Basic multivariate data analysis

DC

28/09/2021

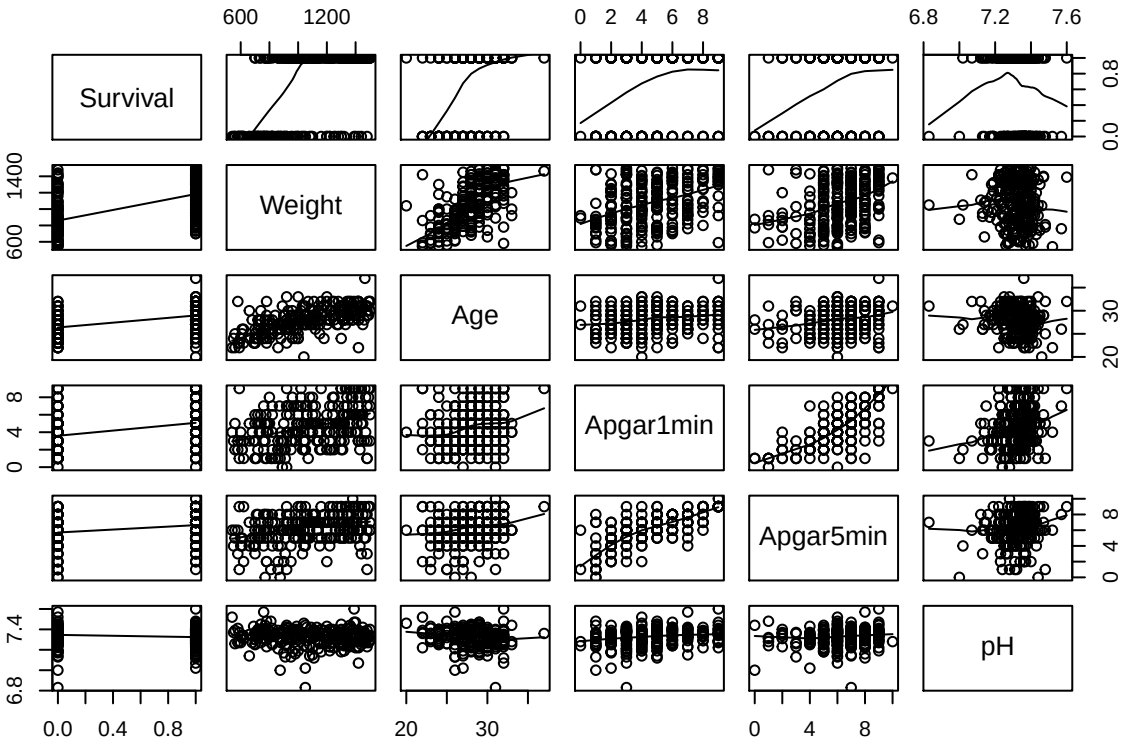
This chapter uses dataset to go through methods to visualise and analyse high dimensional data in 2-D view.

Load some useful packages first.

```
## Loading required package: carData
```

As usual, scatterplot matrix can provide us with a good initial understanding of the variable, but it's not effective for a big picture.

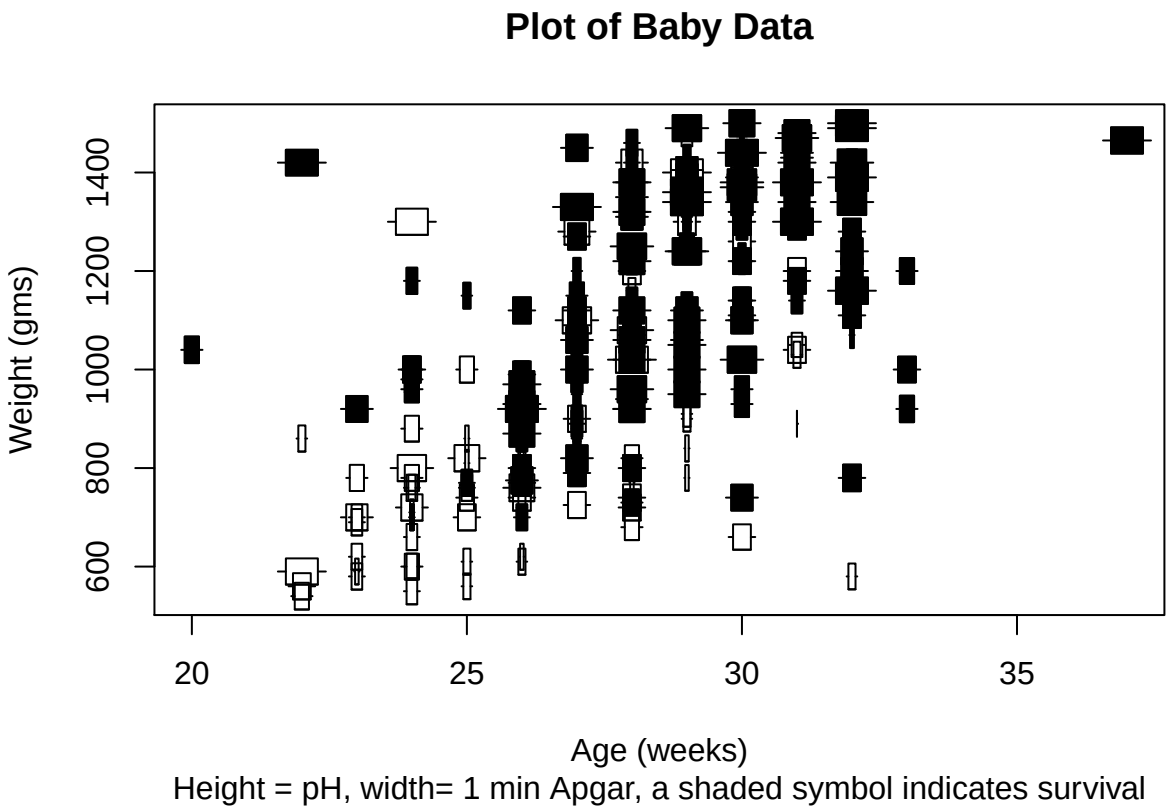
```
pairs(baby,panel=function(x,y){points(x,y);lines(lowess(x,y))})
```



We now try symbol plots to display multiple variables in 2 dimensions. This plot can present two covariates, use length and width to represent another 2 covariates and use the filled symbol to allow 0/1 colour coding.

```
plot(baby[,3],baby[,2],type="n",
     xlab="Age (weeks)",
     ylab="Weight (gms)",
```

