Module 4 - Diagnostics of multiple regression model

Pawel Chilinski

November 19, 2013

Exercise 1.

Create a vector x of n equally distributed numbers on interval [0, 1]. Generate random numbers $Y_i = 2 + 15x_i + \epsilon_i$, $i = 1, ..., n$. where $\epsilon_i \sim N(0, 2)$ - iid. Fit a linear regression model to the data $(x_i, Y_i)_{i=1}^n$.

Function to generate data, residual plots and qq-plots for given n:

```
> generate.data<-function(n,errors=function(n){rnorm(n, 0, sqrt(2))},y_fun=function(X,E){2 + 15*X + E}){
+         X <- seq(from=0,to=1,length.out=n)
+         E <- errors(n)
+         Y <- y_fun(X,E)
+         model <- lm(Y~X)
+         return(list(n=n,X=X,E=E,Y=Y,model=model))
+ }
> residual.plots<-function(data){
+         par(mfrow=c(1,3))
+         for(d in data){
+                 plot(d$model$fitted,resid(d$model),xlab="fitted",ylab="residuals",main=paste("For",d$n))
+                 abline(h=0,lwd=0.5)
+         }
+         par(mfrow=c(1,1))
+ }
> qq.plots<-function(data){
+         par(mfrow=c(1,3))
+         for(d in data){
+                 qqnorm(resid(d$model))
+                 qqline(resid(d$model))
+         }
+         par(mfrow=c(1,1))
+ }
```

- Make residual plots for n = 30, 100, 300.

```
> data<-lapply(c(30,100,300),generate.data)
> residual.plots(data)
```

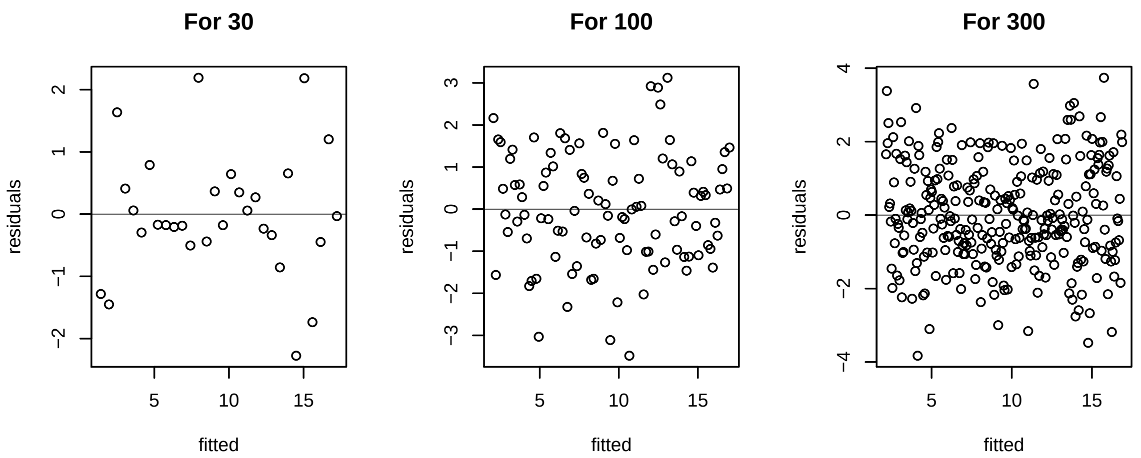

Figure 1: Residual plots for n=30, 100, 300.

- Make normal QQ plots for n = 30, 100, 300.

```
> qq.plots(data)
```

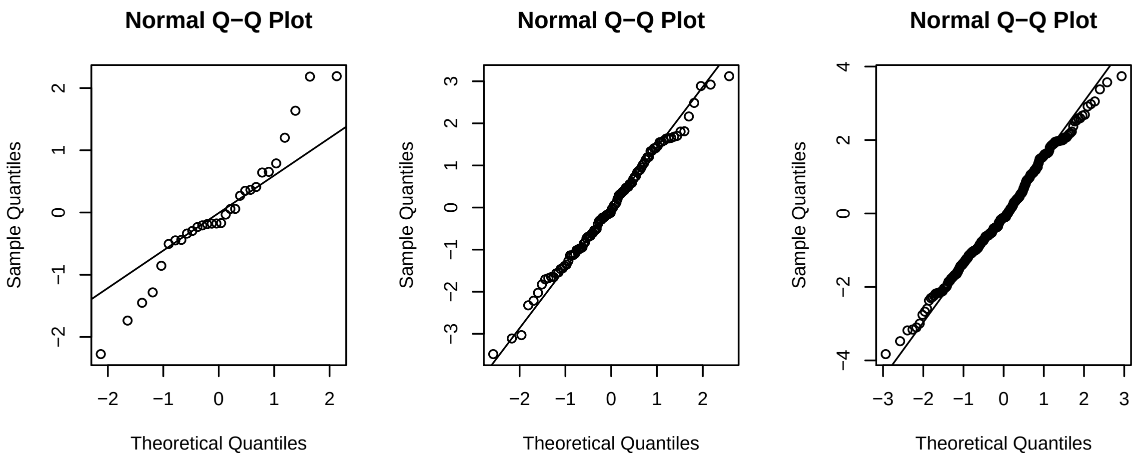

Figure 2: Normal QQ plots for n = 30, 100, 300

- Generate new $(Y_i)_{i=1}^n$ changing the distribution of errors to centered gamma with parameters 2 and 2.Make residual and QQ plots for n = 30, 100, 300.

```
> data<-lapply(c(30,100,300),generate.data,errors=function(n){rgamma(n,2,2)})
> residual.plots(data)
```

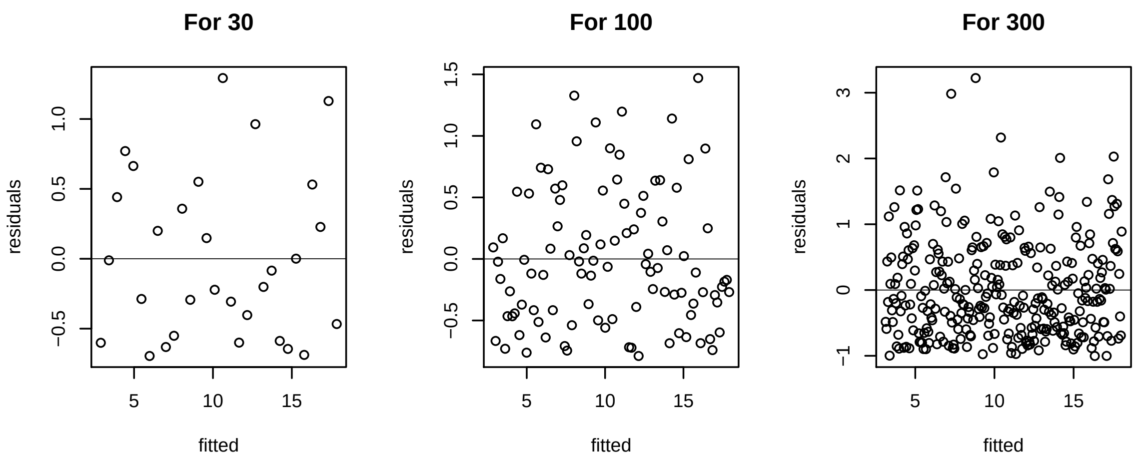

Figure 3: Residual plots for n=30, 100, 300.

```
> qq.plots(data)
```

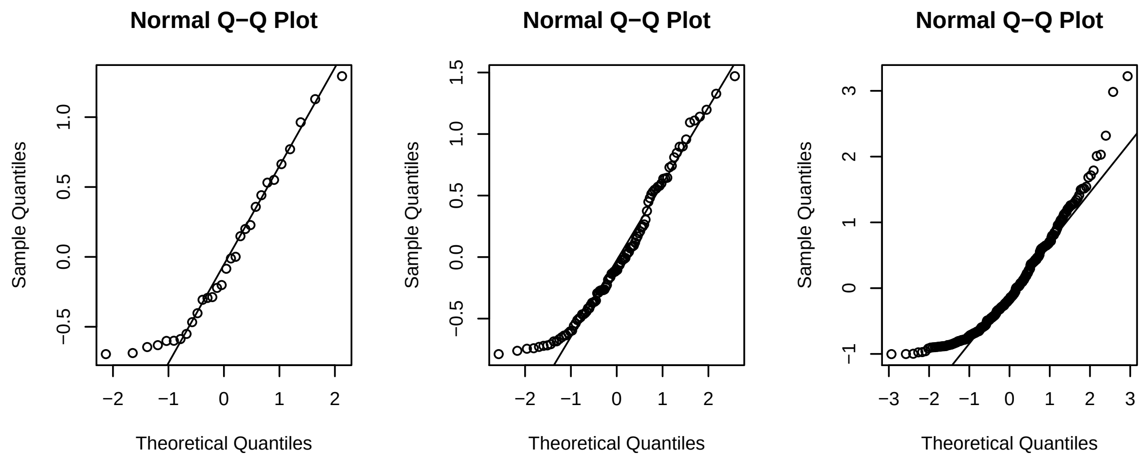

Figure 4: Normal QQ plots for n = 30, 100, 300

- Generate new $(Y_i)_{i=1}^n$ changing the distribution of errors to Cauchy with parameters 0 and 1. Make residual and QQ plots for n = 30, 100, 300.

```
> data<-lapply(c(30,100,300),generate.data,errors=function(n){rcauchy(n,0,1)})
> residual.plots(data)
```

