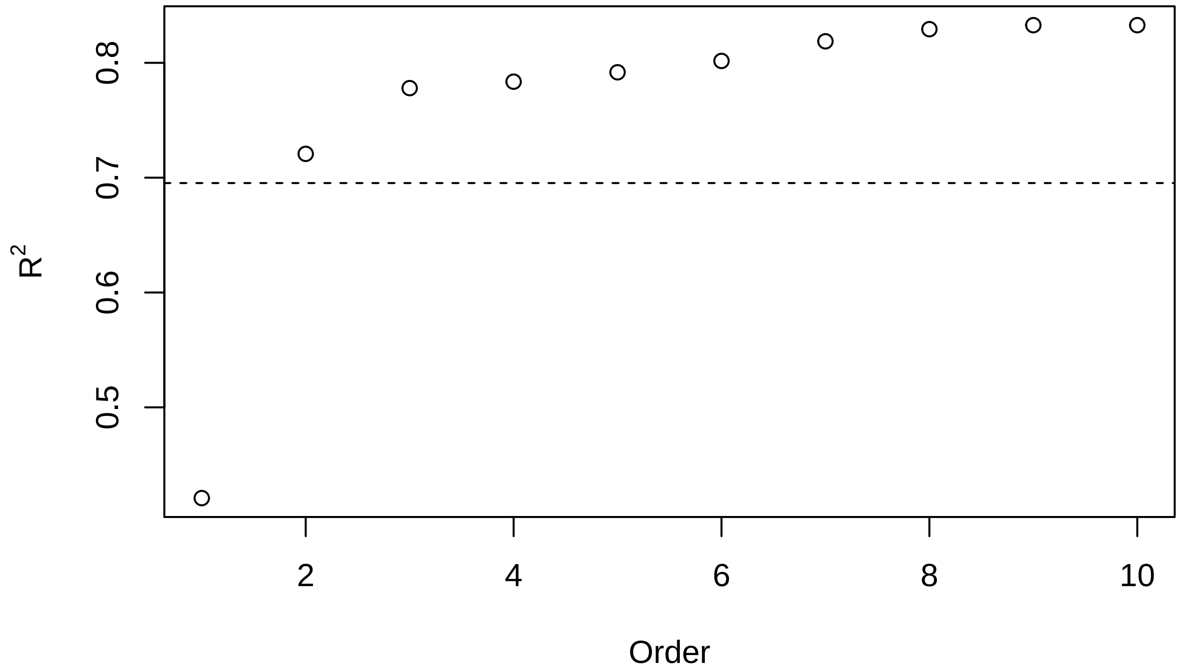


Figure 1: Plot of R^2 for polynomial models of life expectancy data. Dashed Line: R^2 for log-transformed healthcare spending

There are a few ways to decide which order of polynomial is best, but without trying them, what order do you prefer, and why? What factors are important for making this decision?

The quadratic is obviously much better than the linear model: the R^2 goes up from 42% to 72%. But after that the improvements become less, and after the cubic model (i.e. the model with x^3), the increase in R^2 is small. Models with more parameters are more complex - they have more parameters - and in general it is better to stick to simpler models (this principle is known as is "Occam's Razor"), so although the 10th order polynomial fits best, it is not worth using all those extra parameters. Where exactly to stop fitting isn't obvious just from the graph, and in practice you should consider the purpose of the model: if it is just to describe the overall patterns in the data, then a simple model might be better, but for prediction, a more complicated model might be more precise. Here I would use either the quadratic or cubic model: let's go with the cubic model. But I could also see justifications for other order (e.g. order 8), depending on how much you want to emphasise fit or complexity. Without more formal tools to look at this, it is complex.

My conclusion from looking at the data last week was that log-transforming the x-axis (health spending) was the best alternative. So fit that model and compare how well that model to the polynomial models.

I plotted that fit in Fig. 1 too: we can see that even the quadratic model does quite a bit better, with a higher R^2.

Check how well the model fits - are there any outliers, any influential points, any bigger problems (e.g. heteroscedasticity)?

```
NoSA$fitted.p3 <- fitted(LE3)
NoSA$resid.p3 <- resid(LE3)
NoSA$CooksD.p3 <- cooks.distance(LE3)
```

```
par(mfrow=c(1,3), mar=c(4,2,3,1))
plot(NoSA$fitted.p3, NoSA$resid.p3, type="n", main="Residuals vs Fitted")
text(NoSA$fitted.p3, NoSA$resid.p3, NoSA$Country)
qqnorm(NoSA$resid.p3); qqline(NoSA$resid.p3, main="Normal Probability Plot")
plot(NoSA$fitted.p3, NoSA$CooksD.p3, type="n", main="Cook' D")
text(NoSA$fitted.p3, NoSA$CooksD.p3, NoSA$Country)
```