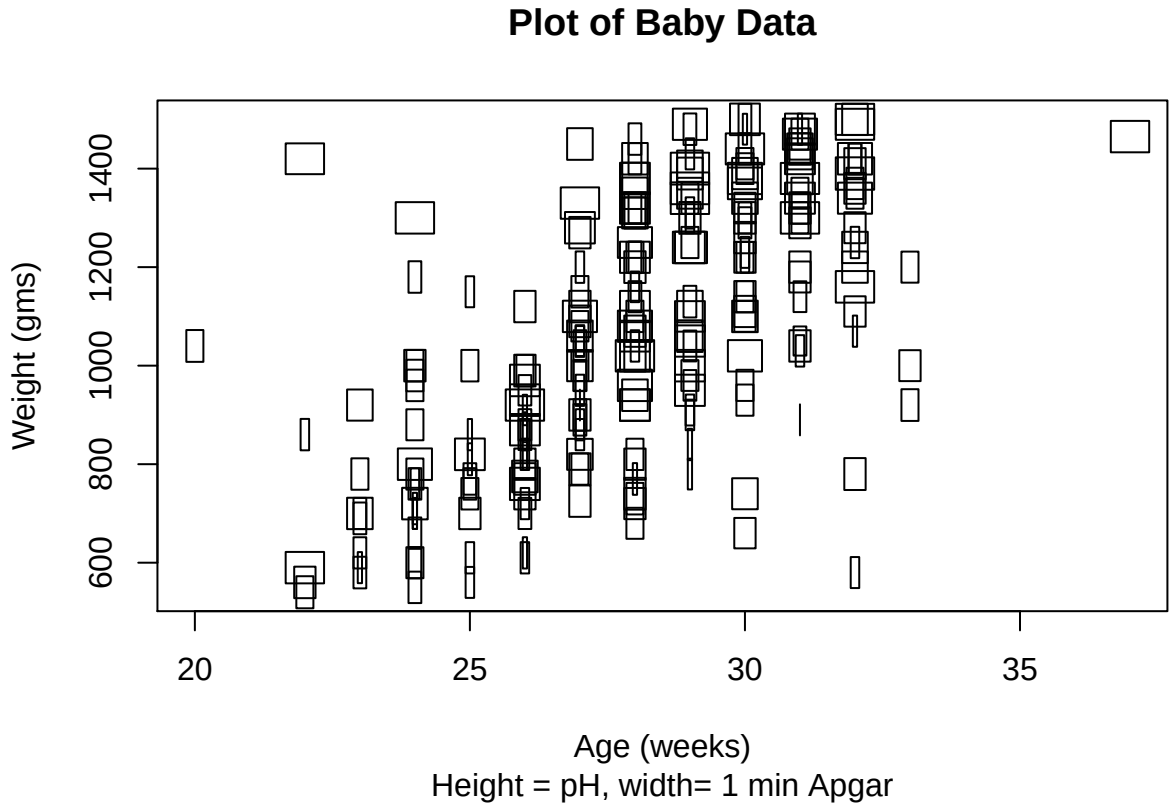
```
     main="Plot of Baby Data",
     sub="Height = pH, width= 1 min Apgar, a shaded symbol indicates survival")
surv = (baby[,1] == 1)
symbols(baby[,3],baby[,2],thermometers=cbind(baby[,4],baby[,6],surv), add=T,inches = 1/6)
```



We may also use circles and square. However, this leads to ambiguity on the size representation as we often analyse the circle area instead of radius, which exaggerates the size difference.

In general, we want the dimension of symbols to be the same as the dimension of the set of variables to represent. So to represent 2 variables, we need rectangles or stars.

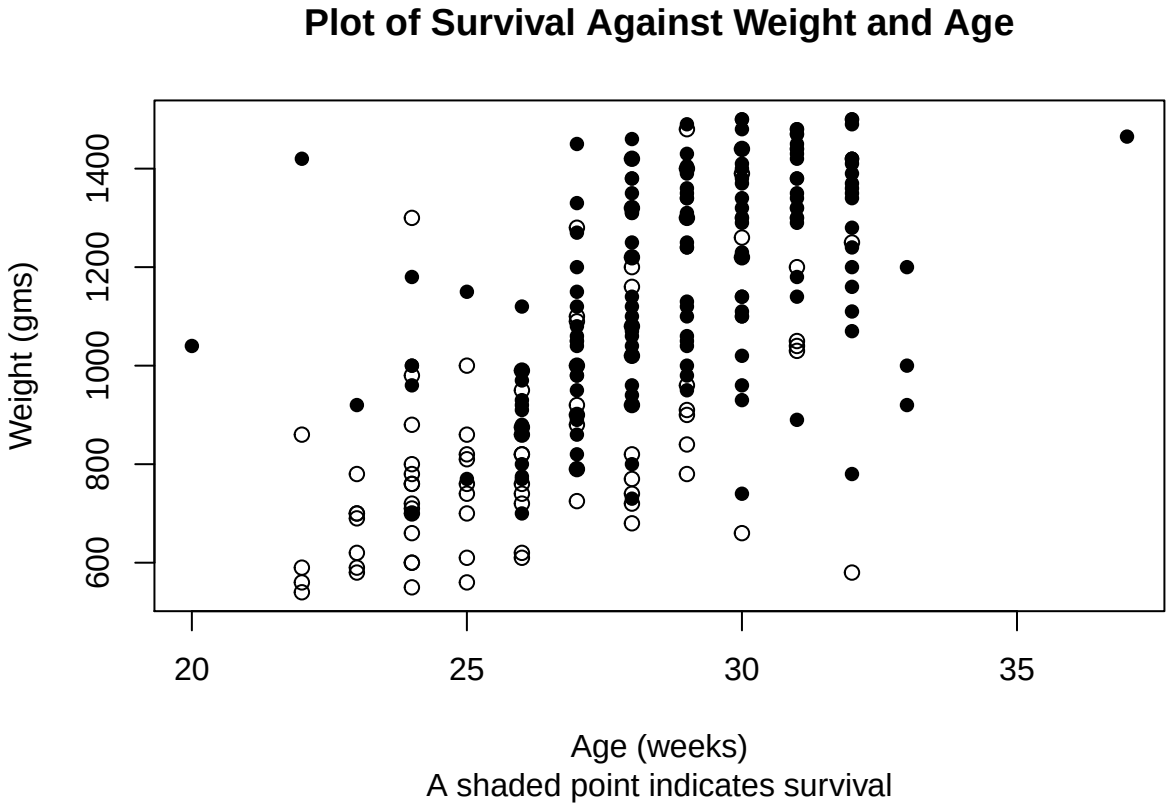
```
plot(baby[,3],baby[,2],type="n", xlab="Age (weeks)",ylab="Weight (gms)",
     main="Plot of Baby Data", sub="Height = pH, width= 1 min Apgar")
## rectangles: first argument is width, then height
symbols(baby[,3],baby[,2],rectangles = cbind(baby[,4],baby[,6]),add=T,inches=1/5)
```



We can see a trend from lower 1min Apgar to higher (wider rectangles) from lower left to upper right.

