# Solutions 1

*Bob O'Hara*

*February 5, 2018*

## Problem 1

First we read the data and plot it (Figure 1).

```
library(gdata)
Link <- "http://onlinelibrary.wiley.com/store/10.1002/ece3.2961/asset/supinfo/ece32961-sup-0001-TableS1
BirdBrains <- read.xls(Link, stringsAsFactors=FALSE)
names(BirdBrains) <- gsub("\\.\\..*", "", names(BirdBrains))
BirdBrains$logBodyMass <- log(BirdBrains$Body.mass)
BirdBrains$logBrainMass <- log(BirdBrains$Brain.mass)
BirdBrains$IsParrot <- BirdBrains$Order=="Psitacciformes"
BirdBrains$IsCorvid <- BirdBrains$Family=="Corvidae"
BirdBrains$Colour <- c("rosybrown","blue","black")[1+BirdBrains$IsParrot + 2*BirdBrains$IsCorvid]
plot(log(BirdBrains$Body.mass), log(BirdBrains$Brain.mass), pch=16,
     xlab="log body mass", ylab="log brain mass", col=BirdBrains$Colour)
legend(2, 2.8, c("Corvid", "Parrot", "Something Else"), col=c("black","red","rosybrown"), pch=16)
```

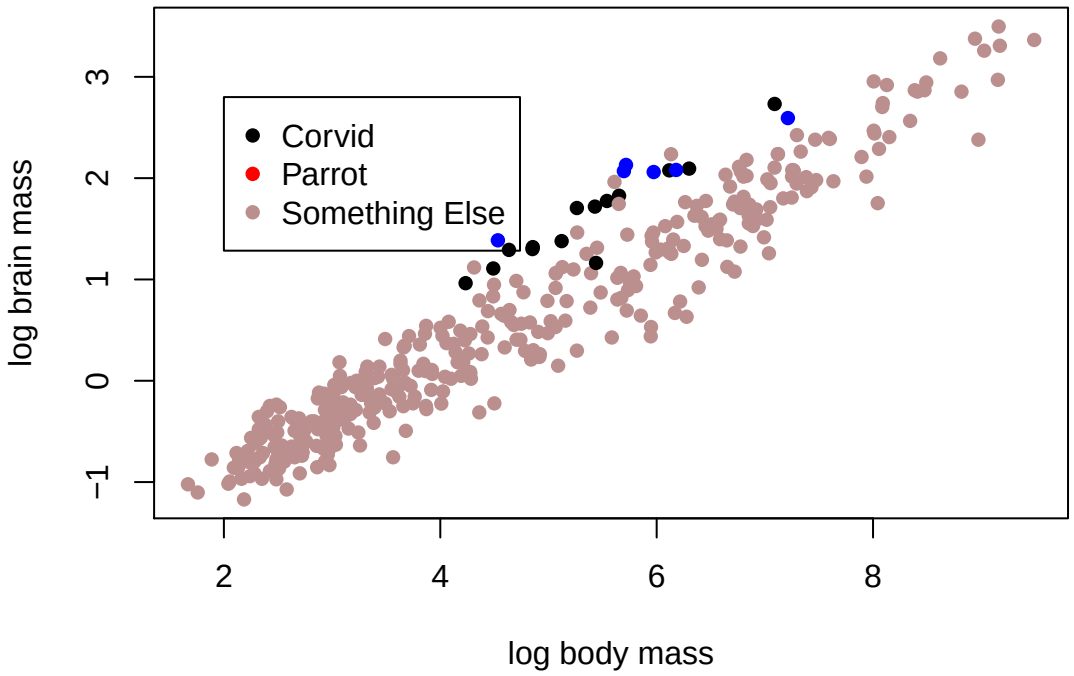



Figure 1: Body and brain mass of birds

This looks fairly linear. Note that corvids and parrots are towards the upper end of the relationship. Now we can fit the model:

```
brain.mod <- lm(logBrainMass ~logBodyMass, data = BirdBrains)
summary(brain.mod)
```

```
##
```

```
## Call:
## lm(formula = logBrainMass ~ logBodyMass, data = BirdBrains)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.87213 -0.20022 -0.01783  0.19395  0.94904
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.037838   0.043114  -47.27   <2e-16 ***
## logBodyMass  0.563159   0.008625   65.29   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3171 on 382 degrees of freedom
## Multiple R-squared:  0.9178, Adjusted R-squared:  0.9175
## F-statistic:  4263 on 1 and 382 DF,  p-value: < 2.2e-16
```

So there is a positive relationship, with a confidence interval for the slope of (0.55, 0.58): if the body size doubles, brain size goes up $e^{0.56 \times log(2)} = 1.48$ times, so it is less than proportional. In other words, big birds have smaller brains (as a proportion of body size) than smaller birds. The model explains most of the data: $R^2 = 0.92$, which is very good.

We can look at the residuals (Fig. 2). There are no obvious outliers, and the relationship looks fine, The corvids and parrots are clustered with large residuals, which suggests that there might be more structure in the data (and the Alpine Chough has a smaller residual than the other corvids). The normal probability plot suggests that the tails of the distribution are a bit thicker than we would expect if the data were normally distributed. Is this a problem? Possibly not. If the intention is to describe the relationship, then it is unlikely that this affects the estimates, particualrly of the slope. But predictions will underestimate extremes. When we look at Cook's D, we see that all of the values are small, suggesting that the model is fine in this regard.

```
# highlight if the residual is aboe 1.2 or less than -1.0
# HighlightPoints <- resid(mod) > 1.2 | resid(mod) < -1
par(mfrow=c(1,3), mar=c(4.1,4.1,2,1), lwd=2)
plot(fitted(brain.mod), resid(brain.mod),
     xlab="Fitted log brain mass", ylab="Residual",
     main="Residuals vs Fitted", lwd=3, col=BirdBrains$Colour)
# points(fitted(mod)[HighlightPoints], resid(mod)[HighlightPoints], col=2, pch=16)

qqnorm(resid(brain.mod), col=BirdBrains$Colour, main="Normal Probability Plot")
qqline(resid(brain.mod))
plot(fitted(brain.mod), cooks.distance(brain.mod),
     xlab="Fitted log brain mass", ylab="Cook' D",
     main="Cook's D", lwd=3, col=BirdBrains$Colour)
```

Overall, we have a good model: it explains most of the variation in the data, and there is only slight evidence for mis-fit. There is still some structure in the data: we can see that corvids and parrots tend to have bigger brains. This is itself interesting, so further work could look at why this is so: is it simply phylogenetic interia? Or can we explain this with other factors (e.g. sociality)? We would need to do multiple regression to find out...

Now, we can predict parrot brain size. We can see that, well, we have some predictions, with confidence intervals.

```
ParrotPred <-BirdBrains[BirdBrains$IsParrot,]
p.pred <- predict(brain.mod, newdata = ParrotPred, interval = "prediction")
rownames(p.pred) <- ParrotPred$Species.name.
```