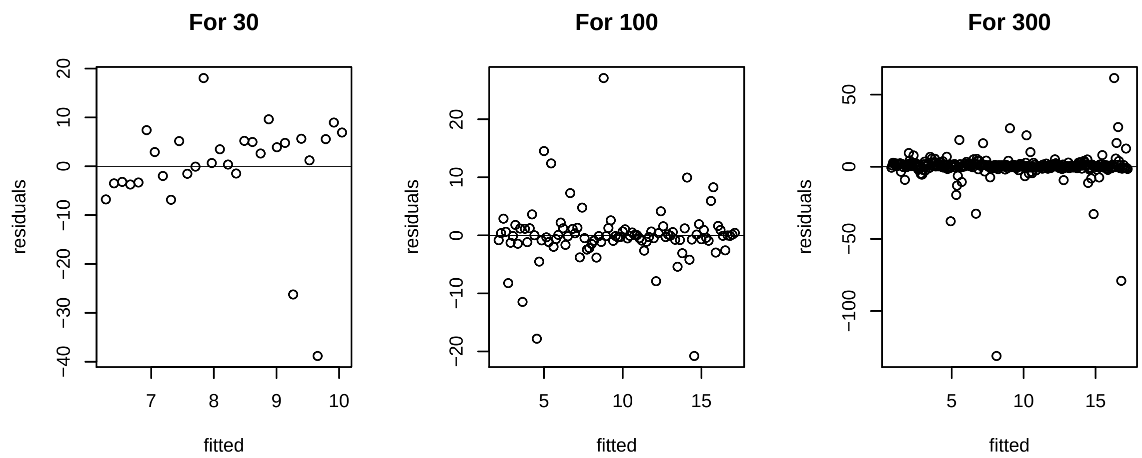

Figure 5: Residual plots for n=30, 100, 300.

```
> qq.plots(data)
```

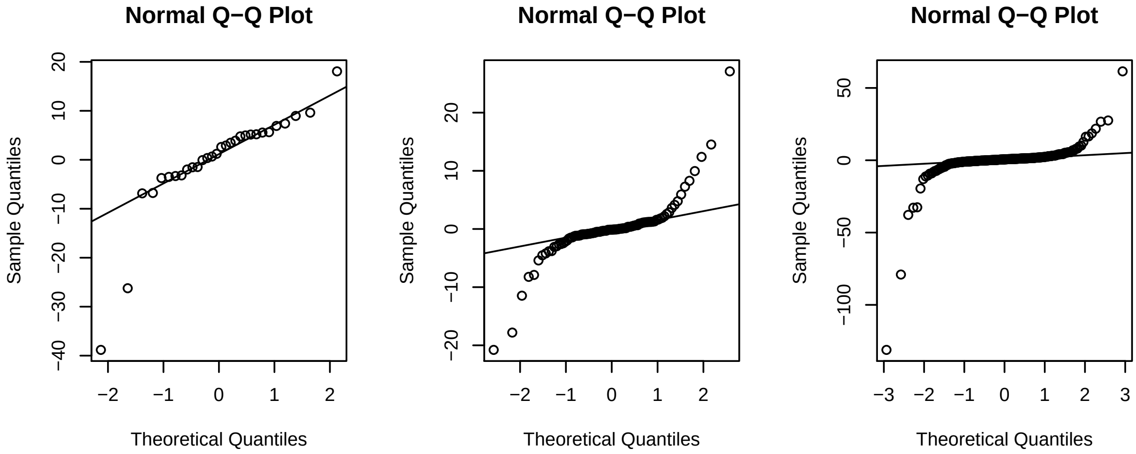

Figure 6: Normal QQ plots for n = 30, 100, 300

- Generate new $(Y_i)_{i=1}^n$ where $Y_i = 2 + 15x_i^2 + \epsilon_i$ and $\epsilon_i \sim N(0, 2)$. Make residual and QQ plots for n = 30, 100, 300.

```
> data<-lapply(c(30,100,300),generate.data,y_fun=function(X,E){2 + 15*X^2 + E})
> residual.plots(data)
```

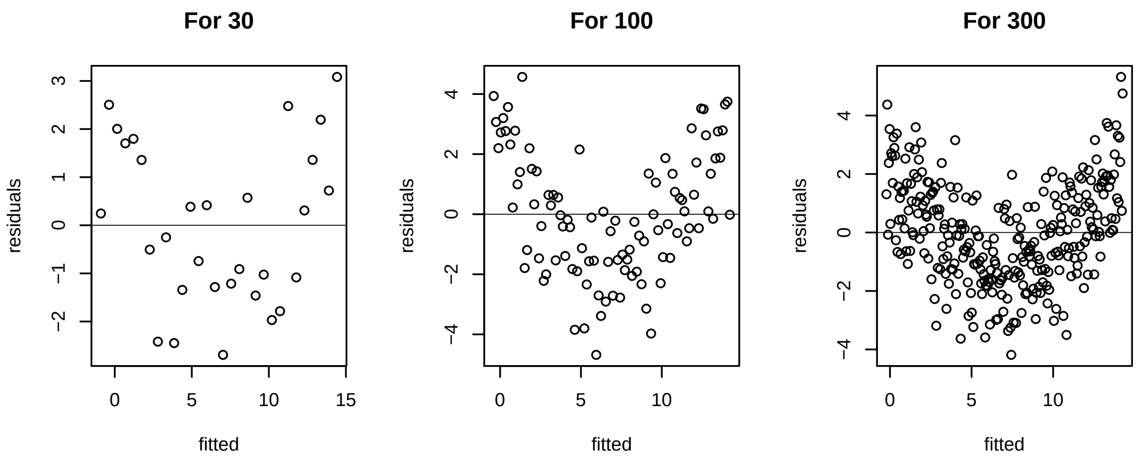

Figure 7: Residual plots for n=30, 100, 300.

```
> qq.plots(data)
```

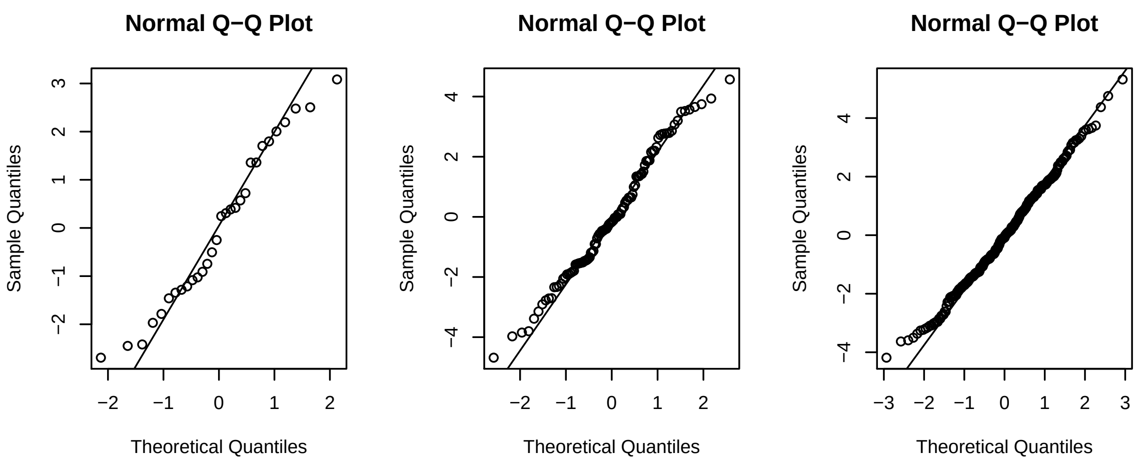

Figure 8: Normal QQ plots for n = 30, 100, 300

When our simulated errors are normal and structural equation is correct we cannot see any anomalies on residual and qq plots. But after changing distribution of errors to gamma(right skewness) and Cauchy (fat tails) we can see changes in the residual and qq plots. After changing equation generating data to quadratic formula we can see the quadratic pattern emerging from the residual plots.

Exercise 2.

Load data trees.

- Fit simple linear regression models to two pairs of variables Volume∼Girth and Volume ∼ Height.

```
> vol_girth.model <- lm(Volume~Girth,trees)
> vol_height.model <- lm(Volume~Height,trees)
```

- For the both considered models analyse residual plots:

```
> par(mfrow=c(1,2))
> plot(resid(vol_girth.model),xlab="i",ylab="residuals",main="Volume~Girth")
> abline(h=0,lwd=0.5)
> plot(resid(vol_height.model),xlab="i",ylab="residuals",main="Volume~Height")
> abline(h=0,lwd=0.5)
> par(mfrow=c(1,1))
```