Also, given we have a categorical variable survival (0/1), we can also colour code each point to allow contrast - this gives 3 variables on one plot and such visualisation can give an initial sense of classification boundary

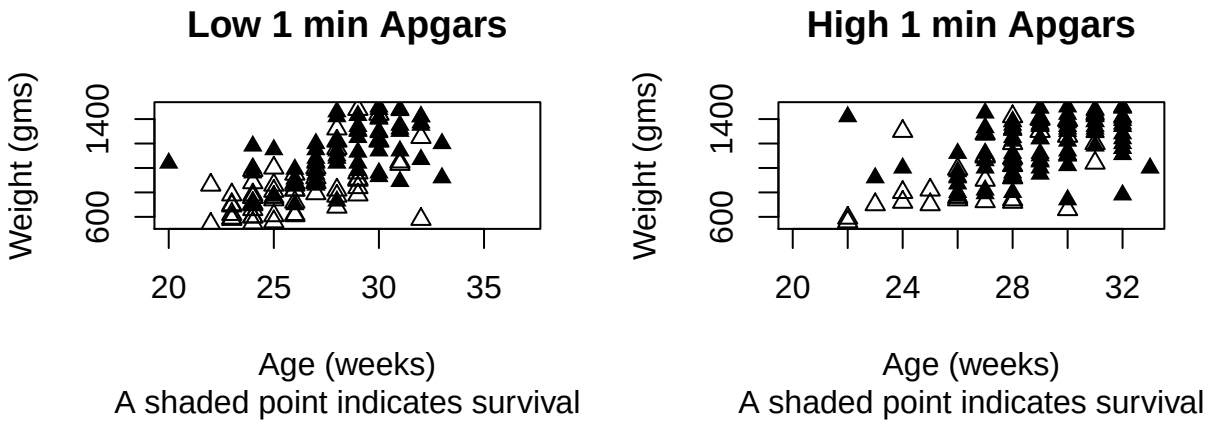
```
plot(baby[,3],baby[,2],type="n",
     xlab="Age (weeks)",
     ylab="Weight (gms)",
     main="Plot of Survival Against Weight and Age",
     sub="A shaded point indicates survival")
surv = (baby[,1] == 1)
points(baby[!surv,3],baby[!surv,2],pch=1)
points(baby[surv,3],baby[surv,2],pch=16)
```



However, when the symbols contain more information, it's better to assign coarse grouping or discretise the covariates into low/high or high/medium/low. This gives a "co-plot" that can see how the relationship changes as a third variable moves from low to high.

```
### divide the Apgars into 2 parts, low and high
ind = (baby[,4] <= 4)
par(mfrow=c(1,2),oma=c(4,0,6,0))
### First plot the symbol plot for the low apgars
plot(baby[,3],baby[,2],type="n",
     xlab="Age (weeks)",
     ylab="Weight (gms)",
     main="Low 1 min Apgars",
     sub="A shaded point indicates survival")
surv = (baby[,1] == 1)
points(baby[as.logical(ind*!surv),3],baby[as.logical(ind*!surv),2],pch=2)
points(baby[as.logical(ind*surv),3],baby[as.logical(ind*surv),2],pch=17)

### Secondly plot the symbol plot for the high apgars
plot(baby[ind,3],baby[ind,2],type="n",
     xlab="Age (weeks)",
     ylab="Weight (gms)",
     main="High 1 min Apgars",
     sub="A shaded point indicates survival")
surv = (baby[,1] == 1)
       # !ind * !surv: high apgra and not survived
points(baby[as.logical((!ind)*!surv),3],baby[as.logical((!ind)*!surv),2],pch=2)
points(baby[as.logical((!ind)*surv),3],baby[as.logical((!ind)*surv),2],pch=17)
mtext("Plot of Survival against Weight and Age",3,1,outer=T,cex=1.5)
```

Plot of Survival against Weight and Age



This plot first shows that there is a larger proportion of survival cases at high 1min Apgars. In addition, we can look at the position of relationship and see that the data cloud moves further right as 1min Apgars increases, so high Apgars tends to have older age.

R has a co-plot function but does not seperate out the range, which creates difficulty in interpretation.

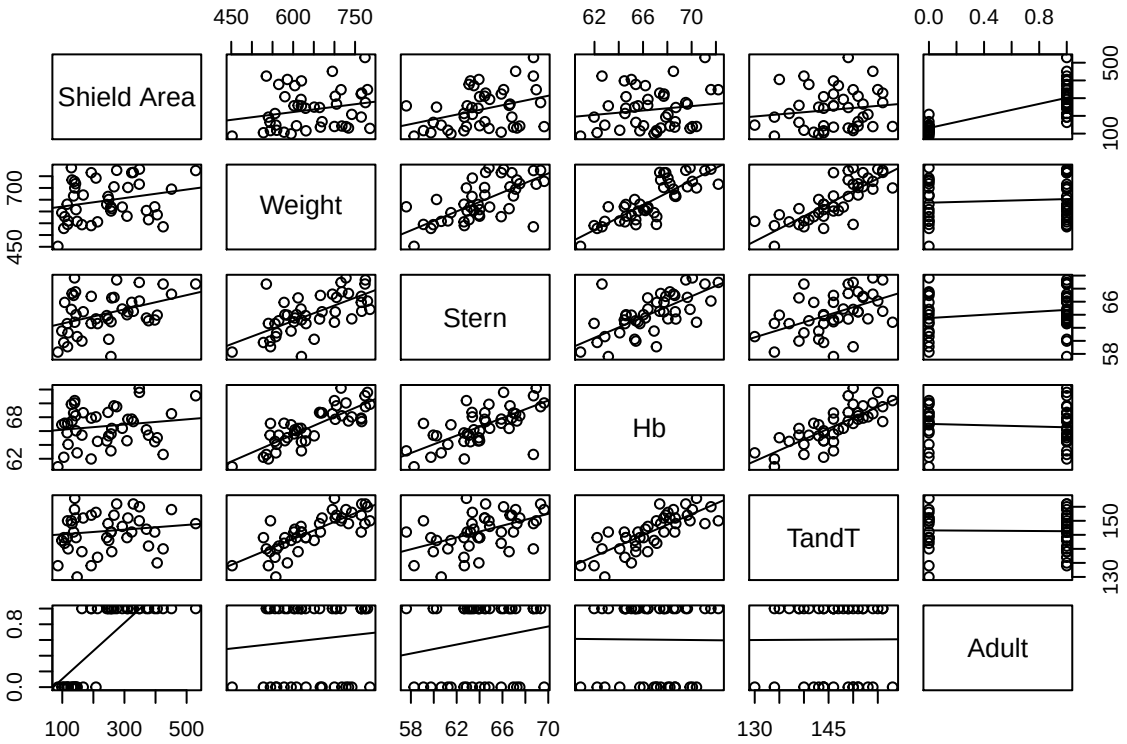
Now we look at Moorhen dataset to summarise what we have learnt on analysing multi-dimensional data.