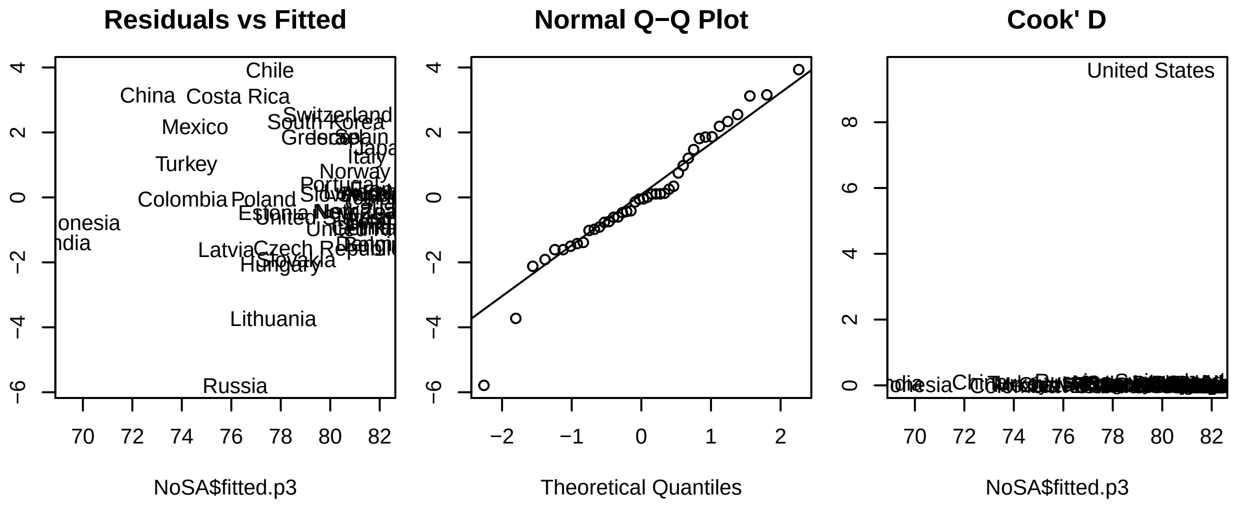


Figure 2: Model fit checks for cubic model

Overall the residuals look OK (Fig. 2), but we can see that Russia is an outlier. The other big influencer is the USA, which has has a huuuge effect on the model fit: Cook's D is massive. We can look at what the US is doing by plotting the fitted model on top of the data.

Plot the fitted model with the data

```
pred.raw <- data.frame(Health.Spending.per.capita=
                         seq(min(NoSA$Health.Spending.per.capita),
                             max(NoSA$Health.Spending.per.capita), length=50))
pred <- predict.lm(LE3, newdata=pred.raw, interval = "pred")
pred.raw <- cbind(pred.raw, pred)
LE3noUS <- update(LE3, .~., data=NoSA[NoSA$Country!="United States",])
pred.noUS <- predict.lm(LE3noUS, newdata=pred.raw, interval = "pred")

par(mar=c(4.1,4.1,1,1))
plot(NoSA$Health.Spending.per.capita, NoSA$Life.exectancy, type="n",
     xlab="Health Spending per capita", ylab="Life expectancy (years)")
lines(pred.raw$Health.Spending.per.capita, pred.noUS[,"fit"], lwd=1.5, col=2)

lines(pred.raw$Health.Spending.per.capita, pred.raw$fit, lwd=1.5, col="grey50")
lines(pred.raw$Health.Spending.per.capita, pred.raw$lwr, lty=3, col="grey30")
lines(pred.raw$Health.Spending.per.capita, pred.raw$upr, lty=3, col="grey30")
text(NoSA$Health.Spending.per.capita, NoSA$Life.exectancy, NoSA$Country)
```

We can see that the general pattern is that there is a large increase in life expectancy if countries with low healthcare spending increase their spending slightly, but for richer countries the effect is less. The model also suggestst that very rich countries havea decrease in life expectancy as spending increases, which is odd. But

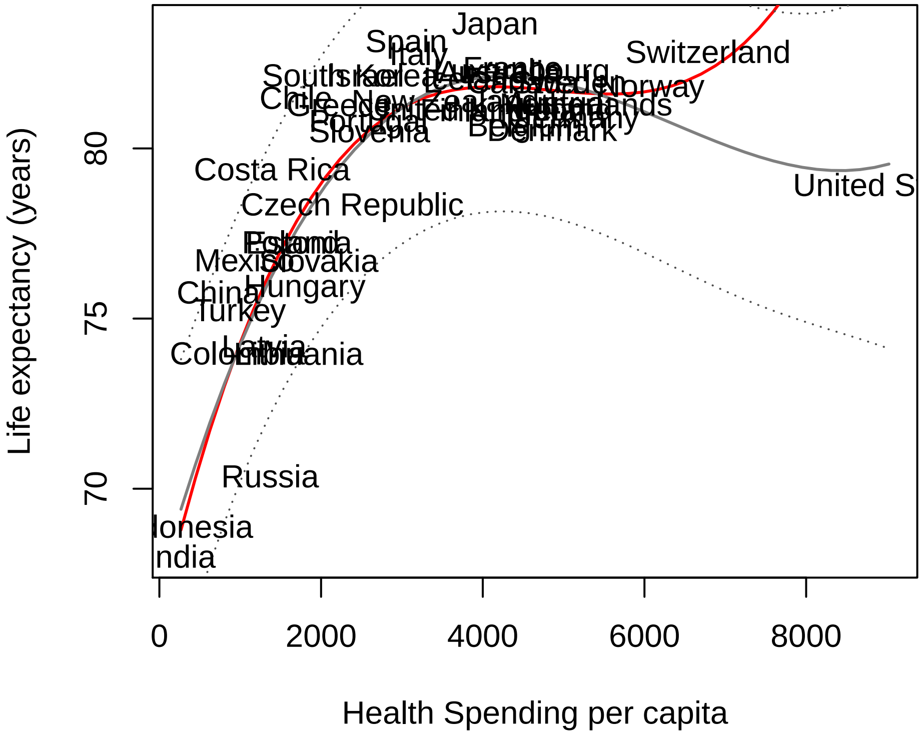


Figure 3: Life expectancy data, with cubic model plotted

the USA has large spending, and we already know that it is influential. So what happens when we remove the US?

The red line in Fig. 3 is the model without the USA: we can see that their effect is to pull down the fitted line for countries that spend a lot on healthcare, although for most countries the effect is minimal. Also note that without the US, the effect of healthcare really shoots up (according to this model, the predicted life expectancy of an American would be 91 years. This is probably unrealistic, the reason this happens is because the polynomials have to go to plus or minus infinity, so they will explode when extrapolated beyond the data. This is not usually a problem, as long as you don't extrapolate beyond the data.