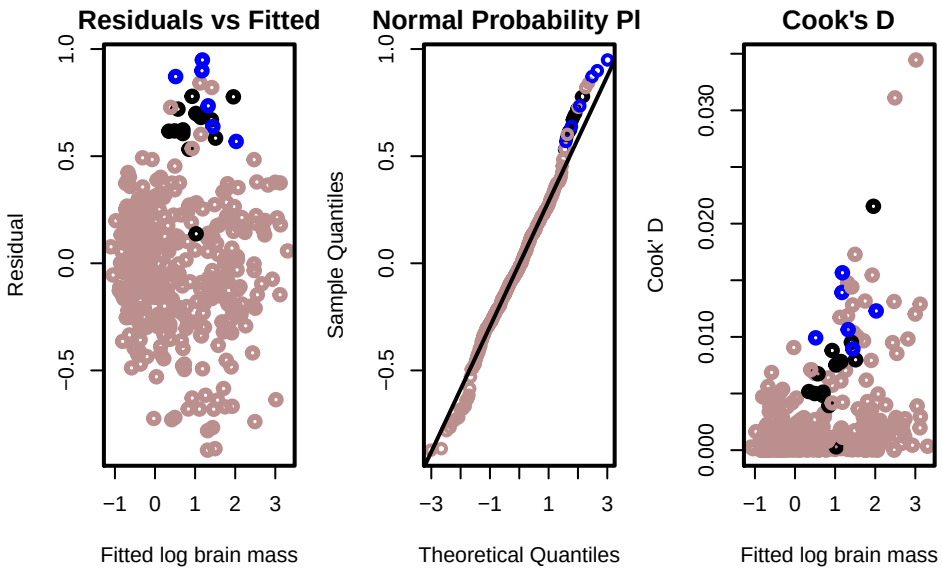



Figure 2: Residuals of body-brain mass regression

```
print(p.pred)
```

```
##                            fit        lwr      upr
## Amazona aestiva      1.3256521  0.7009294 1.950375
## Amazona amazonica    1.1815746  0.5569946 1.806155
## Amazona finschi      1.1705338  0.5459635 1.795104
## Amazona ochrocephala 1.4424965  0.8176359 2.067357
## Ara ararauna         2.0242485  1.3984077 2.650089
## Poicephalus meyeri   0.5147381 -0.1095738 1.139050
```

We can go further, and compare them to the actual values. From the residual plots, we would expect the actual values to be higher than the predicted values (the predictions are essentially the same as values with no residual). When we look at Figure 3, we see that the actual values are consistently outside the 95% confidence intervals. This just back up the point that there is more structure in the data.

```
ParrotPred <- cbind(ParrotPred, p.pred)

plot(log(BirdBrains$Body.mass), log(BirdBrains$Brain.mass), pch=16,
     xlab="log body mass", ylab="log brain mass", col=BirdBrains$Colour)
legend(2, 2.8, c("Corvid", "Parrot", "Something Else"), col=c("black","blue","rosybrown"), pch=16)
points(ParrotPred$logBodyMass, ParrotPred$fit, col="darkblue", pch=17)
segments(ParrotPred$logBodyMass, ParrotPred$lwr, ParrotPred$logBodyMass, ParrotPred$upr, col="darkblue"
```

## Problem 2

First, we simulate some good data.

```
set.seed(101)

alpha <- 5
beta <- 5
sigma <- 1 # standard deviation
```

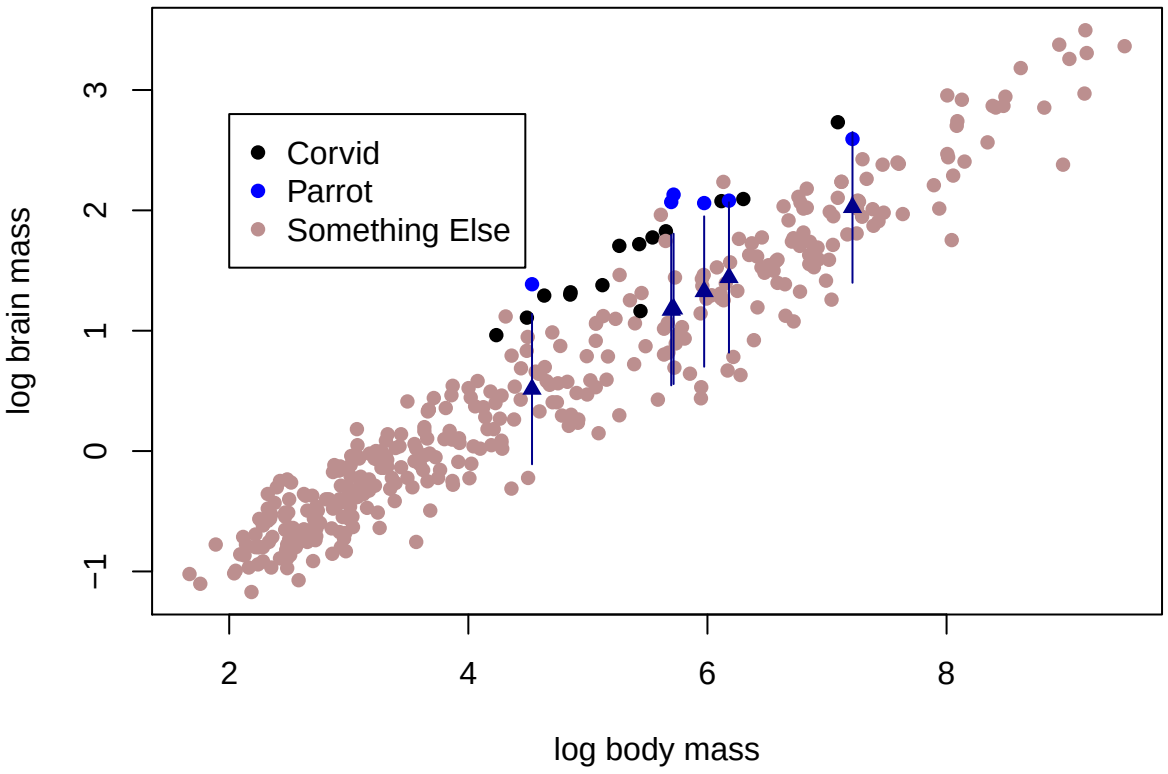



Figure 3: Predictions for parrots from body-brain size model

```
x <- runif(100, 0, 1) # random uniform distribution
mu <- alpha + beta*x
y.good <- rnorm(length(mu), mu, sigma)
mod.good <- lm(y.good~x)

plot(x, y.good)
abline(mod.good) # add the fitted line
```

(using set.seed() means that I will get the same numbers every time). The data look OK (Fig. 4), and the $R^2$ is 0.69. Looking at the residuals (Fig. 5), they are OK, as they should be.

```
par(mfrow=c(1,3))
plot(fitted(mod.good), resid(mod.good), main="Residuals vs Fitted")
qqnorm(resid(mod.good)); qqline(resid(mod.good), main="Normal Probability Plot")
plot(fitted(mod.good), cooks.distance(mod.good), main="Cook' D")
```

Now, the second data (Fig. 6).

```
mu.bad <- alpha + beta*x^2
y.bad <- rnorm(length(mu.bad), mu.bad, sigma)
mod.bad <- lm(y.bad~x)

plot(x, y.bad)
abline(mod.bad) # add the fitted line
```

This doesn't look too bad, and the $R^2$ is 0.71, so a bit higher than with the good model. The residuals largely look OK (Fig. 7), but there might be some curvature in the residuals.