- Step 1: Brief exploration - scatterplot matrix

```
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following objects are masked _by_ '.GlobalEnv':
##
##     cement, cpus, eqscplot, forbes, gehan, genotype, hills, leuk,
##     mammals, mcycle, painters, road, rotifer, ships, snails, stdres,
##     steam, stormer, studres, survey, write.matrix
```

```
## Provide a linear fit to the scatterplot
pairs(moorhen,
      panel=function(x,y){points(x,y);abline(rlm(y~x))})
```



We can see that shield area has some relationship with covariates but all with large spread. The most evident relationship is with the factor adult - adult tends to be much higher shiled area. Also, weight and stern and stern and Hb are highly correlated, sugges

- Step 2: covariate space analysis

Given the observed correlation between predictors, we use PCA to explore the effective dimension in predictors.

```
y = moorhen[,1]
x = moorhen[,-1]
w = sweep(x,2,apply(x,2,mean))
s = prcomp(w %*% diag(1/sqrt(diag(crossprod(w)))))
cumsum(s$sdev^2)/sum(s$sdev^2)
```

```
## [1] 0.6191796 0.8267471 0.9314980 0.9670436 1.0000000
```

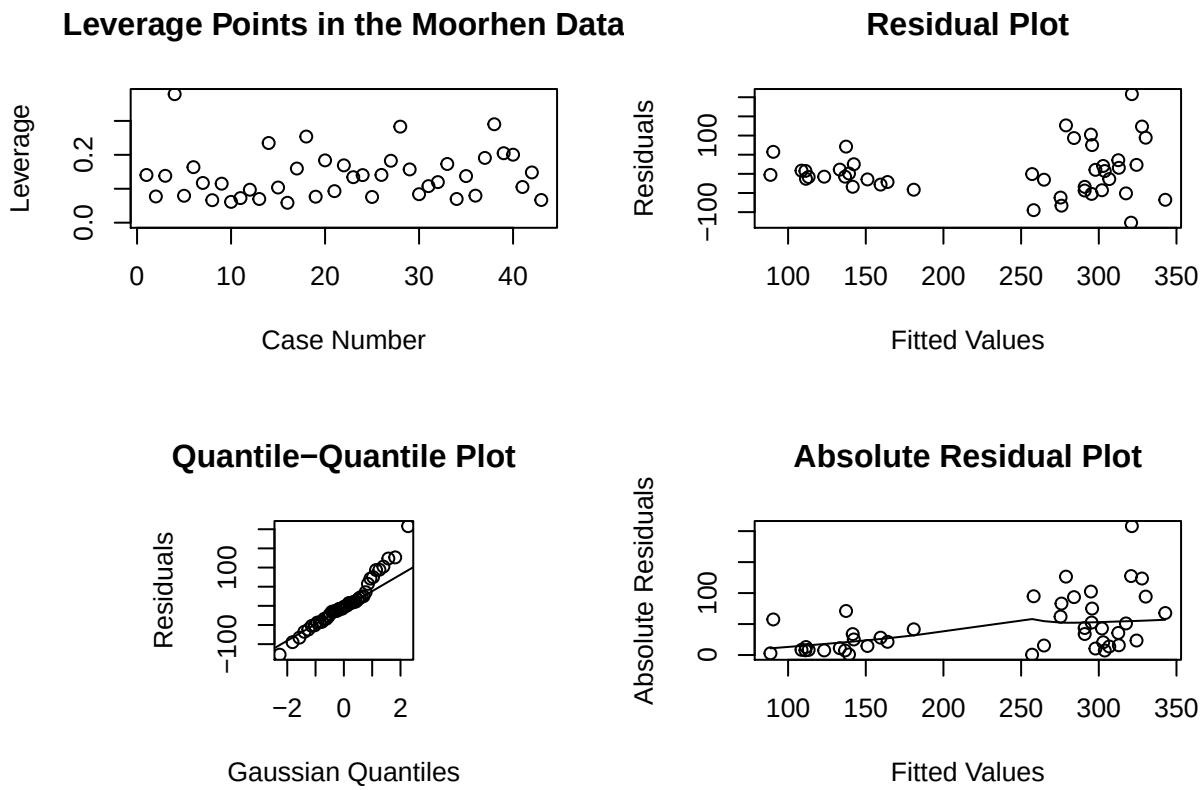
We can see that the first 3 principal components captured 93.1% of variance in the predictors. This implies that there are quite strong correlation between predictors and our effective dimension is only around 3.