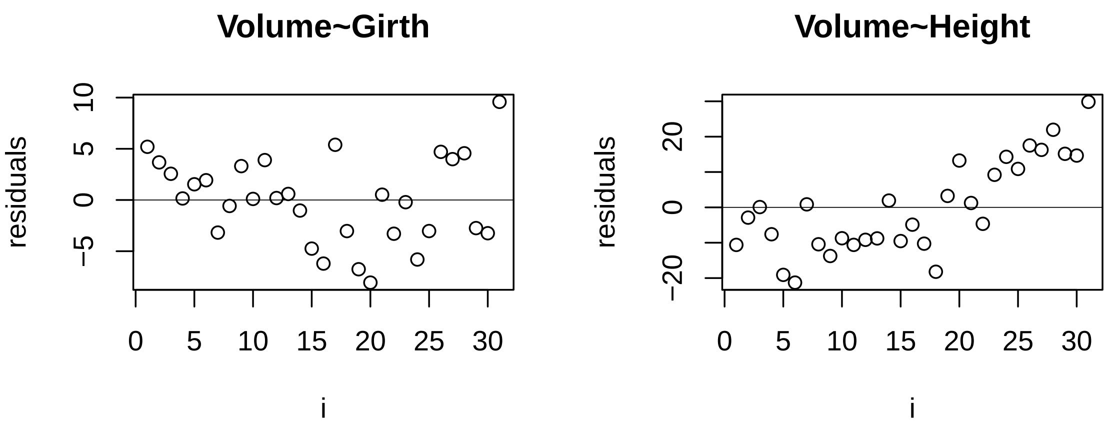

Figure 9: Residuals versus index $(i, e_i)_{i=1}^{n}$

```
> par(mfrow=c(1,2))
> plot(vol_girth.model$model$Girth,resid(vol_girth.model),xlab="Girth",ylab="residuals",
+                 main="Volume~Girth")
> abline(h=0,lwd=0.5)
> plot(vol_height.model$model$Height,resid(vol_height.model),xlab="Height",ylab="residuals",
+                 main="Volume~Height")
> abline(h=0,lwd=0.5)
> par(mfrow=c(1,1))
```

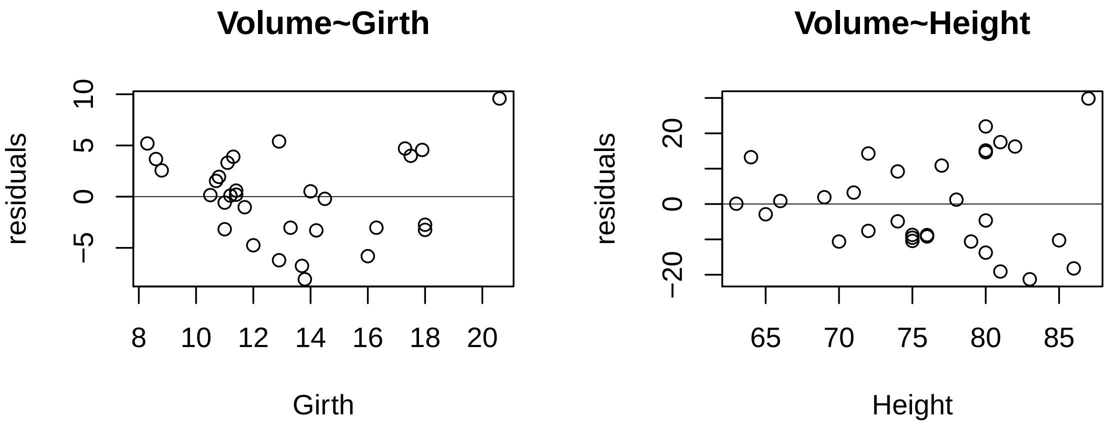

Figure 10: Residuals versus explanatory variable $(x_i, e_i)_{i=1}^n$

```
> par(mfrow=c(1,2))
> plot(vol_girth.model$fitted,resid(vol_girth.model),xlab="Fitted volume",ylab="residuals",
+                 main="Volume~Girth")
> abline(h=0,lwd=0.5)
> plot(vol_height.model$fitted,resid(vol_height.model),xlab="Fitted volume",ylab="residuals",
+                 main="Volume~Height")
> abline(h=0,lwd=0.5)
> par(mfrow=c(1,1))
```

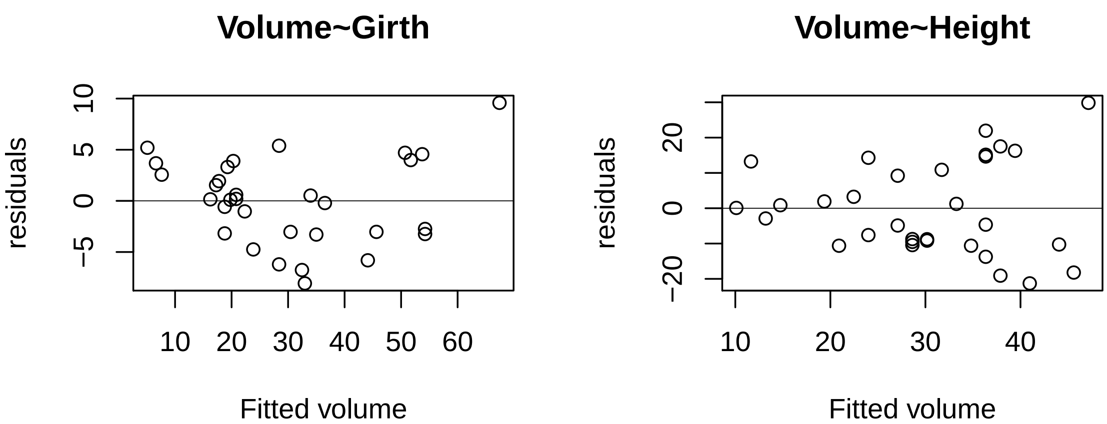

Figure 11: Residuals versus predicted values $(\hat{Y}_i, e_i)_{i=1}^n$

- On the basis of residual plots propose a nonlinear model describing relationship between variables Volume and Girth. Fit the new model and compare it with the linear model. In particular compare the estimated variances of volume (σ^2)

  From the residual plot residuals versus explanatory variable we can see the quadratic dependence of residual from the explanatory variable. So new model:

  ```
  > vol_girth_quad.model <- lm(Volume~Girth+I(Girth*Girth),trees)
  ```

  On residual plot for a new model we can see that quadratic dependence of residuals on Girth has disappeared:

```
> par(mfrow=c(1,2))
> plot(vol_girth_quad.model$model$Girth,resid(vol_girth_quad.model),xlab="Girth",ylab="residuals",
+                 main="Volume~Girth+I(Girth*Girth)")
> abline(h=0,lwd=0.5)
> par(mfrow=c(1,1))
```

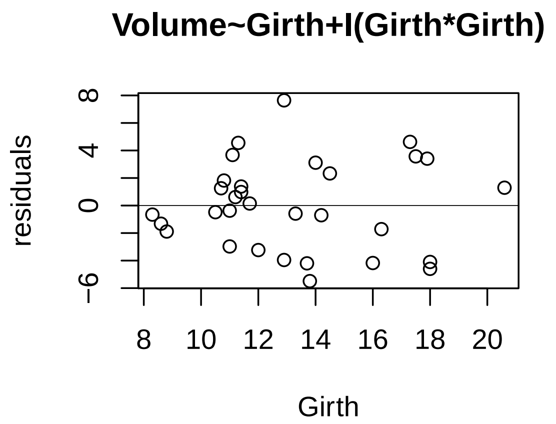

Figure 12: Residuals versus explanatory variable $(x_i, e_i)_{i=1}^n$ for quadratic model

Comparing two models we can see that R^2 for the quadratic model is bigger, estimated variances of volume is smaller in case of complex model (Residual standard error squared) and RSS is significantly smaller for complex model:

```
> summary(vol_girth.model)

Call:
lm(formula = Volume ~ Girth, data = trees)

Residuals:
   Min     1Q Median     3Q    Max 
-8.065 -3.107  0.152  3.495  9.587 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -36.9435     3.3651  -10.98 7.62e-12 ***
Girth         5.0659     0.2474   20.48  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.252 on 29 degrees of freedom
Multiple R-squared:  0.9353,	Adjusted R-squared:  0.9331 
F-statistic: 419.4 on 1 and 29 DF,  p-value: < 2.2e-16

> summary(vol_girth_quad.model)

Call:
lm(formula = Volume ~ Girth + I(Girth * Girth), data = trees)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.4889 -2.4293 -0.3718  2.0764  7.6447 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)      10.78627   11.22282   0.961 0.344728    
Girth            -2.09214    1.64734  -1.270 0.214534    
I(Girth * Girth)  0.25454    0.05817   4.376 0.000152 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.335 on 28 degrees of freedom
Multiple R-squared:  0.9616,	Adjusted R-squared:  0.9588 
F-statistic: 350.5 on 2 and 28 DF,  p-value: < 2.2e-16
```

```
> anova(vol_girth.model,vol_girth_quad.model)

Analysis of Variance Table

Model 1: Volume ~ Girth
Model 2: Volume ~ Girth + I(Girth * Girth)
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1     29 524.30
2     28 311.38  1    212.92 19.146 0.0001524 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Exercise 3.

File realest.txt contains data related to houses in Chicago. Fit a linear regression model taking price of house as a response variable and the rest of variables in the data set as explanatory variables.

```
> realest.data <- read.table(file="realest.txt",header=T)
> price_all.model <- lm(Price~.,realest.data)
```

- Analyse diagnostic plots of the model. Use function plot(m, which=1:4) where m is the fitted model returned by function lm() to obtain residual plot, QQ-plot and other diagnostic plots.