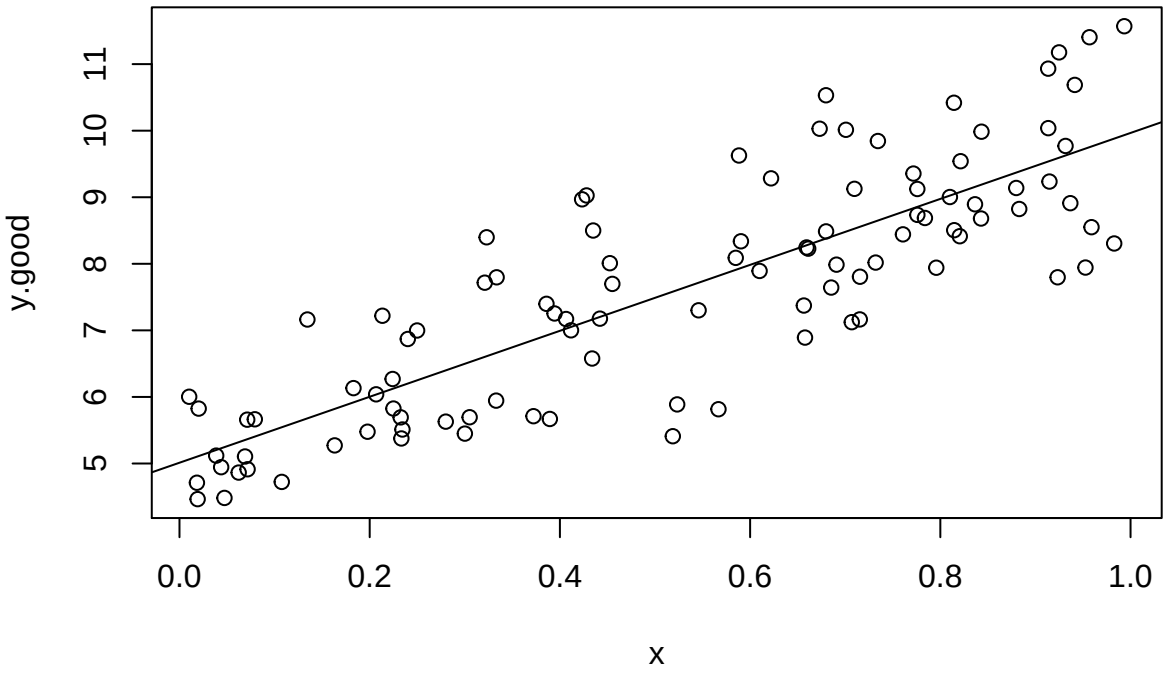



Figure 4: Some good data

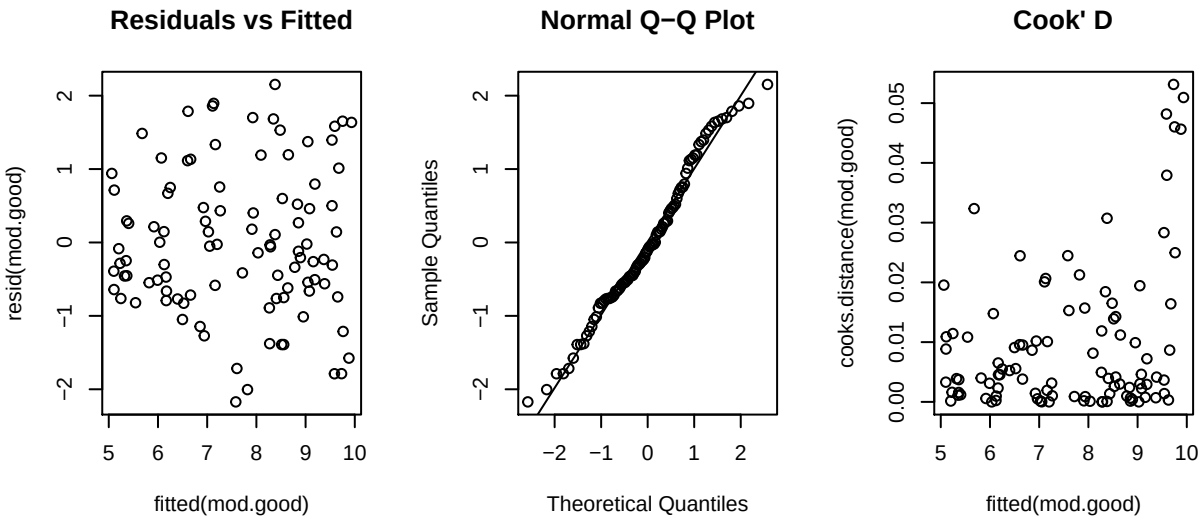



Figure 5: Model fit plots for some good data

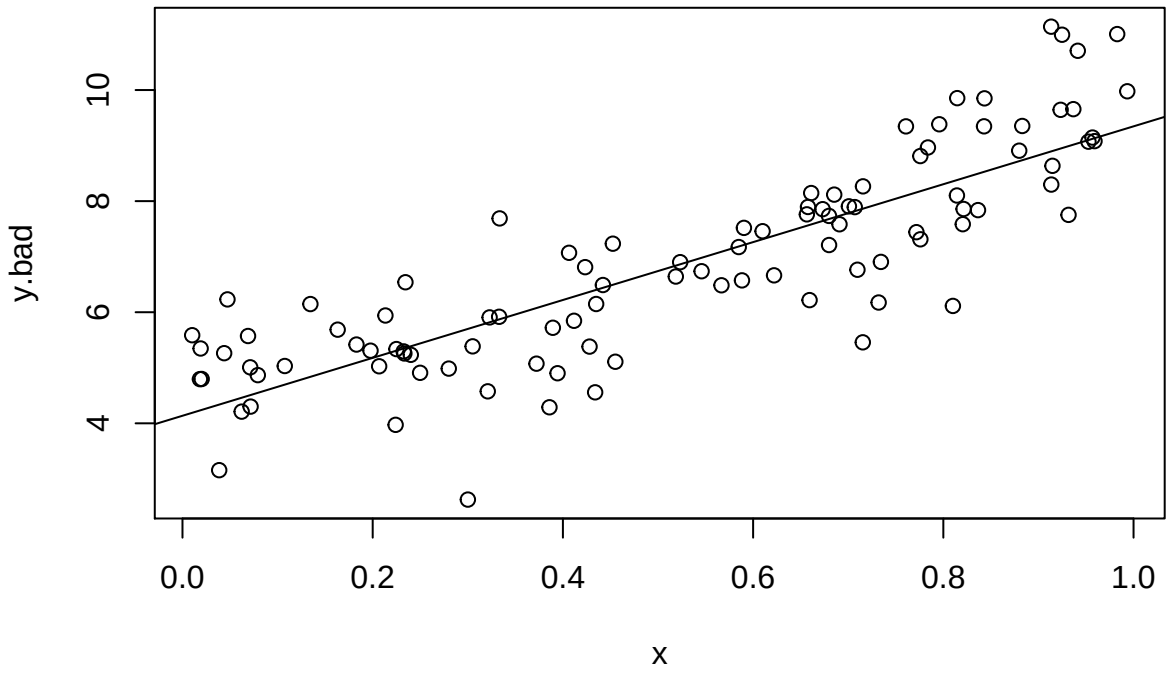



Figure 6: Some not so good data

```
par(mfrow=c(1,3))
plot(fitted(mod.bad), resid(mod.bad), main="Residuals vs Fitted")
qqnorm(resid(mod.bad)); qqline(resid(mod.bad), main="Normal Probability Plot")
plot(fitted(mod.bad), cooks.distance(mod.bad), main="Cook' D")
```

The positive curvature suggests that a Box-Cox transformation with a power less than 1 might work. So we can try a square root and a log transformation (Fig. 8). In both cases the curve in the residuals stays, and the distribution becomes more skewed. This suggest that a Box-Cox transformation is not the way to solve this problem.

```
y.bad2 <- sqrt(y.bad)
mod.bad2 <- lm(y.bad2~x)
y.bad3 <- sqrt(y.bad)
mod.bad3 <- lm(y.bad3~x)

par(mfrow=c(2,3), mar=c(4,3,5.5,1))
plot(fitted(mod.bad2), resid(mod.bad2), main="Residuals vs Fitted")
qqnorm(resid(mod.bad2)); qqline(resid(mod.bad2), main="Normal Probability Plot")
mtext("Square Root Transformation", 3, line=4)
plot(fitted(mod.bad2), cooks.distance(mod.bad2), main="Cook' D")

plot(fitted(mod.bad3), resid(mod.bad3), main="Residuals vs Fitted")
qqnorm(resid(mod.bad3)); qqline(resid(mod.bad3), main="Normal Probability Plot")
mtext("Log Transformation", 3, line=4)
plot(fitted(mod.bad3), cooks.distance(mod.bad3), main="Cook' D")
```

Now the second set of bad data (Fig. 9).

```
mu.reallybad <- alpha + beta*x
y.reallybad <- (rnorm(length(mu.reallybad), mu.reallybad, sigma))^2
```