- Step 3: tentative model fit

We now turn to a model-based apporach. At this stage, we have no specific idea of model form, so we fit a linear model to see there is any problem or a linear model is adequate.

```
fit = rlm(y~x)
### Model diagnostics
# Leverage plot
```

```r
par(mfcol=c(2,2),oma = c(0,0,0,0))
plot(1:43,hat(x),ylim=c(0,max(hat(x))),main="Leverage Points in the Moorhen Data",
xlab="Case Number",ylab="Leverage")
# Normal Q-Q plot
par(pty="s")
qqnorm(residuals(fit),
       main="Quantile-Quantile Plot",
       xlab="Gaussian Quantiles",
       ylab="Residuals")
qqline(residuals(fit)) # plausibly normal, some heavy tail.
par(pty="m")

# residual vs fitted
plot(fitted(fit),residuals(fit), main="Residual Plot",xlab="Fitted Values",
     ylab="Residuals")

# absolute residuals with loess curve
plot(fitted(fit),abs(residuals(fit)),main="Absolute Residual Plot",
     xlab="Fitted Values", ylab="Absolute Residuals")
lines(lowess(fitted(fit),abs(residuals(fit))))

mtext("Diagnostics for the model fit to the Moorhen data",
      side=3,line=2,outer=T,cex=1.5)
mtext("The model includes all variables on the raw scale",
      side=1,line=2,outer=T)
```

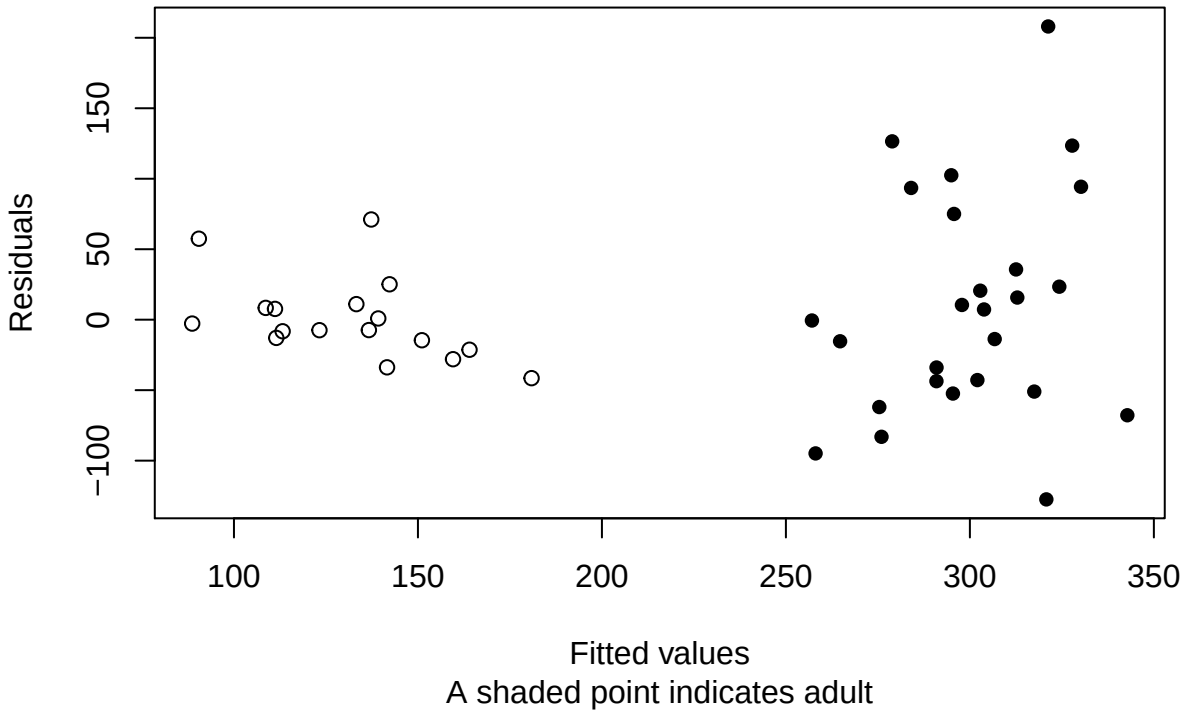


The model residuals show some right skewedness that slightly violates the normality assumption. However, the residuals also show clear heteroscedasticity as the spread of residuals increases as fitted values increases,

and the lowess curve for absolute residuals is not flat and shows some increasing trend. This non-constant variance suggests that a log-transformation may be needed to resolve the issue.

We also see clusters in residuals, so we explore it further by colour-coding it with a categorical variable.

```
adult=moorhen[,6]
plot(fitted(fit)[adult==0],residuals(fit)[adult==0],sub="A shaded point indicates adult",
xlab="Fitted values",ylab="Residuals",xlim=range(fitted(fit)),ylim=range(residuals(fit)))
points(fitted(fit)[adult==1],residuals(fit)[adult==1],pch=16)
```