Problem 2

Simulate x_1 and x_2 from a standard normal distribution with no correlation. Then simulate the model with $\beta_1 = 1$ and $\beta_2 = 1$. Plot y against each x, and regress y against each x separately. Explain the results. Then regress y against both x's, and again explain the results.

```
library(MASS)
N <- 50; alpha <- 0; sigma <- 1; muX <- c(0,0) # same throughout
beta1 <- 1
beta2 <- 1

Corr <- 0 # correlation
sigmaX <- matrix(c(1,Corr,Corr,1), nrow=2) # covariance matrix
x1 <- mvrnorm(N, muX, Sigma=sigmaX) # 2 columns: x[,1] & x[,2]

mu1 <- alpha + beta1*x1[,1] + beta2*x1[,2]
```

```
y1 <- rnorm(N, mu1, sigma)

par(mfrow=c(1,2), mar=c(4.1,4,1,1))
plot(x1[,1], y1)
plot(x1[,2], y1)
```

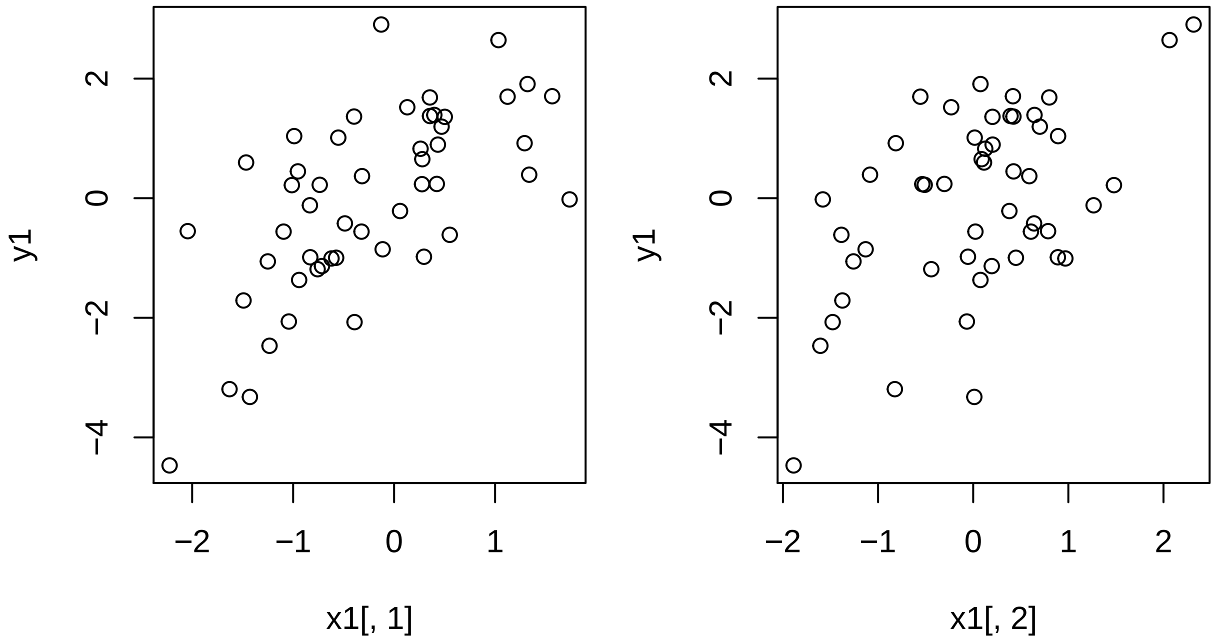


Figure 4: Plots of straightforward model data

This is straightforward: from the plot there are positive correlations between each x and y. When we look at the results of fitting the models (Tables 1-3), we see that the estimates are all fairly close to the true values (1). With the model with both variables in it (i.e. the true model), we see that the standard errors are smaller.

```
mod1.1 <- lm(y1 ~ x1[,1])
mod1.2 <- lm(y1 ~ x1[,2])
mod1.12 <- lm(y1 ~ x1)

knitr::kable(summary(mod1.1)$coefficients, digits = 2,
             caption = "Parameter estimates for straightforward model with $x_1$")
```

Table 1: Parameter estimates for straightforward model with x_1

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.20	0.17	1.18	0.24
x1[, 1]	1.07	0.18	6.07	0.00

```
knitr::kable(summary(mod1.2)$coefficients, digits = 2,
             caption = "Parameter estimates for straightforward model with $x_2$")
```

Table 2: Parameter estimates for straightforward model with x_2

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-0.08	0.18	-0.45	0.66
x1[, 2]	0.90	0.20	4.42	0.00

```r
knitr::kable(summary(mod1.12)$coefficients, digits = 2,
             caption = "Parameter estimates for straightforward model with $x_1$ and $x_2$")
```

Table 3: Parameter estimates for straightforward model with x_1 and x_2

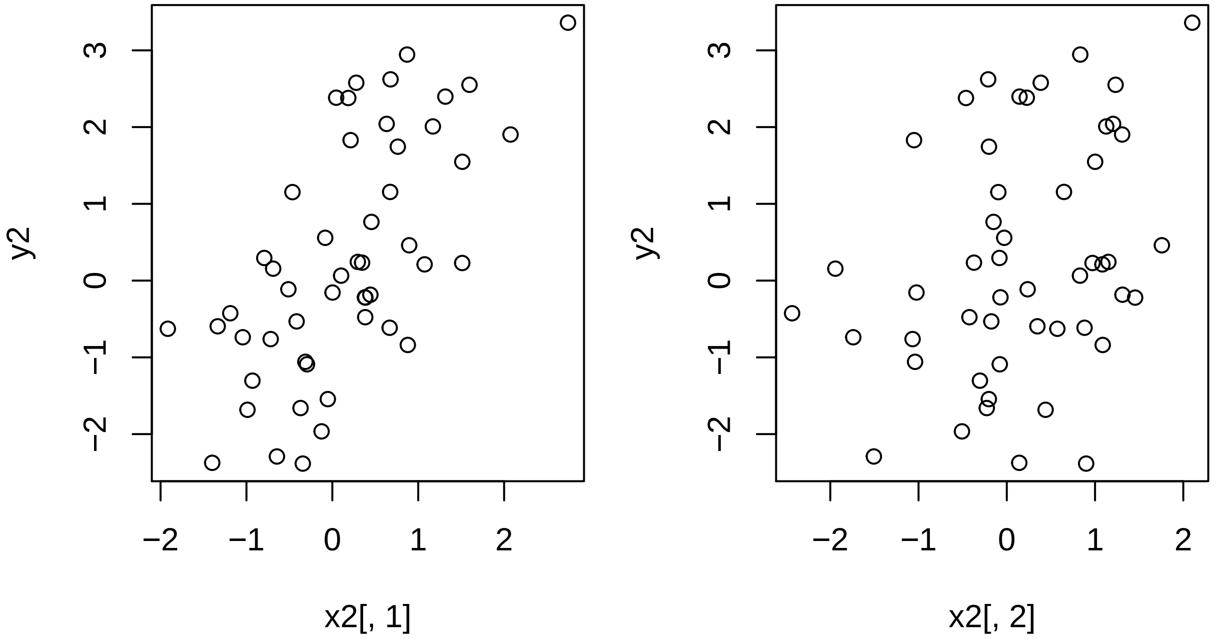
	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.18	0.12	1.43	0.16
x11	1.05	0.13	8.17	0.00
x12	0.87	0.13	6.60	0.00