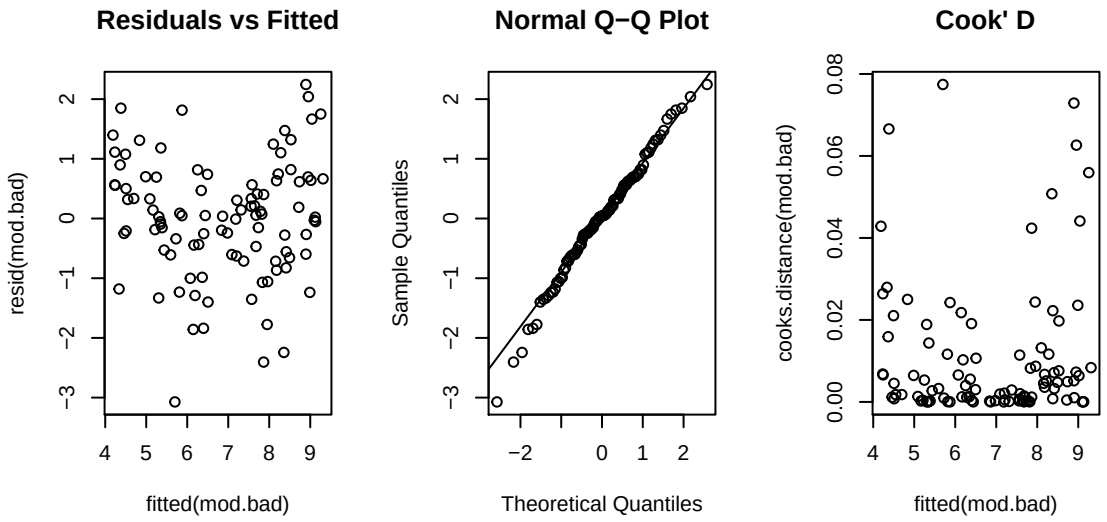



Figure 7: Model fit plots for some not so good data

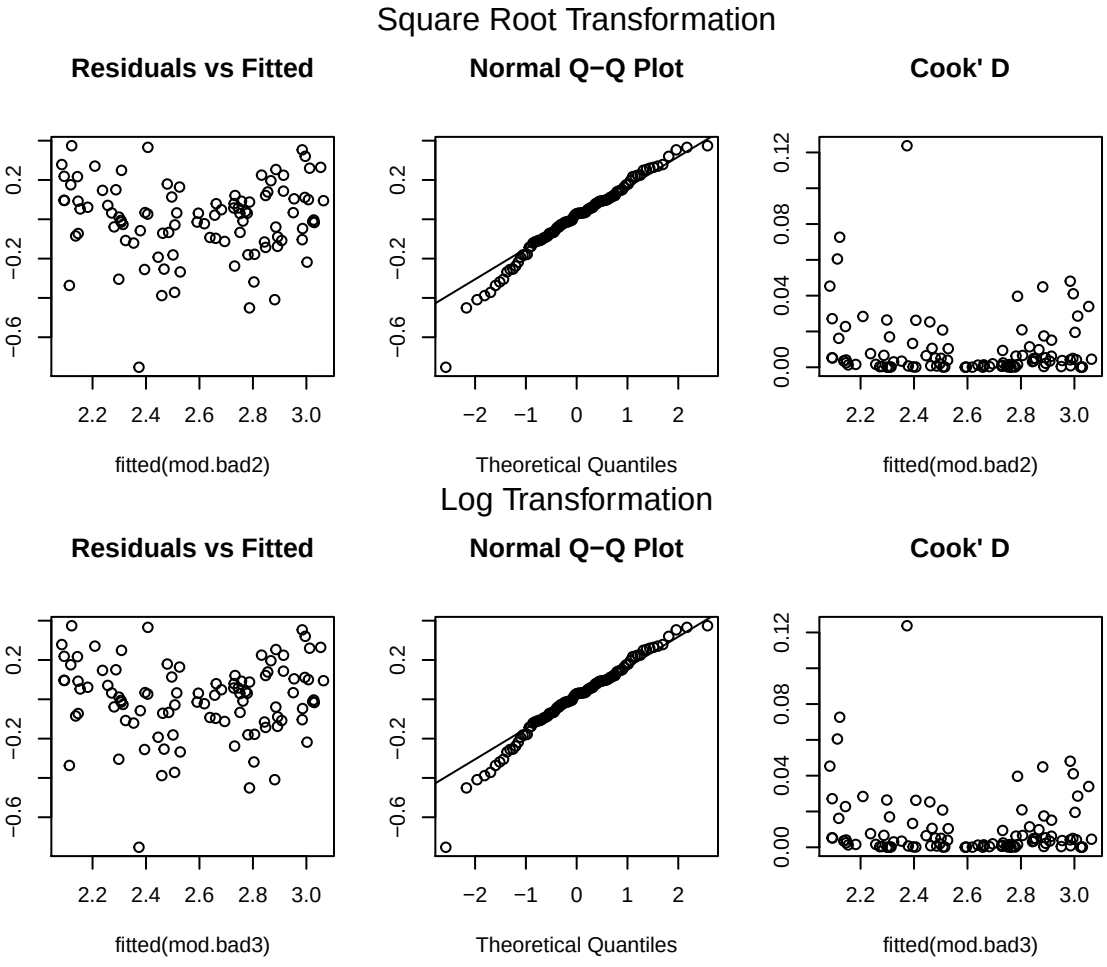



Figure 8: Transformations for some not so good data

```r
mod.reallybad <- lm(y.reallybad~x)

plot(x, y.reallybad)
abline(mod.reallybad) # add the fitted line
```

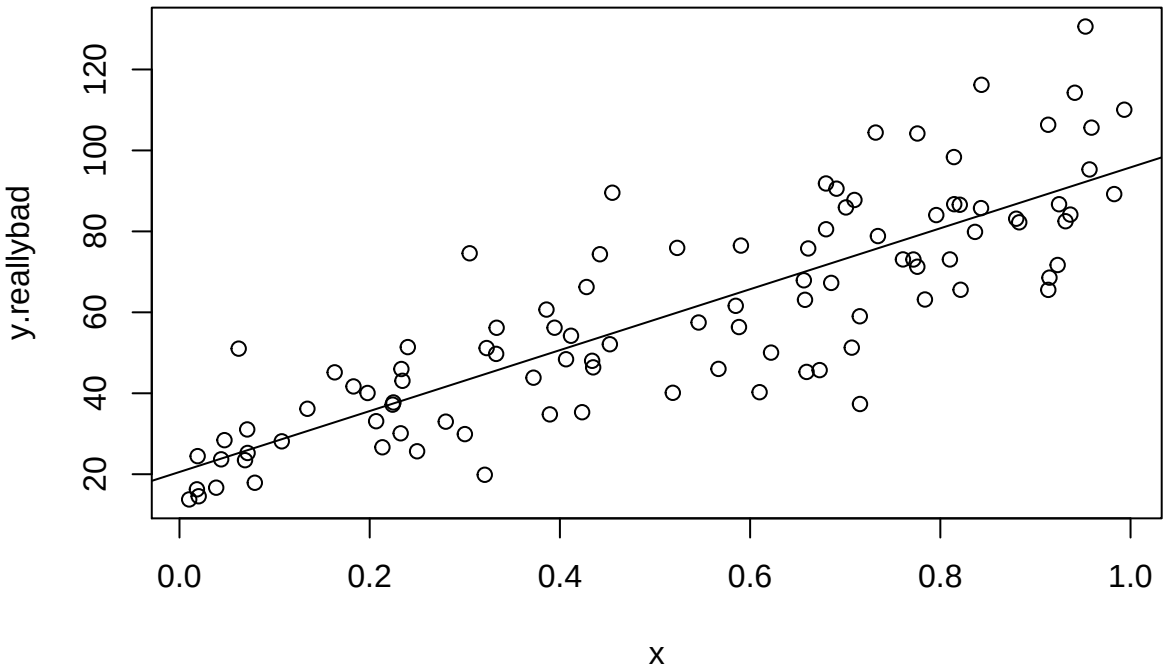



Figure 9: Some bad data

We can see some heteroscedasticity: the variation increases with the mean. This is also reflected in the residuals, and again there looks to be some curvature (Fig. 10). Notice how the larger residuals tend to be off the line in the normal probability plot: this suggests that the data might be skewed.

```r
par(mfrow=c(1,3))
plot(fitted(mod.reallybad), resid(mod.reallybad), main="Residuals vs Fitted")
qqnorm(resid(mod.reallybad)); qqline(resid(mod.reallybad), main="Normal Probability Plot")
plot(fitted(mod.reallybad), cooks.distance(mod.reallybad), main="Cook' D")
```