```
> op <- par(mfrow=c(2,2),mar = par("mar")/2)
> plot(price_all.model, which=1:4)
> par(op)
```

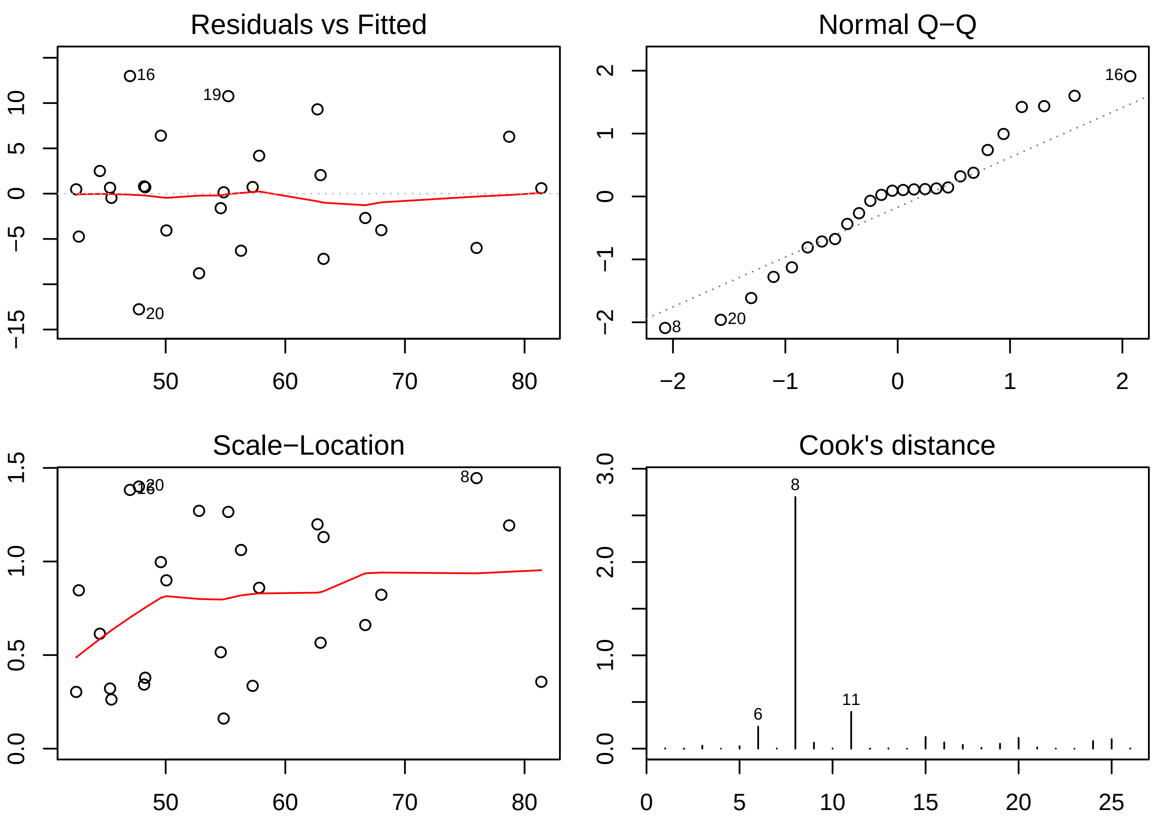

Figure 13: Diagnostic plots for Price~ .

  - Residuals vs Fitted - There is a pattern of decreasing variance of errors with increase in fitted value. It looks like residuals are negatively skewed.
  - Scale-Location - There is a little bit if upward trend which isn't good news for error normal assumption.
  - Normal Q-Q - there are two modes in the residuals.
  - Cook's distance - shows that three observations i.e. 6th, 8th and 11th are influential and should be examined for correctness.

- Are there any outliers in the data?

To find outliers I use heuristic rule that observations with absolute value studentized residual ≥ 2 are good candidates for outliers:

```
> library(MASS)
> price_all.model.studres <- studres(price_all.model)
> plot(price_all.model.studres,ylab="studentized residual",xlab="i")
> outliers <- which(abs(price_all.model.studres)>=2)
> points(outliers, price_all.model.studres[outliers],col="red",pch=16)
> text(outliers+1,price_all.model.studres[outliers],labels=outliers)
> legend(x="topleft",legend=c("outlier"),pch=c(16),col=c("red"))
```

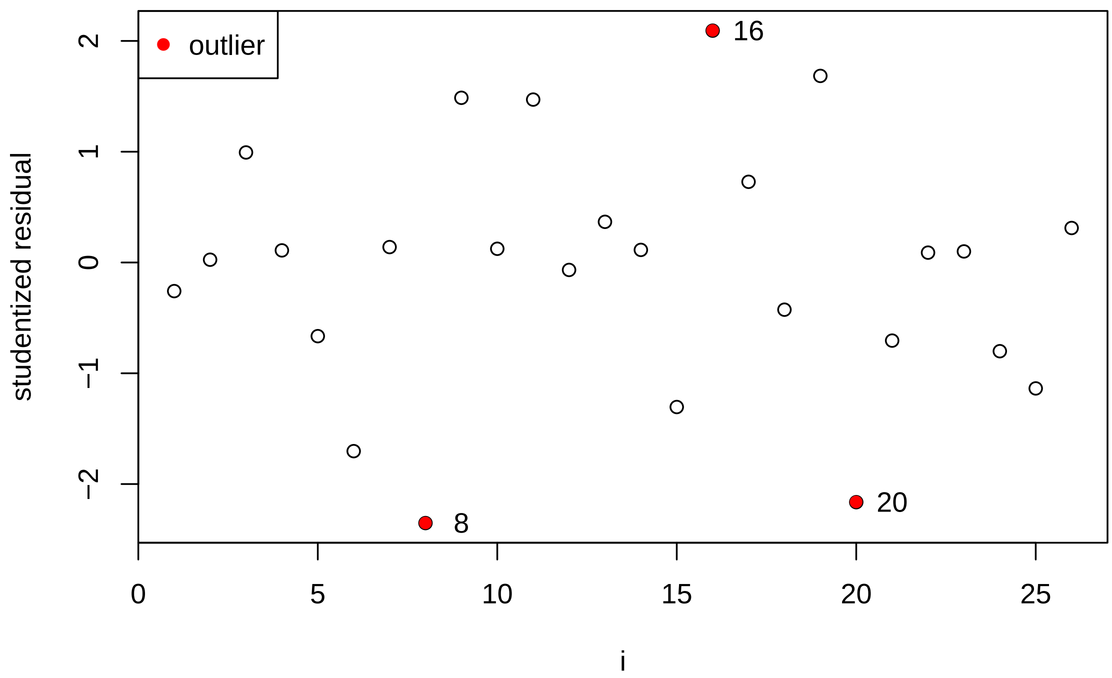

Figure 14: Outliers

- Identify influence observations in the data (use Cook's distance and hatvalues, functions: cooks.distance() and hatvalues()).

  Using heuristic rule that observation is potentially influential if $h_{ii} \geq \frac{2p}{n}$ we can identify them:

```
> as.vector(which(hatvalues(price_all.model)>= length(price_all.model$coefficients)/nrow(realest.data)))
```

```
 [1]  2  6  7  8 10 11 15 17 22 24 25
```

  Observations 6, 8 and 11 are also pointed out by the Cook's distance plot.
  Only 8th observation has Cook's distance greater than 1 so can be qualified as influential:

```
> as.vector(which(cooks.distance(price_all.model)>1))
```

```
[1] 8
```

  So we can see that 8 is influential and outlier. The 20th and 16th observations are outliers but are not influential.

Exercise 4.

File activity.txt contains data describing effectiveness of work done during 1 hour (variable Y) and two possibly related to it variables (X1 and X2).

```
> activity.data <- read.table(file="activity.txt",header=T)
```

- Fit a linear regression model for variable Y versus X1 and X2.

```
> activity.model <- lm(Y~.,activity.data)
> summary(activity.model)
```

```
Call:
lm(formula = Y ~ ., data = activity.data)

Residuals:
    Min      1Q  Median      3Q     Max
-328.43  -77.73   44.76  149.20  212.69

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 5.123e+02  2.231e+02   2.296  0.05533 .
X1          3.875e-02  4.681e-03   8.278 7.33e-05 ***
X2          5.894e-02  1.640e-02   3.594  0.00881 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 216.7 on 7 degrees of freedom
Multiple R-squared:  0.9468,	Adjusted R-squared:  0.9316
F-statistic:  62.3 on 2 and 7 DF,  p-value: 3.47e-05
```

- Assess the diagnosis plots for the fitted model.