Simulate x_1 and x_2 from a standard normal distribution with correlation 0.5, with $\beta_1 = 1$ and $\beta_2 = 0$.

```r
Corr <- 0.7 # correlation
sigmaX <- matrix(c(1,Corr,Corr,1), nrow=2) # covariance matrix
x2 <- mvrnorm(N, muX, Sigma=sigmaX) # 2 columns: x[,1] & x[,2]
beta1 <- 1
beta2 <- 0

mu2 <- alpha + beta1*x2[,1] + beta2*x2[,2]
y2 <- rnorm(N, mu2, sigma)

par(mfrow=c(1,2), mar=c(4.1,4,1,1))
plot(x2[,1], y2)
plot(x2[,2], y2)
```



Just from plotting the data, we can see a correlation between y and x_2, even though there is no actual effect of x_2 on y. We can see this in the models (Tables 4-6): we estimate a positive effect of x_2 in the model with just x_2. In the model with x_1 and x_2 the estimated effect of x_2 is much lower. This is because the effect x_1 has on y and the correlation between x_1 and x_2 means that there is still a correlation between y and x_2, so that indirect ffect is what we estimate. Also note the high standard error: the high correlation between X_1 and x_2 makes the parameter estimates uncertain: we cannot be sure if we are estimating an effect of x_1 or x_2.

```
mod2.1 <- lm(y2 ~ x2[,1])
mod2.2 <- lm(y2 ~ x2[,2])
mod2.12 <- lm(y2 ~ x2)

knitr::kable(summary(mod2.1)$coefficients, digits = 2,
             caption = "Parameter estimates for straightforward model with $x_1$")
```

Table 4: Parameter estimates for straightforward model with x_1

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.11	0.17	0.65	0.52
x2[, 1]	1.06	0.18	5.94	0.00

```
knitr::kable(summary(mod2.2)$coefficients, digits = 2,
             caption = "Parameter estimates for straightforward model with $x_2$")
```

Table 5: Parameter estimates for straightforward model with x_2

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.2	0.21	0.95	0.35
x2[, 2]	0.5	0.21	2.34	0.02

```
knitr::kable(summary(mod2.12)$coefficients, digits = 2,
             caption = "Parameter estimates for straightforward model with $x_1$ and $x_2$")
```

Table 6: Parameter estimates for straightforward model with x_1 and x_2

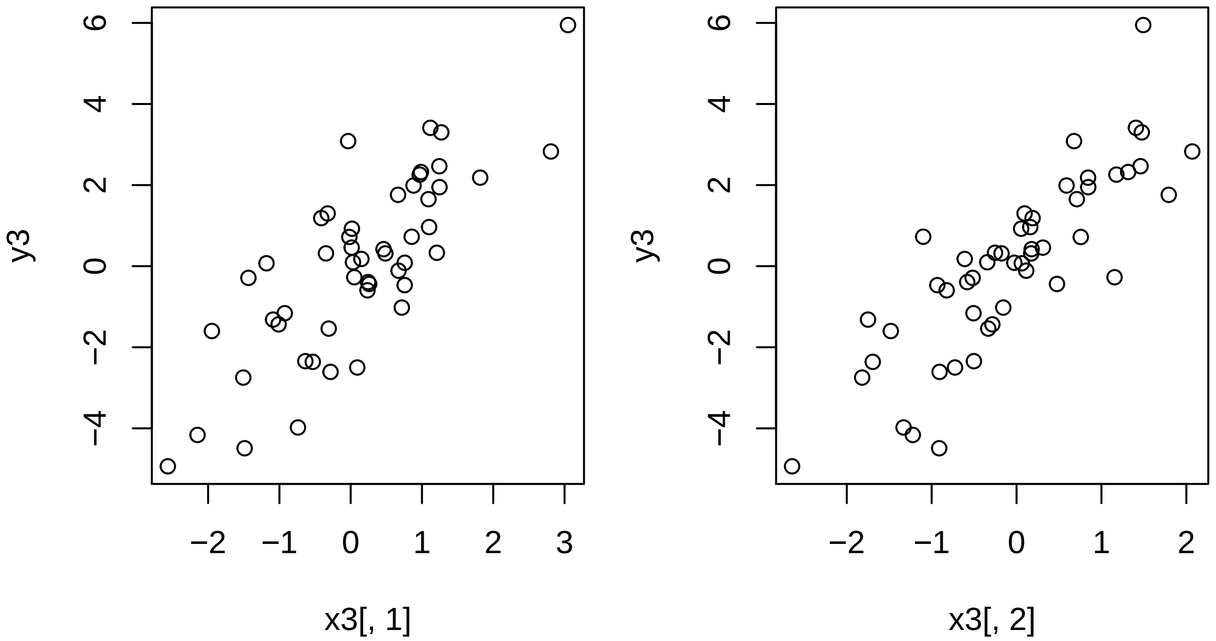
	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.12	0.17	0.71	0.48
x21	1.17	0.22	5.21	0.00
x22	-0.17	0.22	-0.79	0.43

Simulate x_1 and x_2 from a standard normal distribution with correlation 0.7, with $\beta_1 = 1$ and $\beta_2 = 1$.

```r
Corr <- 0.7 # correlation
sigmaX <- matrix(c(1,Corr,Corr,1), nrow=2) # covariance matrix
x3 <- mvrnorm(N, muX, Sigma=sigmaX) # 2 columns: x[,1] & x[,2]
beta1 <- 1
beta2 <- 1

mu3 <- alpha + beta1*x3[,1] + beta2*x3[,2]
y3 <- rnorm(N, mu3, sigma)

par(mfrow=c(1,2), mar=c(4.1,4,1,1))
plot(x3[,1], y3)
plot(x3[,2], y3)
```