Using a square root helps, especially with the skewness, but the log transformation seems to solve all of the problems: the residuals look unstructured, and the normal probability plot looks fine.

```r
y.reallybad2 <- sqrt(y.reallybad)
mod.reallybad2 <- lm(y.reallybad2~x)
y.reallybad3 <- log(y.reallybad)
mod.reallybad3 <- lm(y.reallybad3~x)

par(mfrow=c(2,3), mar=c(4,3,5.5,1))
plot(fitted(mod.reallybad2), resid(mod.reallybad2), main="Residuals vs Fitted")
qqnorm(resid(mod.reallybad2)); qqline(resid(mod.reallybad2), main="Normal Probability Plot")
mtext("Square Root Transformation", 3, line=4)
plot(fitted(mod.reallybad2), cooks.distance(mod.reallybad2), main="Cook' D")

plot(fitted(mod.reallybad3), resid(mod.reallybad3), main="Residuals vs Fitted")
qqnorm(resid(mod.reallybad3)); qqline(resid(mod.reallybad3), main="Normal Probability Plot")
mtext("Log Transformation", 3, line=4)
```

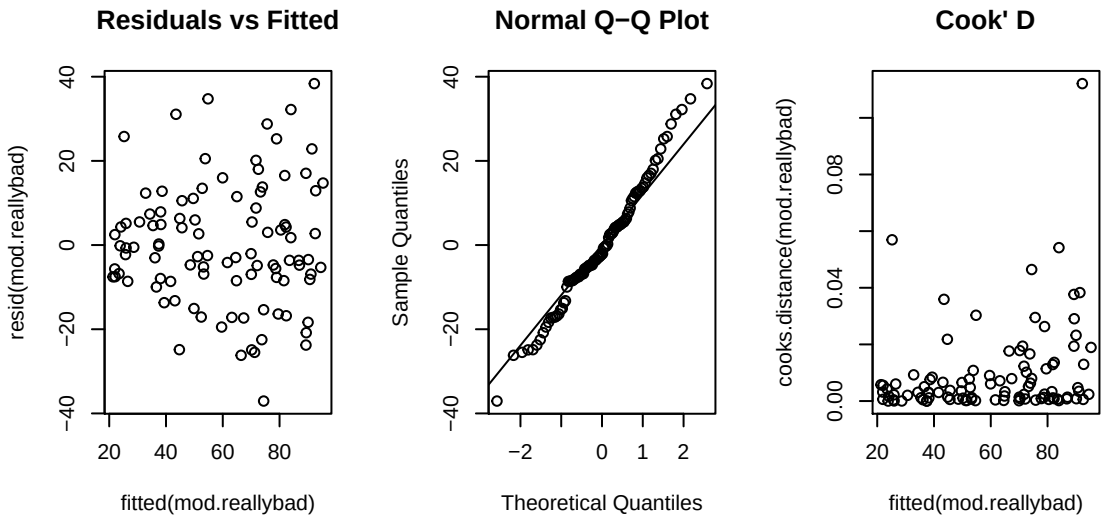



Figure 10: Model checking plots for some bad data

```
plot(fitted(mod.reallybad3), cooks.distance(mod.reallybad3), main="Cook' D")
```

## Problem 3

There are several approaches to this problem, and not necessarily a unique solution. So, first get the data and plot it (Fig. 10):

```
library(RCurl)
library(XML)
# First we need to extract the table from the webpage. This is a bit tricky.
fileURL1 <- "https://www.ineteconomics.org/perspectives/blog/"
fileURL2 <- "the-link-between-health-spending-and-life-expectancy"
fileURL3 <- "-the-us-is-an-outlier"
fileURL <- paste0(fileURL1, fileURL2, fileURL3)
xData <- getURL(fileURL, ssl.verifyPeer=FALSE)
rawdata <- readHTMLTable(xData, stringsAsFactors=FALSE)[[1]]
# Now we need to format the data extracted from the html page
names(rawdata) <- gsub("\n", "", names(rawdata))
names(rawdata) <- gsub("\r", "", names(rawdata))
names(rawdata) <- gsub("\t", "", names(rawdata))
rawdata$`Life exectancy` <- as.numeric(rawdata$`Life exectancy`)
rawdata$`Health Spending per capita` <-
  as.numeric(gsub(",", "", gsub(" \\$", "",
                                rawdata$`Health Spending per capita`)))

# Finally, plot the data
plot(rawdata$`Health Spending per capita`, rawdata$`Life exectancy`, type="n",
     xlab="Health Spending per capita ($)", ylab = "Life Expectancy")
text(rawdata$`Health Spending per capita`, rawdata$`Life exectancy`, rawdata$Country, cex=0.7)
```

The data are obviously non-linear. So first, we can try log-transforming the x-axis:

```
rawdata$logHealthSpending <- log(rawdata$`Health Spending per capita`)
plot(rawdata$logHealthSpending, rawdata$`Life exectancy`, type="n",
```