```
> op <- par(mfrow=c(2,2),mar = par("mar")/2)
> plot(activity.model, which=1:4)
> par(op)
```

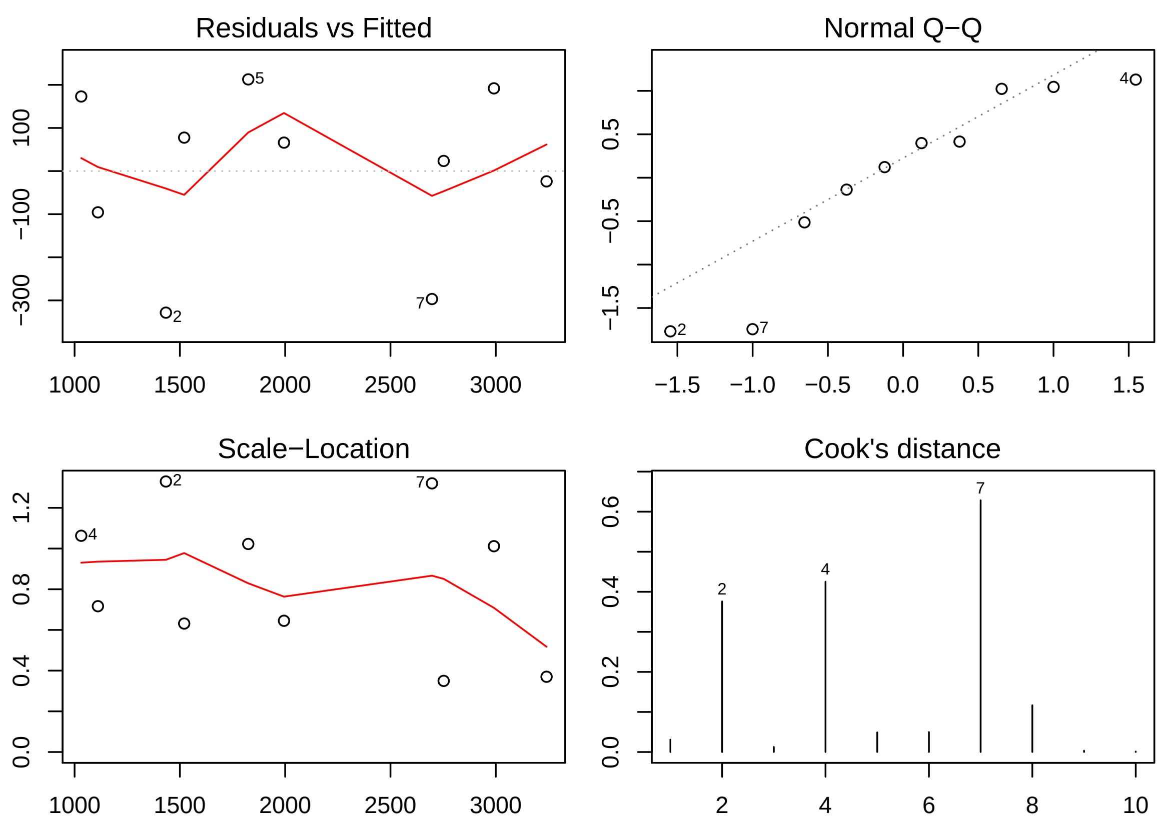

Figure 15: Diagnostic plots for Y∼ .

There is visible zigzag pattern for residuals. None of the observations has Cook's distance greater than 1 so there are no candidates for influential variables. The qqplot shows divergence from the normality but we have too few observations to conclude.

- Make partial regression plots and partial residual plots for both explanatory variables (use function prplot() in a library named faraway).

```
> library(car)
> op <- par(mfrow=c(1,2),mar = par("mar")/2)
> avPlots(activity.model)
> par(op)
```

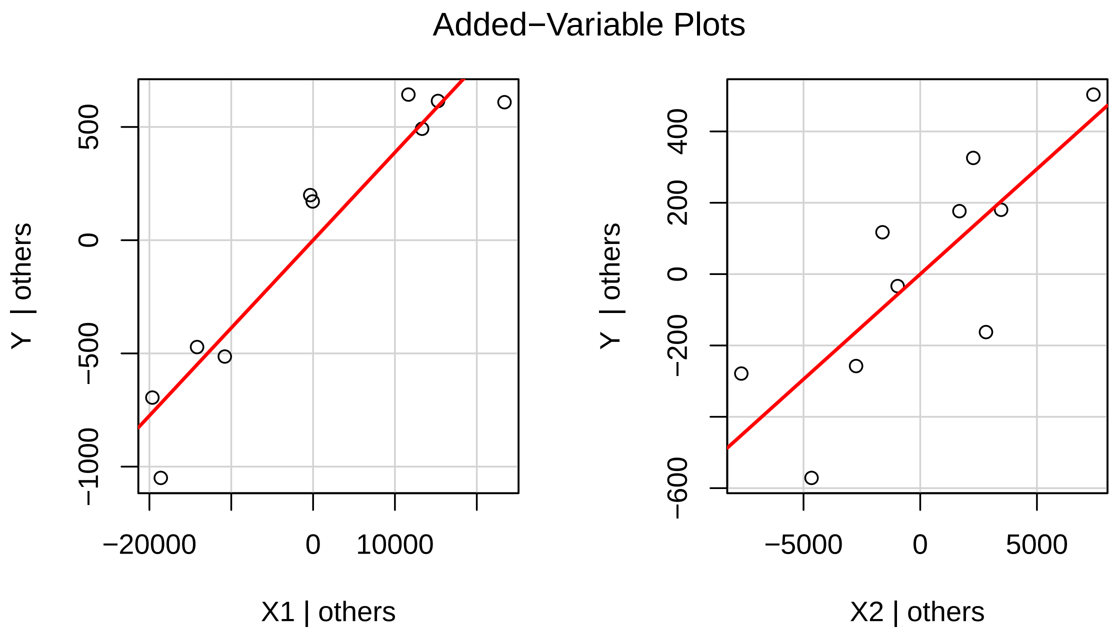

Figure 16: Partial regression plots

We can see linear dependence and no visible outliers and influential observations for each predictor variables taking into consideration another predictor.

```
> library(faraway)
> op <- par(mfrow=c(1,2),mar = par("mar")/2)
> prplot(activity.model,1)
> prplot(activity.model,2)
> par(op)
```

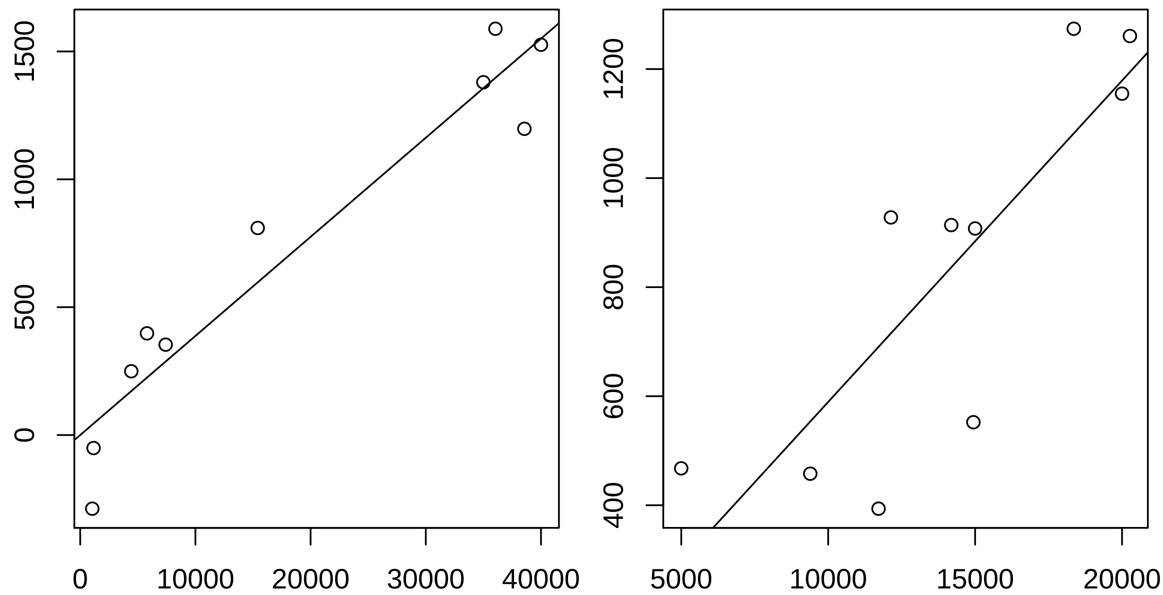

Figure 17: Partial residual plots

We can also use prettier plots from the car package:

```
> crPlots(activity.model)
```

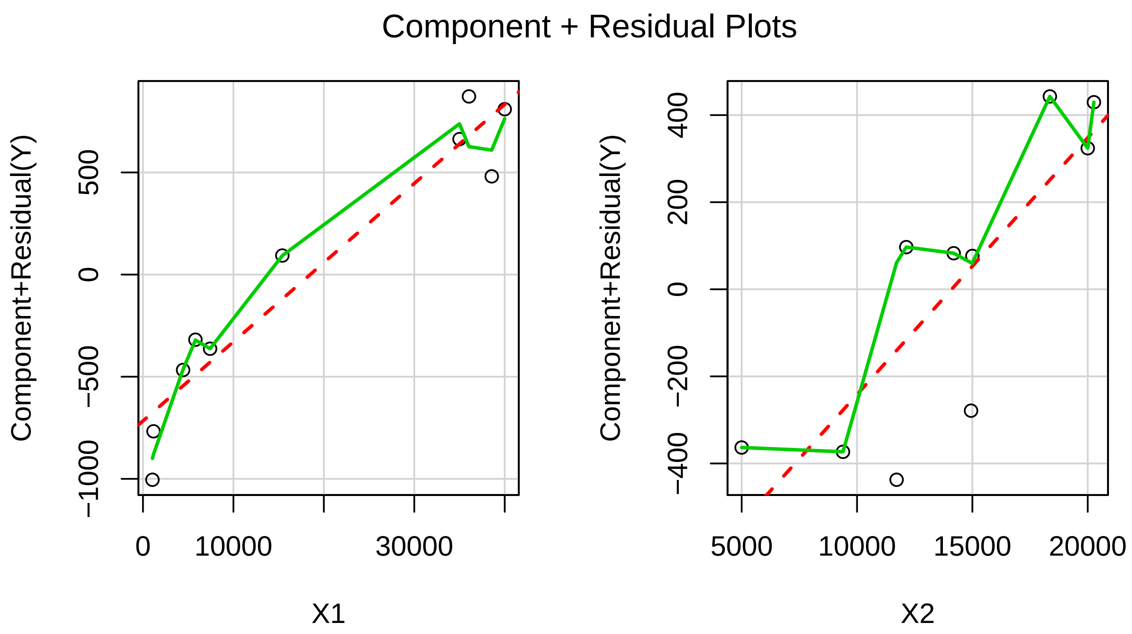

Figure 18: Component+Residual plots

We can can see that there could be discerned 2 groups in the data plots (2 for X1 and 2 for X2). So possibly we should create separate models for these 2 groups. Also we could notice that X_1 variable shows positive concave pattern.

- Propose a transformation of variable X1 on the basis of these plots. Compare values of R2 in the initial and proposed models. Because on the partial residual plot we can see positive concave relationship between remaining information in Y without predicted part from X_2 on X_1 variable we can try $log(X_1)$ transformation:

```
> activity_transformed.model <- lm(Y~log(X1)+X2,activity.data)
> summary(activity.model)$r.squared

[1] 0.9468106

> summary(activity_transformed.model)$r.squared

[1] 0.9587212
```

which has better R^2.

Exercise 5.

File strongx.txt contains results of an experiment in particle physics. Variables in the data set are:
crossx - cross-section of a particle,
energy - an inverse of an energy of a particle,
momentum - momentum of a particle,
sd - estimated standard deviation of crossx for a given value of momentum.
For each value of momentum an experiment was performed repeatedly. Thus an estimator of standard deviation of cross-section could be calculated for each value of momentum.
It is expected that cross-section should be a linear function of the inverse of energy of a particle.

```
> strongx.data <- read.table(file="strongx.txt",header=T)
```

- Fit a regression line describing dependence $crossx \sim energy$ using the least squares method.

```
> strongx.ls.model <- lm(crossx~energy, strongx.data)
> summary(strongx.ls.model)

Call:
lm(formula = crossx ~ energy, data = strongx.data)

Residuals:
    Min      1Q  Median      3Q     Max
-14.773  -9.319  -2.829   5.571  19.817
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   135.00      10.08    13.4 9.21e-07 ***
energy        619.71      47.68    13.0 1.16e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 12.69 on 8 degrees of freedom
Multiple R-squared:  0.9548,	Adjusted R-squared:  0.9491 
F-statistic: 168.9 on 1 and 8 DF,  p-value: 1.165e-06
```

Diagnostic plots:

```
> op <- par(mfrow=c(2,2),mar = par("mar")/2)
> plot(strongx.ls.model, which=1:4)
> par(op)
```

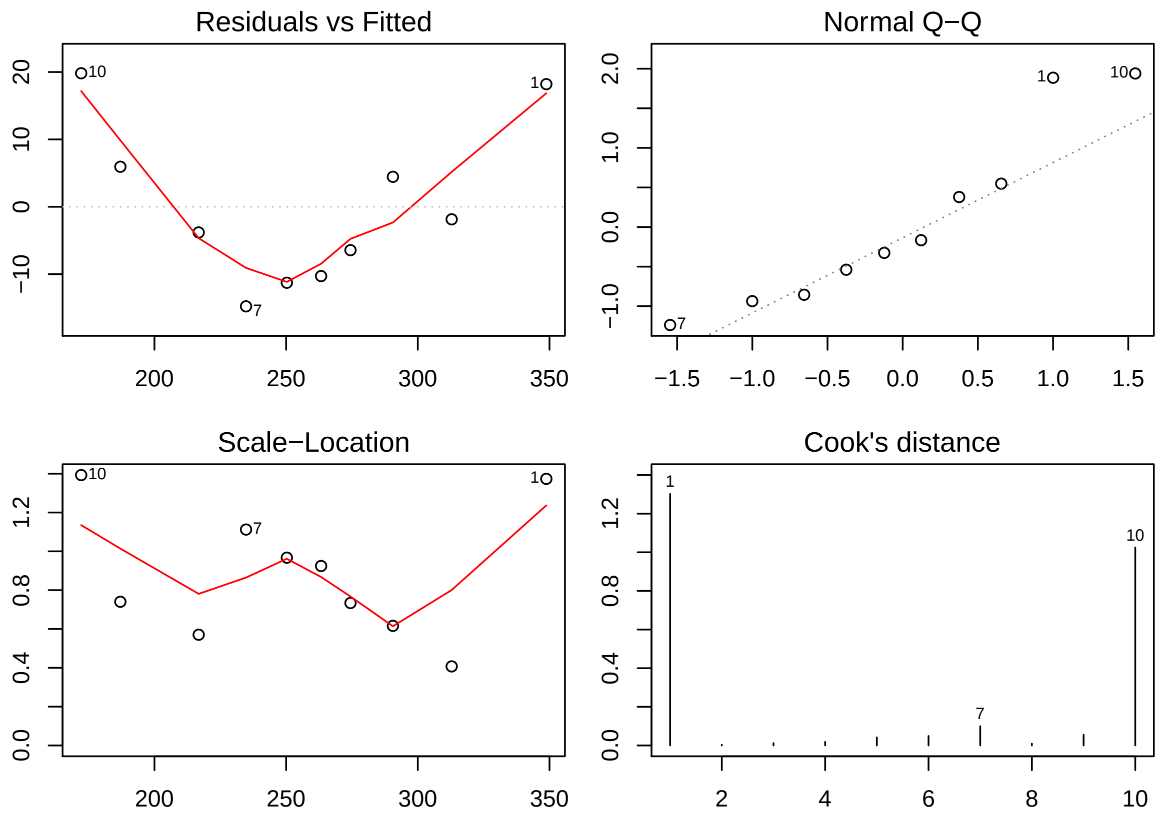

Figure 19: Diagnostic plots for LSM

Regression line and data points:

```
> plot(strongx.data$energy,strongx.data$crossx, xlab="energy",ylab="crossx",main="crossx~energy")
> abline(strongx.ls.model,col="blue")
> legend(x="topleft",col=c("black","blue"),pch=c(1,NA),legend=c("data","fitted line"),lty=c(0,1))
```

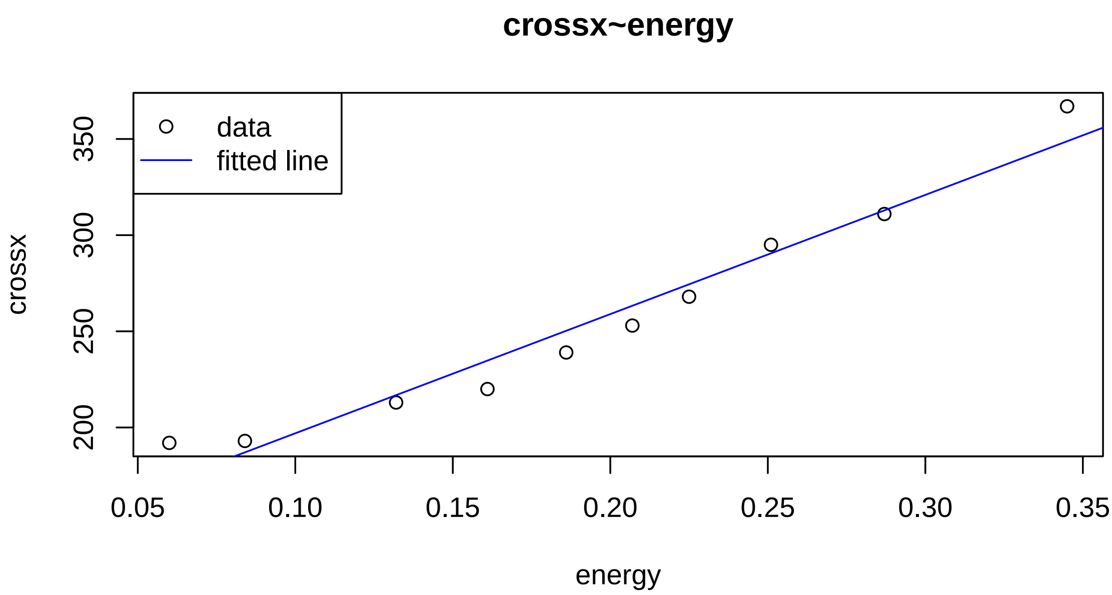

Figure 20: Data and fitted model

- Fit a regression line describing dependence *crossx* $\sim$ *energy* using the weighted least squares method (use parameter weights in function lm() and set it to sd^{-2}).