Just from plotting the data, we can see a strong correlation between y and both x's. We can see this in the models too (Tables 7-9), with positive effects of both x_1 and x_2 individually. This is not surprising, but note that the parameters are both above the true values: the correlations make the effects seem stronger. In the model with both variables, these estimates are smaller.

```r
mod3.1 <- lm(y3 ~ x3[,1])
mod3.2 <- lm(y3 ~ x3[,2])
mod3.12 <- lm(y3 ~ x3)
```

```
knitr::kable(summary(mod3.1)$coefficients, digits = 2,
             caption = "Parameter estimates for correlated model with $x_1$")
```

Table 7: Parameter estimates for correlated model with x_1

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-0.15	0.19	-0.77	0.45
x3[, 1]	1.55	0.17	9.25	0.00

```
knitr::kable(summary(mod3.2)$coefficients, digits = 2,
             caption = "Parameter estimates for correlated model with $x_2$")
```

Table 8: Parameter estimates for correlated model with x_2

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.12	0.17	0.72	0.48
x3[, 2]	1.79	0.17	10.73	0.00

```
knitr::kable(summary(mod3.12)$coefficients, digits = 2,
             caption = "Parameter estimates for correlated model with $x_1$ and $x_2$")
```

Table 9: Parameter estimates for correlated model with x_1 and x_2

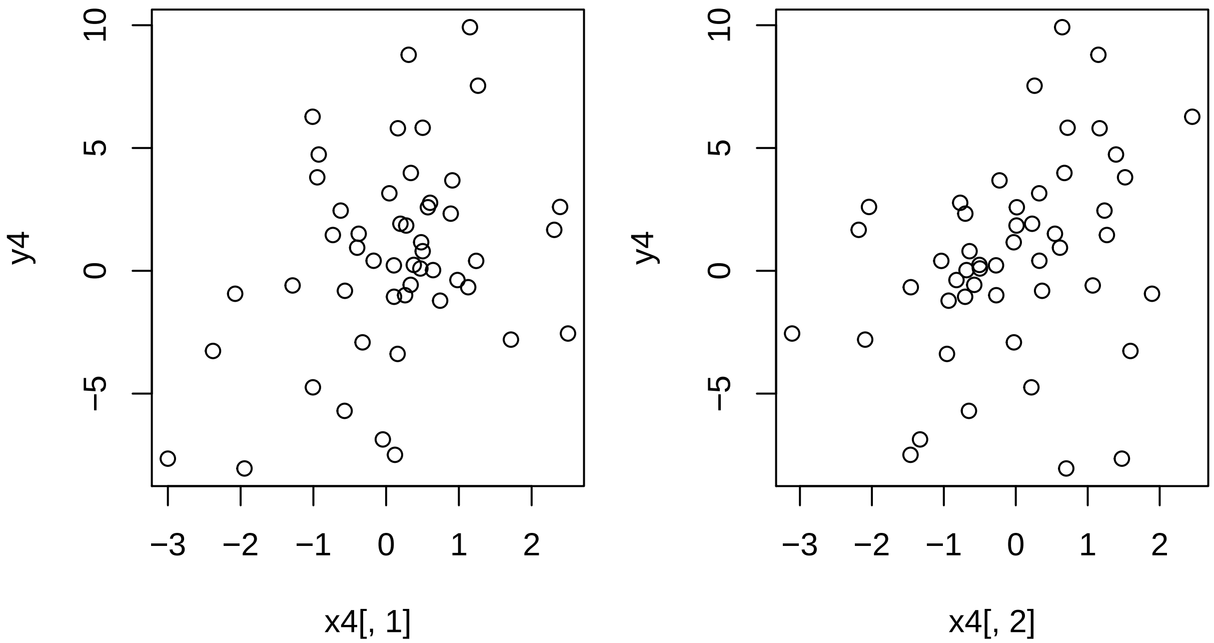
	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.00	0.16	0.02	0.99
x31	0.75	0.20	3.68	0.00
x32	1.17	0.22	5.19	0.00

Simulate x_1 and x_2 from a standard normal distribution with correlation -0.8. Then simulate the model (above) with $\beta_1 = 5$ and $\beta_2 = 5$.

```
Corr <- -0.8 # correlation
sigmaX <- matrix(c(1,Corr,Corr,1), nrow=2) # covariance matrix
x4 <- mvrnorm(N, muX, Sigma=sigmaX) # 2 columns: x[,1] & x[,2]
beta1 <- 5
beta2 <- 5

mu4 <- alpha + beta1*x4[,1] + beta2*x4[,2]
y4 <- rnorm(N, mu4, sigma)

par(mfrow=c(1,2), mar=c(4.1,4,1,1))
plot(x4[,1], y4)
plot(x4[,2], y4)
```



Plotting the data, we don't see any strong correlation between y and the x's, even though we know it is there (different simulations give different answers, so you might see an effect in one variable). We can see this in the models too (Tables 10-12): the estimates of x_1 and x_2 on their own are much smaller than the true estimates. But in the model with both variables, these estimates are larger and close to the true values. What is going on is that the megative correlation between X_1 and x_2 masks the positive effects of each on y: they counteract each other.

```
mod4.1 <- lm(y4 ~ x4[,1])
mod4.2 <- lm(y4 ~ x4[,2])
mod4.12 <- lm(y4 ~ x4)

knitr::kable(summary(mod4.1)$coefficients, digits = 2,
             caption = "Parameter estimates for correlated model with $x_1$")
```

Table 10: Parameter estimates for correlated model with x_1

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.40	0.54	0.73	0.47
x4[, 1]	1.19	0.49	2.44	0.02

```
knitr::kable(summary(mod4.2)$coefficients, digits = 2,
             caption = "Parameter estimates for correlated model with $x_2$")
```

Table 11: Parameter estimates for correlated model with x_2

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.57	0.54	1.06	0.30
x4[, 2]	1.12	0.47	2.37	0.02

```
knitr::kable(summary(mod4.12)$coefficients, digits = 2,
             caption = "Parameter estimates for correlated model with $x_1$ and $x_2$")
```

Table 12: Parameter estimates for correlated model with x_1 and x_2

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.18	0.15	1.23	0.22
x41	5.10	0.21	24.24	0.00
x42	4.94	0.20	24.15	0.00

The overall summary of this is that models can be awkward. This is a particular problem when the data do not come from an experiment: then there may be variables that have an effect, but have not been measured. It is often noted in statistics that "correlation does not imply causation" (https://xkcd.com/552/). These exercises show why.