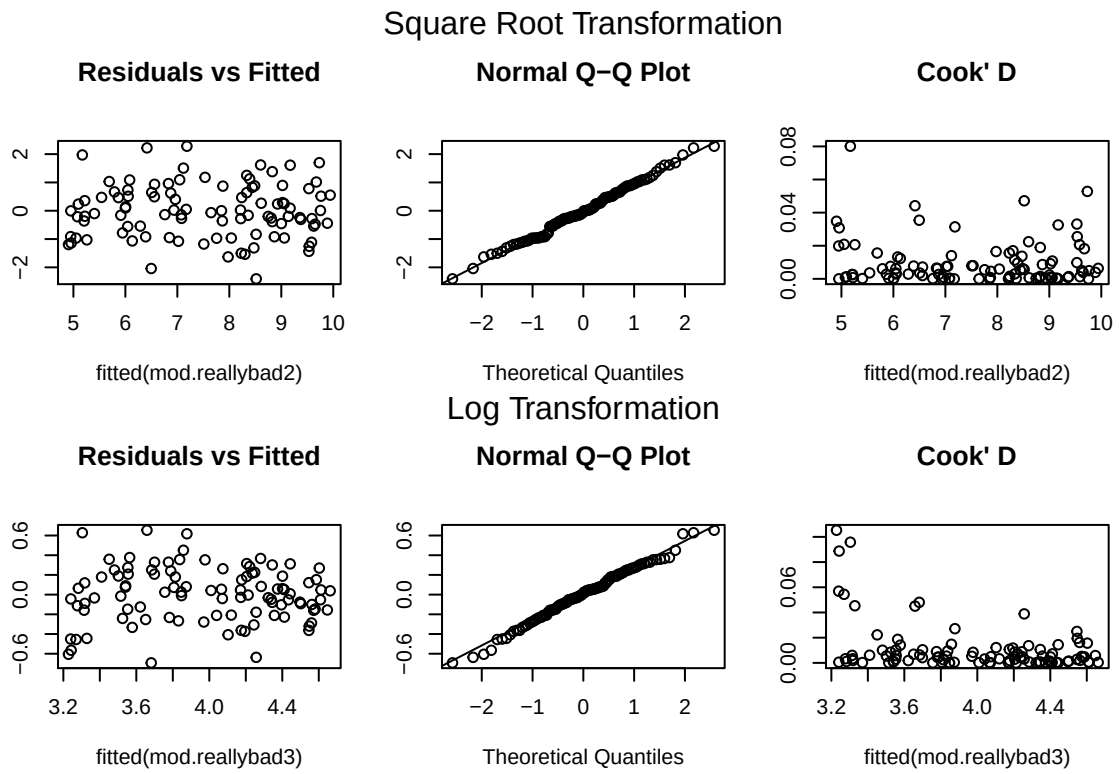



Figure 11: Transformations for some bad data

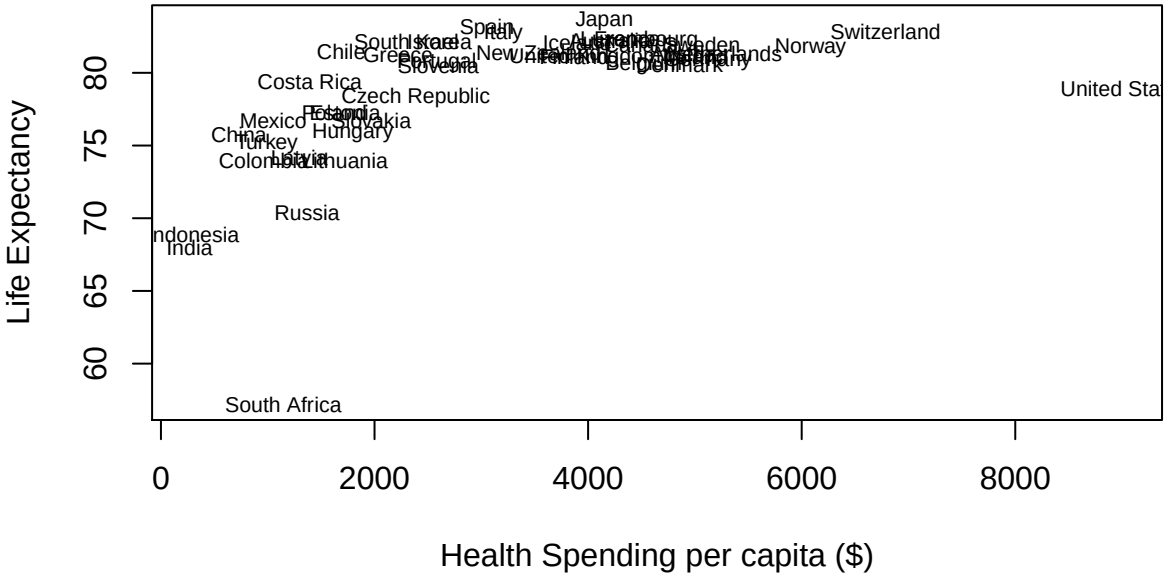
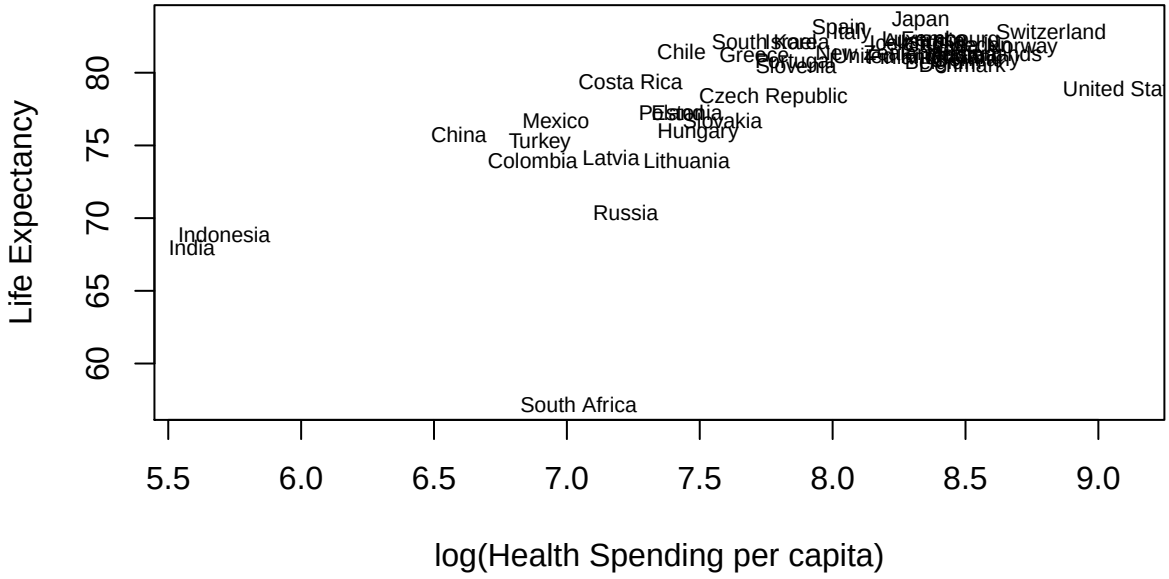



Figure 12: Life expectancy and healthcare spending

```
    xlab="log(Health Spending per capita)", ylab = "Life Expectancy")
text(rawdata$logHealthSpending, rawdata$`Life exectancy`, rawdata$Country, cex=0.7)
```




This looks better (Fig. 11), but South Africa now looks like an outlier. We can fit a model to this, and see what happens:

```
Health.mod1 <- lm(`Life exectancy` ~ logHealthSpending, data=rawdata)
summary(Health.mod1)
```

Call: lm(formula = `Life exectancy` ~ logHealthSpending, data = rawdata)

Residuals: Min 1Q Median 3Q Max -17.7787 -0.9171 0.3303 1.7471 4.6830

Coefficients: Estimate Std. Error t value Pr(>|t|)
(Intercept) 41.147 5.599 7.348 5.34e-09 ***logHealthSpending 4.800 0.713 6.732 3.95e-08*** — Signif. codes: 0 '*** 0.001 ** 0.01 * 0.05 '. 0.1 ' 1

Residual standard error: 3.573 on 41 degrees of freedom Multiple R-squared: 0.525, Adjusted R-squared: 0.5134 F-statistic: 45.31 on 1 and 41 DF, p-value: 3.95e-08

So, there is a positive effect of spending: doubling spending increases life expectancy by 3.3 years. The $R^2$ is reasonable. But when we look a the plot (Fig. 12), one country stands out, with a residual below -10. This is South Africa (not the USA!), although both ofthe US and Russia have low residuals.

```
rawdata$fitted.mod1 <- fitted(Health.mod1)
rawdata$resid.mod1 <- resid(Health.mod1)

par(mfrow=c(1,3), mar=c(4,3,5.5,1))
plot(rawdata$fitted.mod1, rawdata$resid.mod1, type="n", main="Residuals vs Fitted")
text(rawdata$fitted.mod1, rawdata$resid.mod1, rawdata$Country)
qqnorm(rawdata$resid.mod1); qqline(rawdata$resid.mod1, main="Normal Probability Plot")
plot(rawdata$fitted.mod1, cooks.distance(Health.mod1), main="Cook' D")
```

We can try a model without South Africa:

```
rawdata$NoSA <- rawdata$resid.mod1 > -10
Health.mod2 <- lm(`Life exectancy` ~ logHealthSpending, data=rawdata[rawdata$NoSA,], na.action = na.exc
summary(Health.mod2)
```

Call: lm(formula = `Life exectancy` ~ logHealthSpending, data = rawdata[rawdata$NoSA, ], na.action =