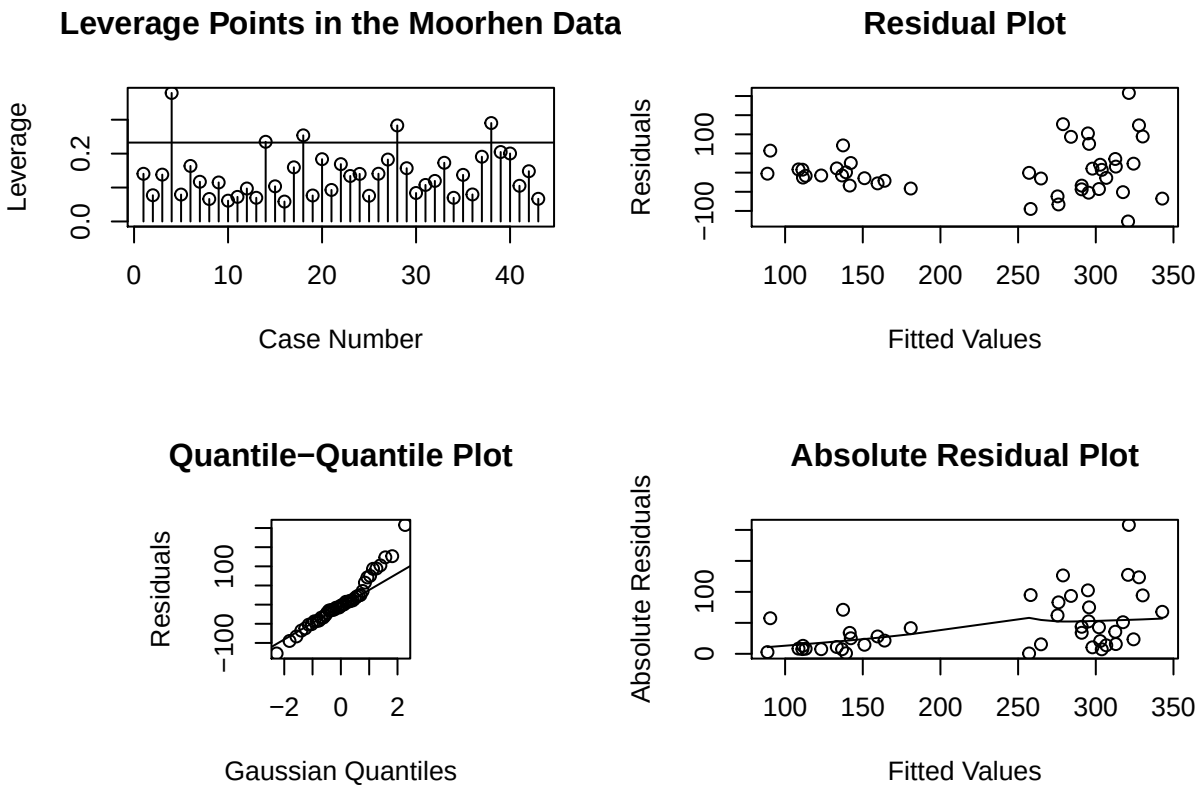


We can clearly see that the right cluster belongs to adult.

- Step 4: after getting info about transformation of response, re-fit the model and do diagnostics

```
fit2 = rlm(log(y)~x)
par(mfcol=c(2,2))
plot(c(1:43),hat(x),ylim=c(0,max(hat(x))),
     main="Leverage Points in the Moorhen Data",
     xlab="Case Number",
     ylab="Leverage")
segments(c(1:43),0,c(1:43),hat(x))# haven't change, leverage only depends on x
abline(h=10/43) #2p/n

par(pty="s")
qqnorm(residuals(fit),
       main="Quantile-Quantile Plot",
       xlab="Gaussian Quantiles",
       ylab="Residuals")
qqline(residuals(fit))
par(pty="m")
```

```r
plot(fitted(fit),residuals(fit),
     main="Residual Plot",
     xlab="Fitted Values",
     ylab="Residuals")# change much better, to homoscedasticity

plot(fitted(fit),abs(residuals(fit)),
     main="Absolute Residual Plot",
     xlab="Fitted Values",
     ylab="Absolute Residuals")
lines(lowess(fitted(fit),abs(residuals(fit))))
```



Now the spread of residuals become more consistent and the heavier right tail becomes less evident, so this transformation is reasonable.

- Step 5: resolve the multicollinearity & model selection

We previously use PCA to find that the predictor space can be represented by a 3-D subspace. So we resort to model selecton algorithms to leave out less important variables.

```r
# best subset selection over all 32 candidate model
regsubsets.out = regsubsets(log(y) ~ x, data=as.data.frame(moorhen),
                            nbest=10,method="exhaustive")
summary(regsubsets.out)
```

```
## Subset selection object
## Call: regsubsets.formula(log(y) ~ x, data = as.data.frame(moorhen),
##     nbest = 10, method = "exhaustive")
```

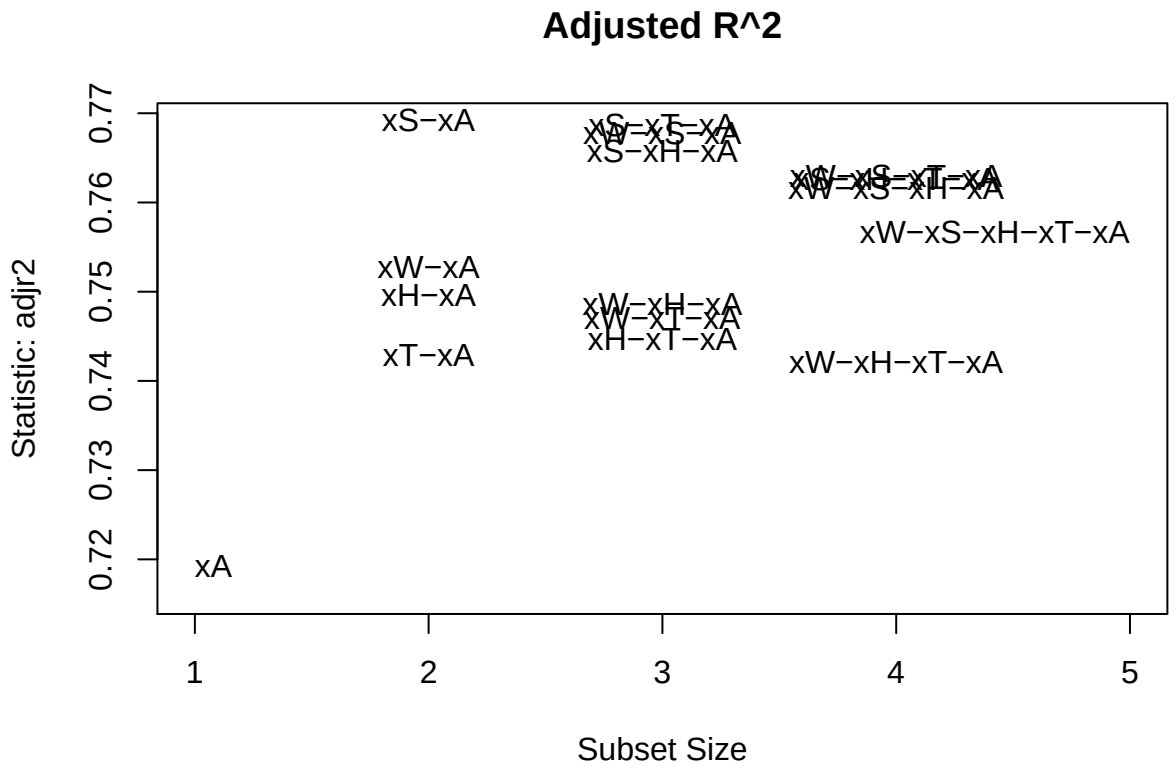
```
## 5 Variables  (and intercept)
##          Forced in Forced out
## xWeight      FALSE      FALSE
## xStern       FALSE      FALSE
## xHb          FALSE      FALSE
## xTandT       FALSE      FALSE
## xAdult       FALSE      FALSE
## 10 subsets of each size up to 5
## Selection Algorithm: exhaustive
##           xWeight xStern xHb xTandT xAdult
## 1  ( 1 )  " "     " "    " " " "    "*"
## 1  ( 2 )  " "     "*"    " " " "    " "
## 1  ( 3 )  "*"     " "    " " " "    " "
## 1  ( 4 )  " "     " "    "*" " "    " "
## 1  ( 5 )  " "     " "    " " "*"    " "
## 2  ( 1 )  " "     "*"    " " " "    "*"
## 2  ( 2 )  "*"     " "    " " " "    "*"
## 2  ( 3 )  " "     " "    "*" " "    "*"
## 2  ( 4 )  " "     " "    " " "*"    "*"
## 2  ( 5 )  " "     "*"    "*" " "    " "
## 2  ( 6 )  "*"     "*"    " " " "    " "
## 2  ( 7 )  " "     "*"    " " "*"    " "
## 2  ( 8 )  "*"     " "    "*" " "    " "
## 2  ( 9 )  "*"     " "    " " "*"    " "
## 2  ( 10 ) " "     " "    "*" "*"    " "
## 3  ( 1 )  " "     "*"    " " "*"    "*"
## 3  ( 2 )  "*"     "*"    " " " "    "*"
## 3  ( 3 )  " "     "*"    "*" " "    "*"
## 3  ( 4 )  "*"     " "    "*" " "    "*"
## 3  ( 5 )  "*"     " "    " " "*"    "*"
## 3  ( 6 )  " "     " "    "*" "*"    "*"
## 3  ( 7 )  "*"     "*"    "*" " "    " "
## 3  ( 8 )  " "     "*"    "*" "*"    " "
## 3  ( 9 )  "*"     "*"    " " "*"    " "
## 3  ( 10 ) "*"     " "    "*" "*"    " "
## 4  ( 1 )  "*"     "*"    " " "*"    "*"
## 4  ( 2 )  " "     "*"    "*" "*"    "*"
## 4  ( 3 )  "*"     "*"    "*" " "    "*"
## 4  ( 4 )  "*"     " "    "*" "*"    "*"
## 4  ( 5 )  "*"     "*"    "*" "*"    " "
## 5  ( 1 )  "*"     "*"    "*" "*"    "*"
```

```
regout=summary(regsubsets.out)
cbind(regout$which,regout$adjr2)
```

```
##   (Intercept) xWeight xStern xHb xTandT xAdult
## 1           1       0      0   0      0      1  0.719276160
## 1           1       0      1   0      0      0  0.123165029
## 1           1       1      0   0      0      0  0.056684177
## 1           1       0      0   1      0      0  0.009467051
## 1           1       0      0   0      1      0  0.006854716
## 2           1       0      1   0      0      1  0.769352752
## 2           1       1      0   0      0      1  0.752789859
## 2           1       0      0   1      0      1  0.749672430
```

```
## 2           1       0       0   0       1       1  0.742971481
## 2           1       0       1   1       0       0  0.108284065
## 2           1       1       1   0       0       0  0.104101481
## 2           1       0       1   0       1       0  0.101244516
## 2           1       1       0   1       0       0  0.041622671
## 2           1       1       0   0       1       0  0.039812952
## 2           1       0       0   1       1       0 -0.012451267
## 3           1       0       1   0       1       1  0.768718531
## 3           1       1       1   0       0       1  0.767814667
## 3           1       0       1   1       0       1  0.765867423
## 3           1       1       0   1       0       1  0.748700566
## 3           1       1       0   0       1       1  0.747062242
## 3           1       0       0   1       1       1  0.744766795
## 3           1       1       1   1       0       0  0.111442363
## 3           1       0       1   1       1       0  0.092809924
## 3           1       1       1   0       1       0  0.084586633
## 3           1       1       0   1       1       0  0.019769759
## 4           1       1       1   0       1       1  0.763003270
## 4           1       0       1   1       1       1  0.762645170
## 4           1       1       1   1       0       1  0.761713557
## 4           1       1       0   1       1       1  0.742159605
## 4           1       1       1   1       1       0  0.088260526
## 5           1       1       1   1       1       1  0.756755374
```