Problem 3

```r
BirdBrains <- read.csv('../Data/BirdBrains.csv') # beware the file path: this is for my computer!

Names <- c("Order", "Family", "Species.name.")
Variables <- c("Maximum.lifespan", "Age.at.first.reprodction", "Incubation.length", "Clutch.size",
               "Mean.latitude", "logBodyMass", "logBrainMass")
BirdClutch <- BirdBrains[,c(Names,Variables)] # select the variables we want
BirdClutch[,Variables] <- scale(BirdClutch[,Variables]) # standardise the variables

# Plot the variables (using sapply() is more advanced R programming)
par(mfrow=c(2,3), mar=c(4.1,1.1,1,1), oma=c(0,2.5,0,0))
sapply(Variables[Variables!="Clutch.size"], function(var, dat) {
  plot(dat[,var], dat$Clutch.size, xlab=var, ylab="")
}, dat=BirdBrains)
mtext("Clutch Size", 2, outer=TRUE, line=1)
```

Why is scaling like this a good idea?

The advantage of scaling variables is that it makes them more comparable. It is difficult to see how to compare log body mass and lifespan, for example: their units ar different, so how does a change in log(1g) compare to 1 year? But if we standardise the variables, then we can ask how a change across one standard deviation in one variable compares to a change in one standard deviation in another.

Be aware that this isn't perfect: it depends on the data. So if we get another dataset, then the variances will be different, and the scaling will be different too.

Fit univariate models, i.e. explain clutch size by each of the covariates individually.

We can plot the effects (or put them in a table!). The R^2 are all low, below 20% (this is typical). Age at first reproduction and body and brain mass all have negative effects: species that are large and wait longer before starting to breed tend to have smaller clutches. Age at first reproduction has the largest effect size, and the largest R^2.

```r
Mods <- sapply(Variables[Variables!="Clutch.size"], function(var, dat) {
  form <- formula(paste0("Clutch.size~", var))
  lm(form, data=dat)
}, dat=BirdBrains, simplify=FALSE)
```