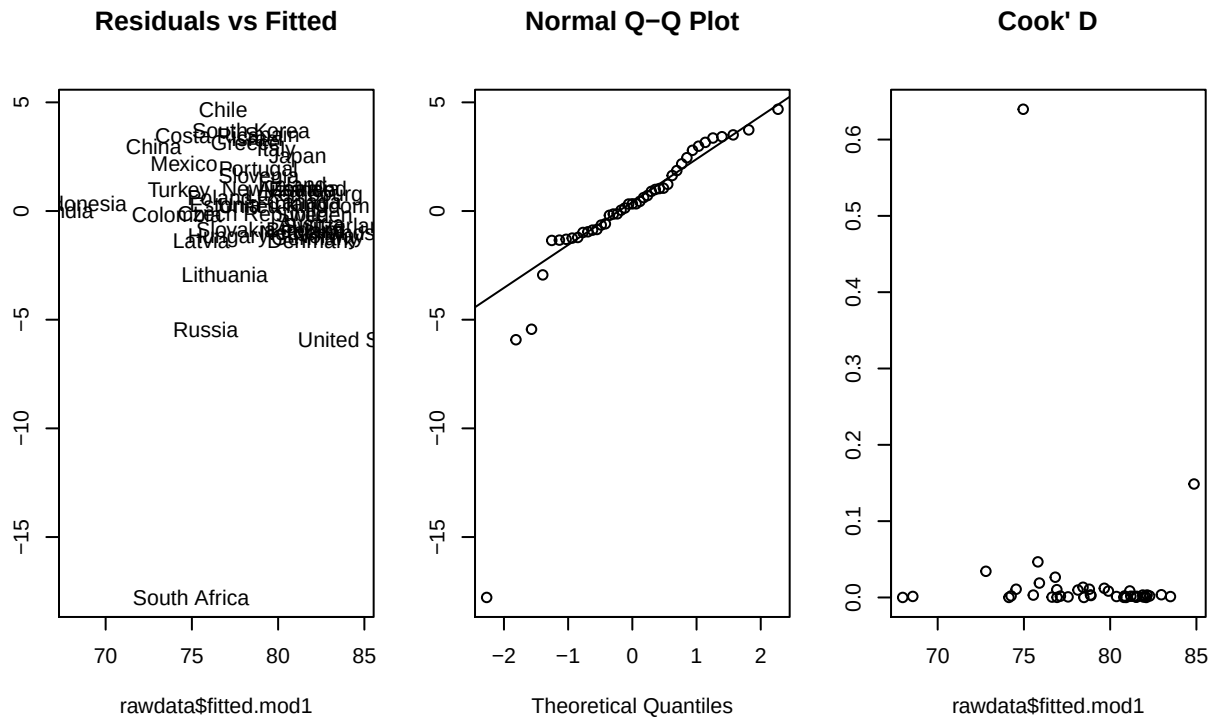



Figure 13: Model fit plots for model of life expectancy and log(healthcare spending)

na.exclude)

Residuals: Min 1Q Median 3Q Max -6.2135 -1.1312 -0.0568 1.2414 4.0291

Coefficients: Estimate Std. Error t value Pr(>|t|)
(Intercept) 46.0584 3.4824 13.226 3.44e-16 ***logHealthSpending 4.2269 0.4424 9.554 7.08e-12*** — Signif. codes: 0 '*** 0.001 '** 0.01 '* 0.05 '. 0.1 ' ' 1

Residual standard error: 2.19 on 40 degrees of freedom Multiple R-squared: 0.6953, Adjusted R-squared: 0.6877 F-statistic: 91.28 on 1 and 40 DF, p-value: 7.08e-12

Removing South Africa increases the $R^2$ from 0.52 to 0.7, which is quite a lot. Looking at the residuals (Fig. 13), this looks OK, except that Russia and the US are still outliers.

```
rawdata$fitted.mod2[rawdata$NoSA] <- fitted(Health.mod2)
rawdata$resid.mod2[rawdata$NoSA] <- resid(Health.mod2)
rawdata$CooksD.mod2[rawdata$NoSA] <- cooks.distance(Health.mod2)
rawdata[!rawdata$NoSA, c("fitted.mod2", "resid.mod2", "CooksD.mod2")] <- NA

par(mfrow=c(1,3), mar=c(4,3,5.5,1))
plot(rawdata$fitted.mod2, rawdata$resid.mod2, type="n", main="Residuals vs Fitted")
text(rawdata$fitted.mod2, rawdata$resid.mod2, rawdata$Country)
qqnorm(rawdata$resid.mod2); qqline(rawdata$resid.mod2, main="Normal Probability Plot")
plot(rawdata$fitted.mod2, rawdata$CooksD.mod2, main="Cook' D")
```

Now we could remove the US and Russia (which does improve the model). If you take this route, you have to be clear when reporting the results that you have done this. One argument agianst this is that 3 outliers is too many: it is 7% of the data. So we will take another approach (which might also be the one some people took from the start), trying a Box-Cox transformation.

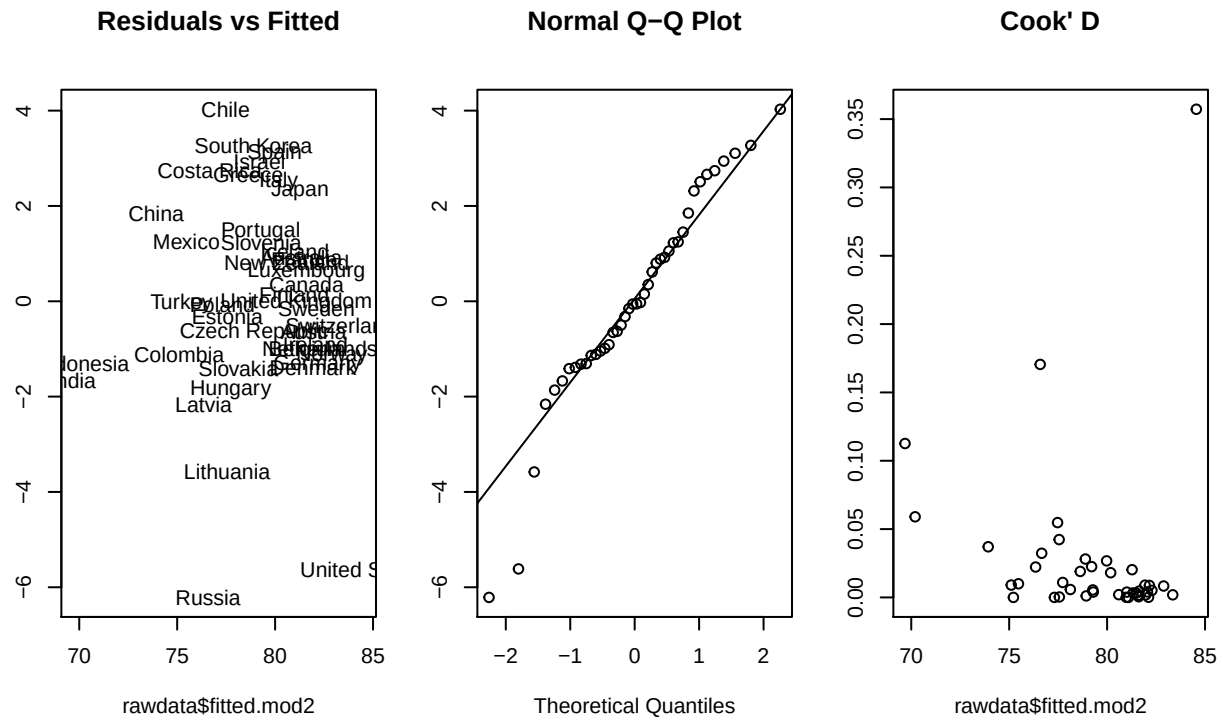



Figure 14: Model fit plots for model of life expectancy and log(healthcare spending), with South Africa removed

```
rawdata$LifeExectancySq <- rawdata$`Life exectancy`^2
rawdata$expLifeExectancy <- exp(rawdata$`Life exectancy`)
Health.mod.sq <- lm(LifeExectancySq ~ `Health Spending per capita`, data=rawdata)
Health.mod.exp <- lm(expLifeExectancy ~ `Health Spending per capita`, data=rawdata)

par(mfrow=c(2,3), mar=c(4,3,5.5,1))
plot(fitted(Health.mod.sq), resid(Health.mod.sq), main="Residuals vs Fitted")
qqnorm(resid(Health.mod.sq)); qqline(resid(Health.mod.sq), main="Normal Probability Plot")
mtext("Square Transformation", 3, line=4)
plot(fitted(Health.mod.sq), cooks.distance(Health.mod.sq), main="Cook' D")

plot(fitted(Health.mod.exp), resid(Health.mod.exp), main="Residuals vs Fitted")
qqnorm(resid(Health.mod.exp)); qqline(resid(Health.mod.exp), main="Normal Probability Plot")
mtext("Exponential Transformation", 3, line=4)
plot(fitted(Health.mod.exp), cooks.distance(Health.mod.exp), main="Cook' D")
```

If we look at the results (Fig. 14), they suggest that the square transformation under-transforms, but the exponential over-transforms. This suggests that a power >2 would be worth trying. We could do this by hand, but here I'll use the boxcox function in the MASS package. It shows that a power of around 10 is good. This might be unrealistic, but is certainly defendable as a model. If we are just predicting, we wouldn't care, as long as we got a good model.

```
Health.mod.linear <- lm(`Life exectancy` ~ `Health Spending per capita`, data=rawdata)
MASS::boxcox(Health.mod.linear, lambda = seq(1, 30, 1/10))
```

What could we conclude? We could look at non-linear models, but not in this course. So I think I would probably use the model with South Africa, the USA and Russia removed.