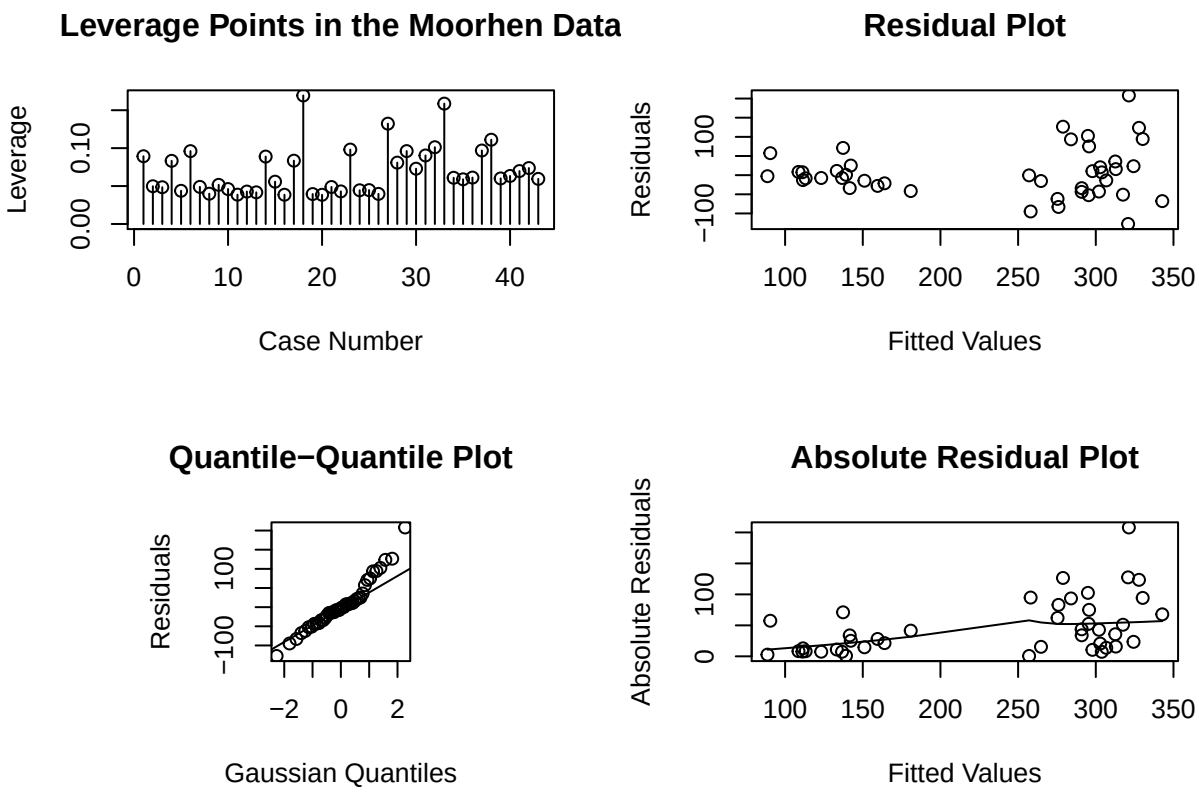
```
## Visualise the selection result

plt=subsets(regsubsets.out, statistic="adjr2", main = "Adjusted R^2",ylim=c(0.716,0.769),legend=FALSE)
```



Based on adjusted R^2, the 2-variable model using stern and adult is the best model.