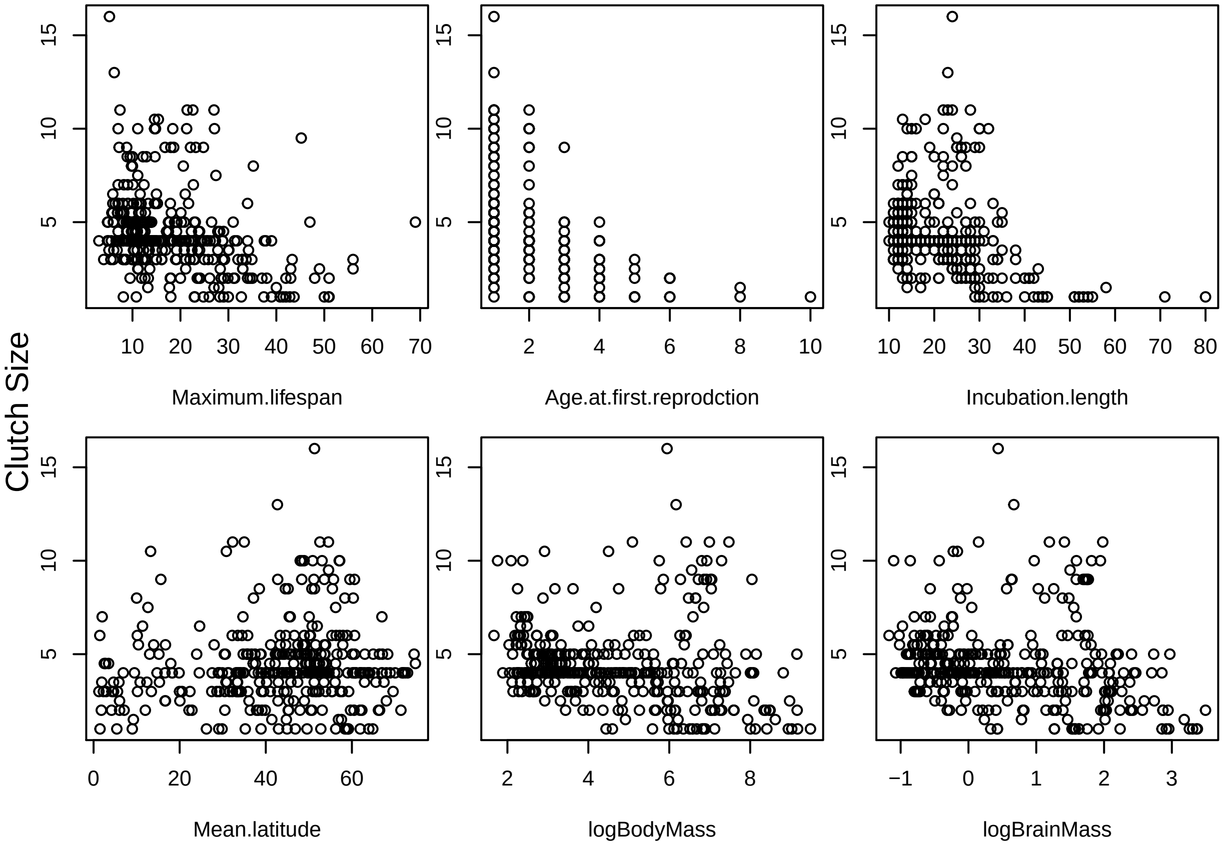


Figure 5: Plot of Clutch Size against Covariates

```
# This is more advance R, I don't expect anyone to have done this.
Ests <- plyr::ldply(Mods, function(mod) {
  CI <- confint(mod)
  c(coef=coef(mod)[2], lwr=CI[2,"2.5 %"], upr=CI[2,"97.5 %"], R2=summary(mod)$r.square)
}, .id=NULL)
rownames(Ests) <- names(Mods)

par(mfrow=c(1,2), oma=c(0,9,0,0), mar=c(4.1,1,1,1))
plot(Ests$coef.Maximum.lifespan, 1:nrow(Ests), xlim=range(Ests[,c("lwr", "upr")]),
     ylab="", yaxt="n", xlab="Estimated Coefficient")
segments(Ests$lwr, 1:nrow(Ests), Ests$upr, 1:nrow(Ests))
abline(v=0)
axis(2, gsub("\\.", " ", rownames(Ests)), at=1:nrow(Ests), las=1)

plot(Ests$R2, 1:nrow(Ests),
     xlim=c(0,0.5), ylab="", yaxt="n",
     xlab=expression(R^2))
```

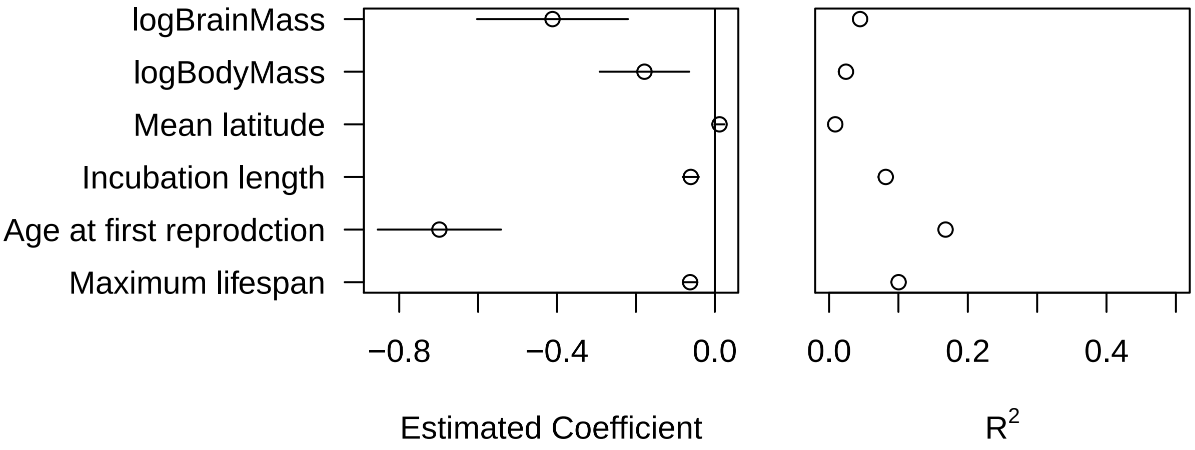


Figure 6: Estimates of parameters, and R^2, from egg clutch models with one covariate per model

Fit a model with all of the variables in it.

```
fullform <- paste("Clutch.size ~", paste(Variables[Variables!="Clutch.size"], collapse=" + "))
mod.clutch <- lm(fullform, data=BirdClutch)
#Extract the coefficients
CIs <- cbind(coef=coef(mod.clutch), confint(mod.clutch))[-1,]
```

When we fit all variables together, we get a better fit - R^2 is 23%. When we look at the coefficients, we see that age at first reproduction and brain mass both still have negative effects, and brain mass now has a larger effect. But body mass has the largest effect, and is now positive.

How (and why) do the results from fitting the individual models and from fitting one model with all variables differ?

Why did the body mass effect flip? Exercise 2 might have helped - body mass and incubation length are strongly correlated, and if we only put one in the model, the negative effect of brain mass is stronger than the positive effect of body mass.

```
par(mar=c(4.1,10,1,1))
plot(CIs[,"coef"], 1:nrow(CIs), xlim=range(CIs), ylab="", yaxt="n",
     xlab="Coefficient Estimate")
segments(CIs[,"2.5 %"], 1:nrow(CIs), CIs[,"97.5 %"], 1:nrow(CIs))
abline(v=0, lty=2)
axis(2, gsub("\\.", " ", rownames(CIs)), at=1:nrow(CIs), las=1)
```

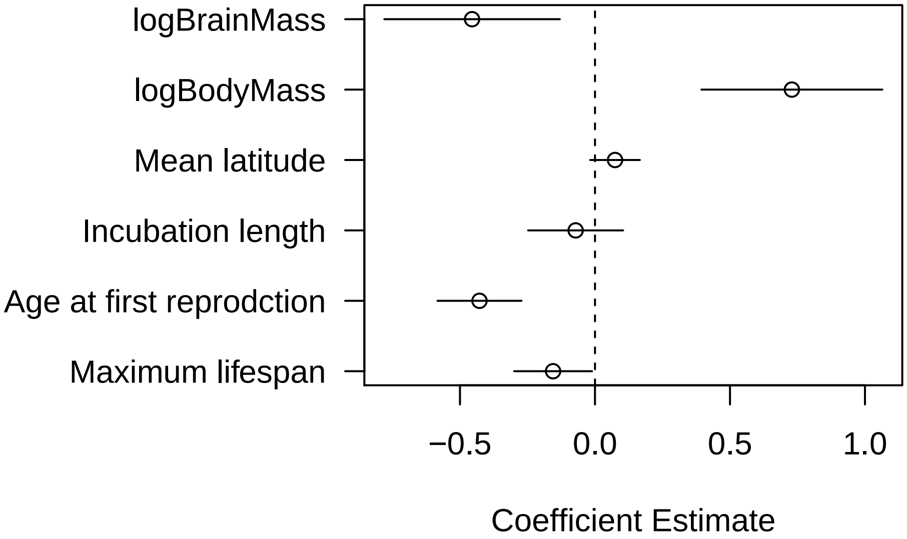


Figure 7: Estimates of parameters from egg clutch model with all covariates

Checking the model fit

```
BirdClutch$fitted <- fitted(mod.clutch)
BirdClutch$resid <- resid(mod.clutch)
BirdClutch$CooksD <- cooks.distance(mod.clutch)

par(mfrow=c(1,3), mar=c(4,3,5.5,1))
plot(BirdClutch$fitted, BirdClutch$resid, type="p", xlab="Fitted Values",
     main="Residuals vs Fitted")
qqnorm(BirdClutch$resid, main="Normal Probability Plot");
qqline(BirdClutch$resid)
plot(BirdClutch$fitted, BirdClutch$CooksD, xlab="Fitted Values", main="Cook' D")
```