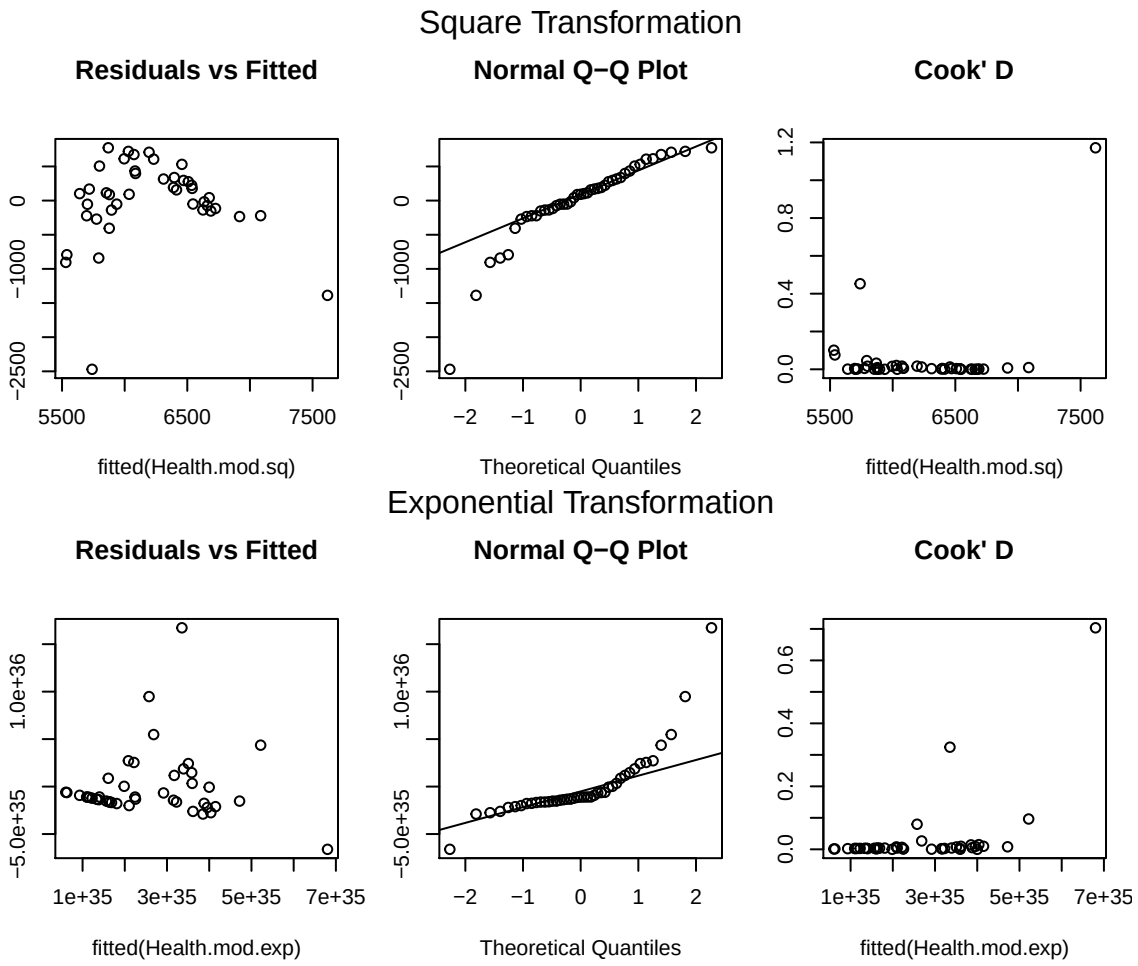



Figure 15: Estiamte of power for Box-Cox transformation for model of life expectancy and healthcare spending

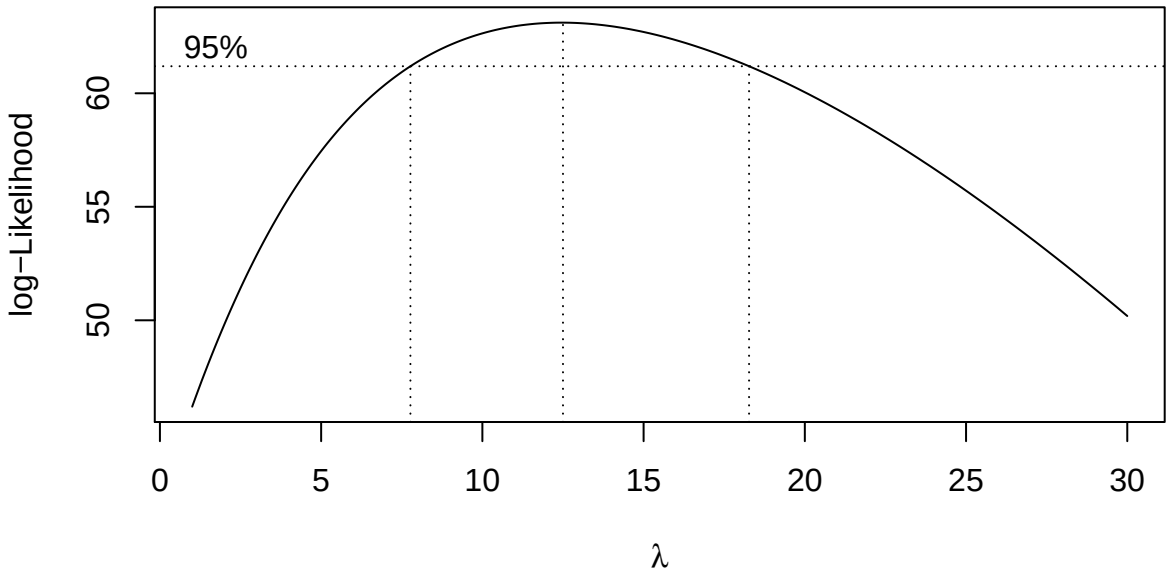



Figure 16: Box-Cox transformation for healthcare spending data

```
rawdata$NoSuperpower <- rawdata$resid.mod1 > -5
Health.modNoSuper <- lm(`Life exectancy` ~ logHealthSpending, data=rawdata[rawdata$NoSuperpower,])
summary(Health.modNoSuper)
```

Call: lm(formula = `Life exectancy` ~ logHealthSpending, data = rawdata[rawdata$NoSuperpower, ])

Residuals: Min 1Q Median 3Q Max -3.8190 -1.3378 -0.2506 1.1016 3.7941

Coefficients: Estimate Std. Error t value Pr(>|t|)
(Intercept) 45.124 2.914 15.49 < 2e-16 ***logHealthSpending 4.384 0.371 11.82 2.71e-14*** — Signif. codes: 0 '*** **0.001** ** *0.01* * 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.76 on 38 degrees of freedom Multiple R-squared: 0.7861, Adjusted R-squared: 0.7805 F-statistic: 139.7 on 1 and 38 DF, p-value: 2.707e-14

```
SuperPowers <- rawdata[!rawdata$NoSuperpower,]

SuperPowers$PredLife <- predict(Health.modNoSuper, newdata = SuperPowers)
SuperPowers$Difference <- SuperPowers$PredLife - SuperPowers$`Life exectancy`
```

This has a good $R^2$ (as it should!). The effect of doubling healthcare spending is to increase life expectancy by 3 years. From this model, we can predict what the life expectancy should be for the US, Russia and South Africa, and we see that they are 6, 6, and 19 years respectively.Who would want to live in a superpower, eh?