- Step 6: fit the selected model and check its diagnositcs.

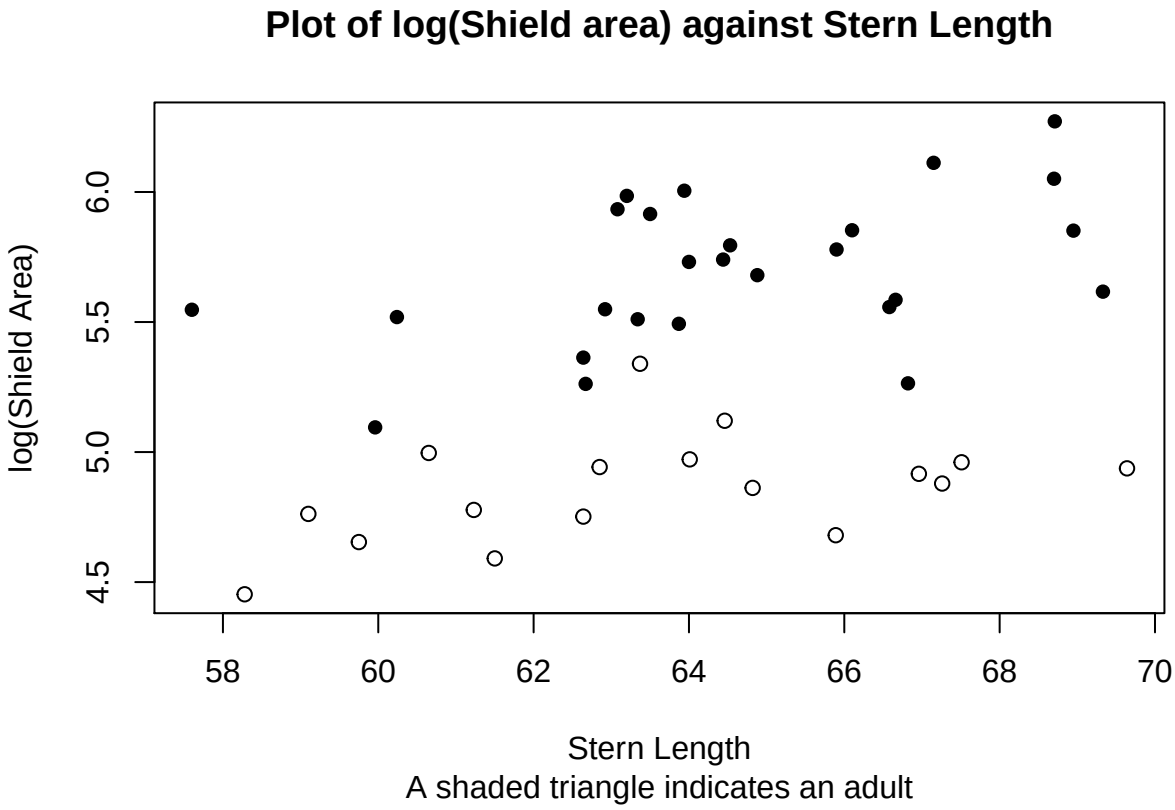
```
par(mfcol=c(2,2),oma = c(0,0,0,0))

plot(c(1:43),hat(x[,c(2,5)]),ylim=c(0,max(hat(x[,c(2,5)]))),
     main="Leverage Points in the Moorhen Data",
     xlab="Case Number",
     ylab="Leverage")
segments(c(1:43),0,c(1:43),hat(x[,c(2,5)]))
abline(h=10/43) #2p/n

par(pty="s")
qqnorm(residuals(fit),
       main="Quantile-Quantile Plot",
       xlab="Gaussian Quantiles",
       ylab="Residuals")
qqline(residuals(fit))
par(pty="m")

plot(fitted(fit),residuals(fit),
     main="Residual Plot",
     xlab="Fitted Values",
     ylab="Residuals")

plot(fitted(fit),abs(residuals(fit)),
     main="Absolute Residual Plot",
     xlab="Fitted Values",
     ylab="Absolute Residuals")
lines(lowess(fitted(fit),abs(residuals(fit))))

mtext("Diagnostics for the model fit to the Moorhen data",
      side=3,line=2,outer=T,cex=1.5)
mtext("The model includes Stern and Adult with shield area on the log scale",
      side=1,line=2,outer=T)
```



The normality, independence and homoscedasticity assumptions of the model are reasonably satisfied.

Importantly, since we have two predictors and one predictor is categorical, we can include an interaction term and basically represent the model by parallel but separate regression lines on each factor level.

```
par(mfrow=c(1,1),oma=c(0,0,0,0))
plot(x[,2],log(y),type="n",
     main="Plot of log(Shield area) against Stern Length",
     xlab="Stern Length",
     ylab="log(Shield Area)",
     sub="A shaded triangle indicates an adult")
adult = (x[,5]==1)
points(x[!adult,2],log(y[!adult]),pch=1)
points(x[adult,2],log(y[adult]),pch=16)
abline(fit$coef[1],fit$coef[2]) # adult == 0 or 1
abline(fit$coef[1]+fit$coef[3],fit$coef[2])
```




```
fit$coefficients
```

```
##   (Intercept)       xWeight        xStern           xHb        xTandT
## -444.52636041   -0.03097264    7.28804590   -2.23331260    1.92779286
##        xAdult
##  160.56481768
```

Finally, after checking model assumptions and visualise the outcomes, we write the final model form:

$$log(ShieldArea) = 2.45 + 0.038 * Stern + 0.795 * I(adult)$$

(0.77) (0.012) (0.074)

We need to backtransform, which gives,

$$ShieldArea = exp(2.45)exp(0.038 * Stern)exp(0.795 * I(adult))$$

Similar multiple regression analysis can follow this combination of visualisation and model-based exploration to refine and diagnose the models.