We can look at the residuals, and the normal probability plot suggests that the data are positively skewed, and with thicker tails than a normal distribution. The Cook's D plot suggests that nothing is having a large effect. But the oddest thing is the plot of the residuals against the fitted values. It is made up of a series of diagonal lines.We can see what is going on by checking the original data: there are a few values that are very frequent (e.g. a clutch size of 4 eggs). Each of these is one of the diagonal lines in the regression.

```
knitr::kable(t(table(BirdBrains$Clutch.size)), rownames=c("Clutch Size", "Frequency"), caption = "Frequ
```

Table 13: Frequencies of clutch sizes

1	1.5	2	2.5	3	3.5	4	4.5	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10	10.5	11	13	16
21	5	26	10	41	21	106	25	50	15	18	4	6	2	4	6	8	1	7	2	4	1	1

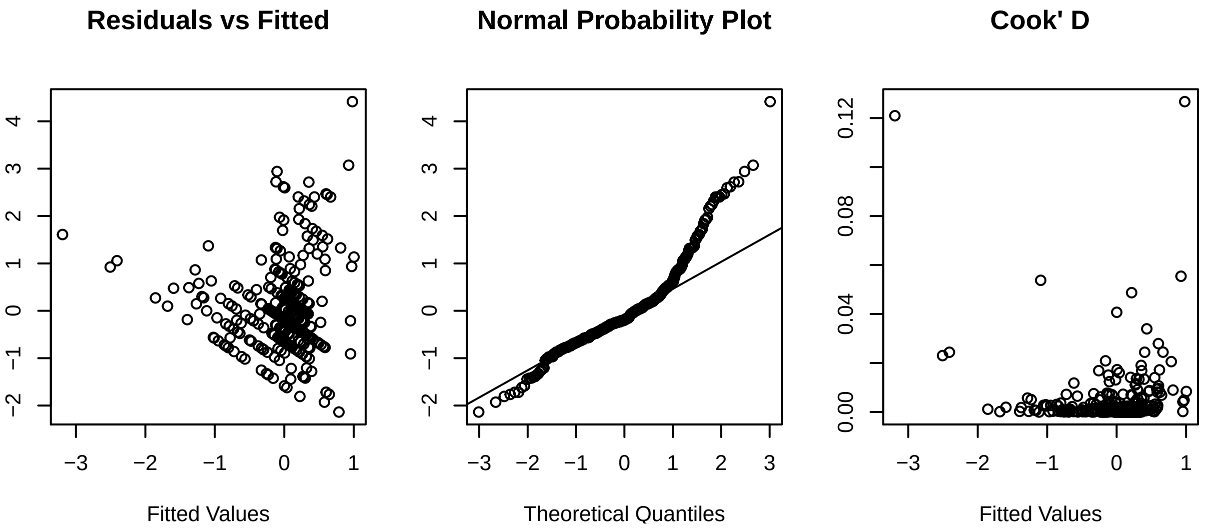


Figure 8: Plots for checking clutch size model

If we plot the residuals against each of the covariates, we can see that in general they look OK, although age at first incubation and incubation length both look like they have heteroscedasticity in their effects.

```
par(mfrow=c(2,3), mar=c(4.1,1,1,1), oma=c(0,7,0,0))
sapply(Variables[Variables!="Clutch.size"], function(v, mod, dat) {
  plot(dat[,v], resid(mod), xlab=v)
}, mod=mod.clutch, dat=BirdClutch)
```

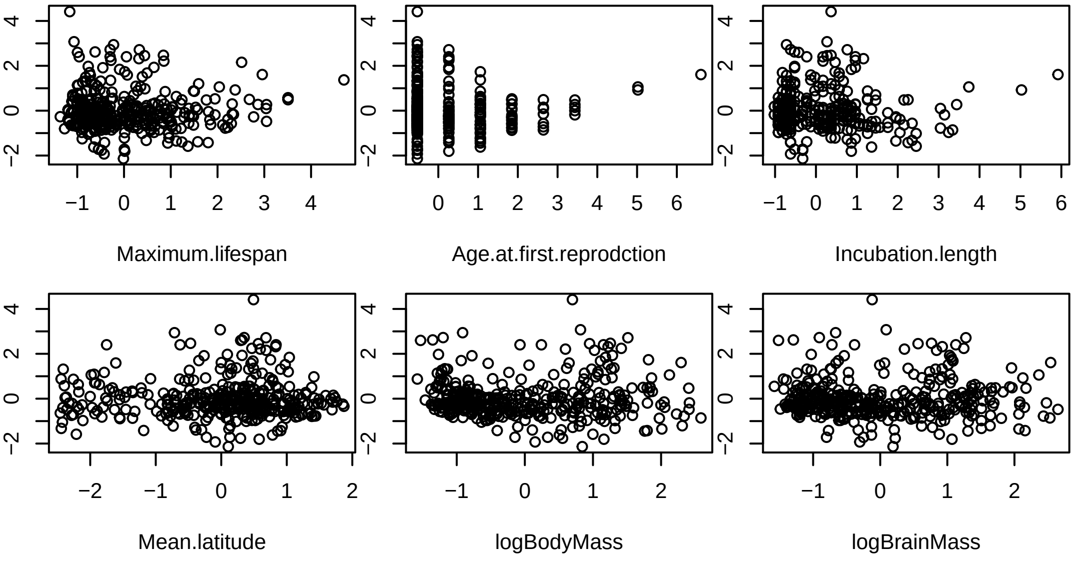


Figure 9: Residual plots against covariates