```
> strongx.wls.model <- lm(crossx~energy, strongx.data,weights=strongx.data$sd^-2)
> summary(strongx.wls.model)

Call:
lm(formula = crossx ~ energy, data = strongx.data, weights = strongx.data$sd^-2)

Weighted Residuals:
    Min      1Q  Median      3Q     Max 
-2.3230 -0.8842  0.0000  1.3900  2.3353 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  148.473      8.079   18.38 7.91e-08 ***
energy       530.835     47.550   11.16 3.71e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 

Residual standard error: 1.657 on 8 degrees of freedom
Multiple R-squared:  0.9397,	Adjusted R-squared:  0.9321 
F-statistic: 124.6 on 1 and 8 DF,  p-value: 3.71e-06
```

Diagnostic plots:

```
> op <- par(mfrow=c(2,2),mar = par("mar")/2)
> plot(strongx.wls.model, which=1:4)
> par(op)
```

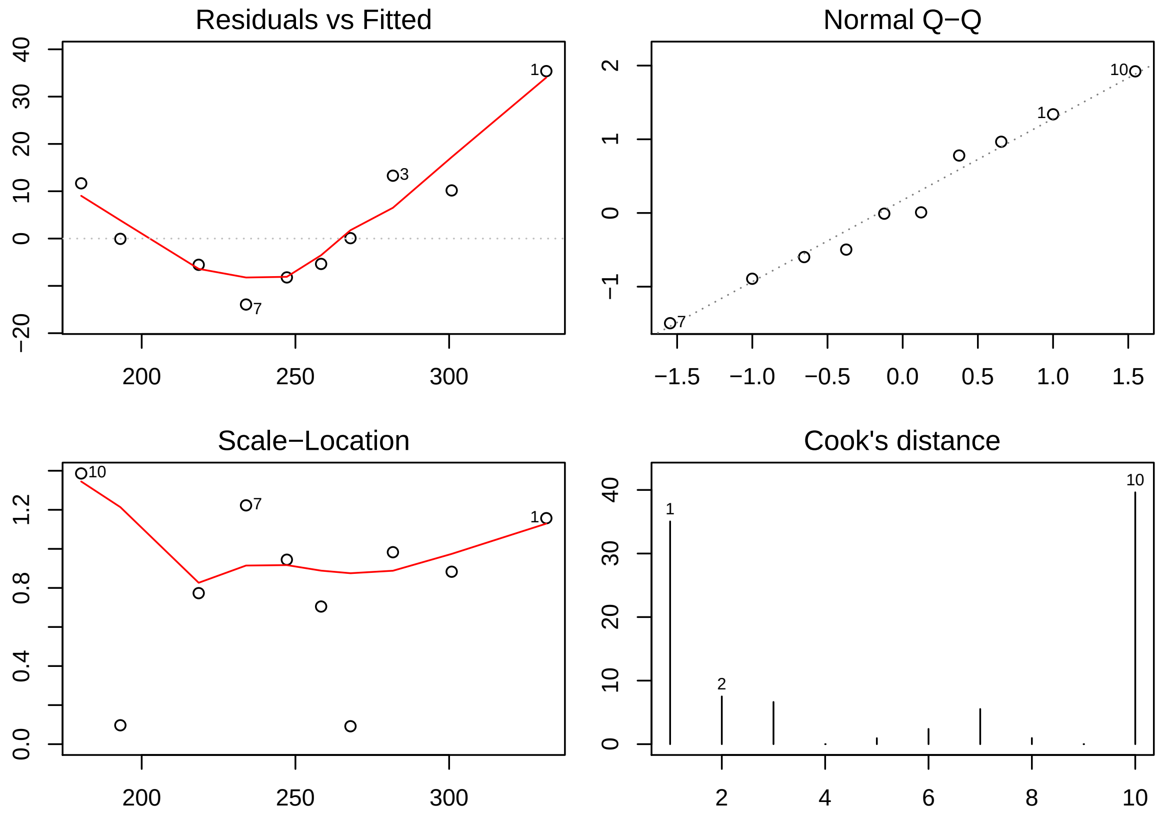

Figure 21: Diagnostic plots for WLSM

- Compare the two fitted models. Why is WLS line better fitted to observations having low energy of a particle?

```
> op <- par(mar = par("mar")/2)
> plot(strongx.data$energy,strongx.data$sd,xlab="energy",ylab="sd")
> par(op)
```

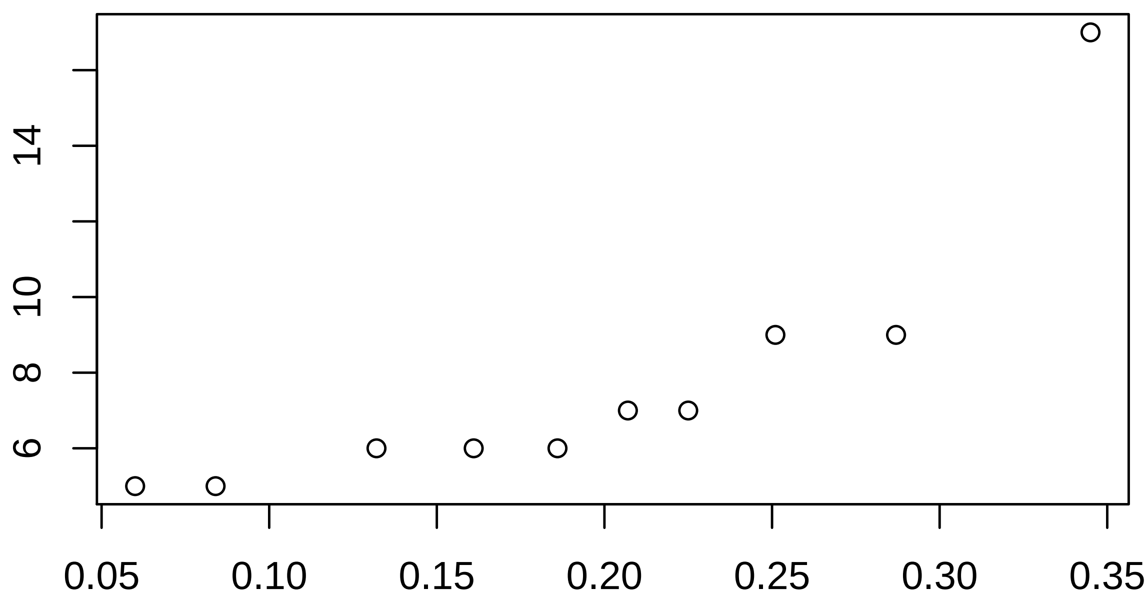

Figure 22: sd vs energy

We can see that for smaller values of energy the sd is smaller. The WLSM weights the squared errors by sd^{-2} giving this way more importance to observations with smaller sd. That's why smaller energy values' residuals are smaller in WLSM compared to LSM:

```
> plot(strongx.data$energy,resid(strongx.ls.model),xlab="energy",ylab="residuals",type="l",col="blue")
> lines(strongx.data$energy,resid(strongx.wls.model),col="red")
> legend(x="top",col=c("blue","red"),pch=c(NA,NA),legend=c("LSM","WLSM"),lty=c(1,1))
> abline(h=0,lwd=0.5)
```

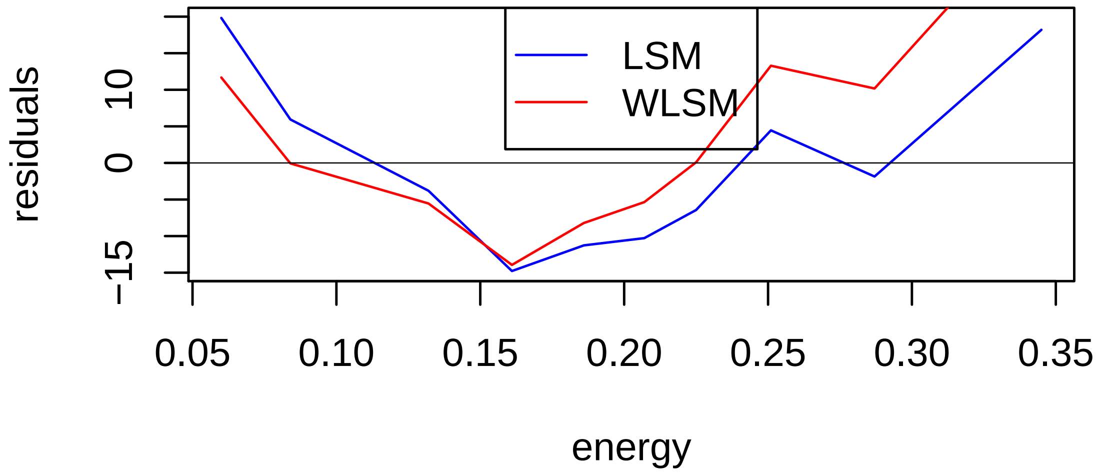

Figure 23: Residuals for LSM and WLSM

- On the basis of diagnostic plots for the WLS line propose a modification of this model. Because $Var(strongx|energy)$ is proportional to $E(strongx|energy)^2$ then we can apply transformation of $log(crossx)$.

```
> strongx.transformed.model <- lm(log(crossx)~energy, strongx.data,weights=strongx.data$sd^-2)
> summary(strongx.transformed.model)

Call:
lm(formula = log(crossx) ~ energy, data = strongx.data, weights = strongx.data$sd^-2)

Weighted Residuals:
       Min         1Q     Median         3Q        Max
-0.0078679 -0.0024135  0.0000592  0.0032524  0.0077494

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.08682    0.02394  212.48 2.69e-16 ***
energy       2.19888    0.14091   15.61 2.84e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.004909 on 8 degrees of freedom
Multiple R-squared:  0.9682,       Adjusted R-squared:  0.9642
F-statistic: 243.5 on 1 and 8 DF,  p-value: 2.835e-07
```

On the diagnostic plots we can see that residuals decreased considerably:

```
> op <- par(mfrow=c(2,2),mar = par("mar")/2)
> plot(strongx.transformed.model, which=1:4)
> par(op)
```

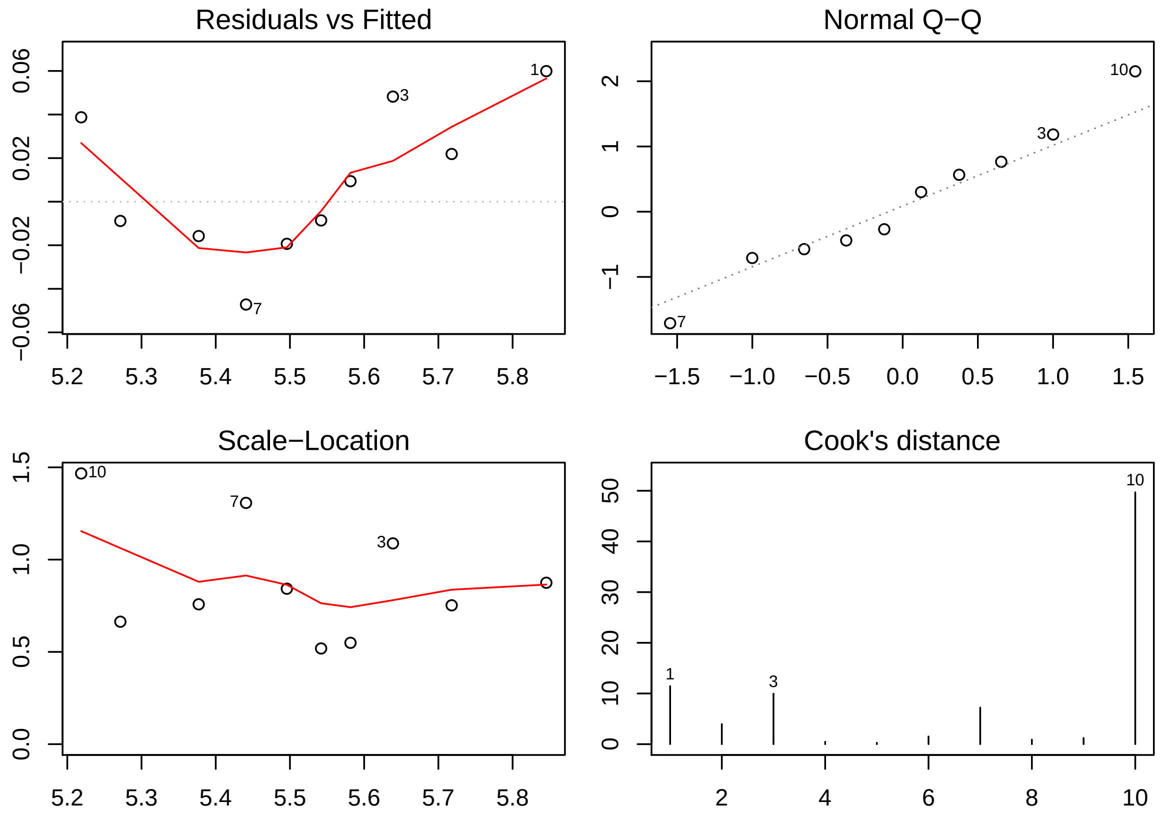

Figure 24: Diagnostic plots for transformed model

- Draw the fitted curve on the data scatterplot.

```
> plot(strongx.data$energy,strongx.data$crossx, xlab="energy",ylab="crossx",main="log(crossx)~energy")
> abline(strongx.ls.model,col="blue")
> curve(exp(strongx.transformed.model$coefficients[1]+strongx.transformed.model$coefficients[2]*x),col="green"
> legend(x="topleft",col=c("black","blue","green"),pch=c(1,NA,NA),legend=c("data","crossx~energy","log(crossx)
```

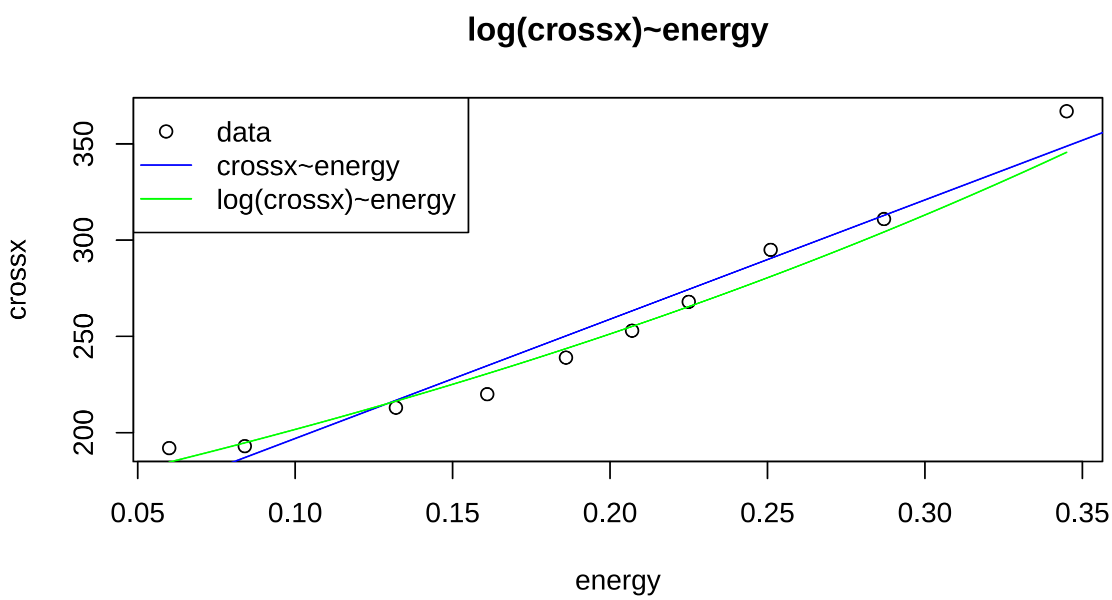

Figure 25: Data and fitted transformed model

Exercise 6.

File uscrime.txt the following data related to 47 states of USA: R - crime rate,
S - =1 (southern states), = 0 (other),
Age - number of men aged 14-24 among 1000 citizens,
Ex0, Ex1 - Police expenses in years 1960 and 1959, respectively,
LF - rate of persons aged 14-24 among all employees,
W - welfare rate,
M - number of men corresponding to every 1000 of women,
N - population of a state (in hundreds of thousands),
NW - number of not-white persons corresponding to every 1000 of citizens,
U1, U2 - unemployment rate among men aged 14-24 and 35-39, respectively,
X - unequality of income rate (number of families among 100 whose income is lower than half of median of all families income).

```
> uscrime.data <- read.table(file="uscrime.txt",header=T)
```

- Fit a linear regression model taking crime rate as a response variable and all the other variables in the set as predictors.

```
> uscrime.model <- lm(R~.,uscrime.data)
> summary(uscrime.model,corr=T)

Call:
lm(formula = R ~ ., data = uscrime.data)

Residuals:
    Min      1Q  Median      3Q     Max
-34.884 -11.923  -1.135  13.495  50.560

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -6.918e+02  1.559e+02  -4.438 9.56e-05 ***
Age          1.040e+00  4.227e-01   2.460  0.01931 *
S           -8.308e+00  1.491e+01  -0.557  0.58117
Ed           1.802e+00  6.496e-01   2.773  0.00906 **
Ex0          1.608e+00  1.059e+00   1.519  0.13836
Ex1         -6.673e-01  1.149e+00  -0.581  0.56529
LF          -4.103e-02  1.535e-01  -0.267  0.79087
M            1.648e-01  2.099e-01   0.785  0.43806
N           -4.128e-02  1.295e-01  -0.319  0.75196
NW           7.175e-03  6.387e-02   0.112  0.91124
U1          -6.017e-01  4.372e-01  -1.376  0.17798
U2           1.792e+00  8.561e-01   2.093  0.04407 *
W            1.374e-01  1.058e-01   1.298  0.20332
X            7.929e-01  2.351e-01   3.373  0.00191 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 21.94 on 33 degrees of freedom
Multiple R-squared:  0.7692,	Adjusted R-squared:  0.6783
F-statistic: 8.462 on 13 and 33 DF,  p-value: 3.686e-07

Correlation of Coefficients:
    (Intercept) Age   S     Ed    Ex0   Ex1   LF    M     N     NW    U1
Age -0.23
S    0.10       -0.02
Ed  -0.16        0.09 -0.09
Ex0  0.20       -0.07 -0.01  0.18
Ex1 -0.14        0.10  0.05 -0.21 -0.97
LF   0.19        0.15  0.50 -0.32 -0.20  0.27
M   -0.65       -0.29 -0.21 -0.10 -0.07 -0.01 -0.52
N   -0.36        0.01  0.08  0.03 -0.13  0.03 -0.14  0.47
NW  -0.09       -0.33 -0.48  0.17  0.10 -0.20 -0.33  0.25 -0.01
U1   0.33        0.12  0.37 -0.23  0.03  0.07  0.43 -0.57 -0.15 -0.17
U2  -0.28        0.15 -0.20  0.32 -0.12  0.05 -0.05  0.21  0.00  0.01 -0.77
W   -0.26        0.10 -0.17 -0.13 -0.06 -0.04 -0.14 -0.08 -0.16  0.22  0.11
X   -0.22        0.03 -0.35  0.22 -0.06  0.08 -0.20 -0.20 -0.29 -0.05  0.07
    U2    W
```

```
Age
S
Ed
Ex0
Ex1
LF
M
N
NW
U1
U2
W   -0.19
X   -0.11  0.59
```

- Check the data set for collinearity of predictors:
  - make scatterplots for all pairs of predictors (use function pairs()) We can see the there is strong linear relationship between:
    - Ex0 and Ex1
    - U1 and U2
    - W and X

```
> op <- par(mar = c(0,0,0,0))
> pairs(uscrime.data[2:ncol(uscrime.data)])
> par(op)
```

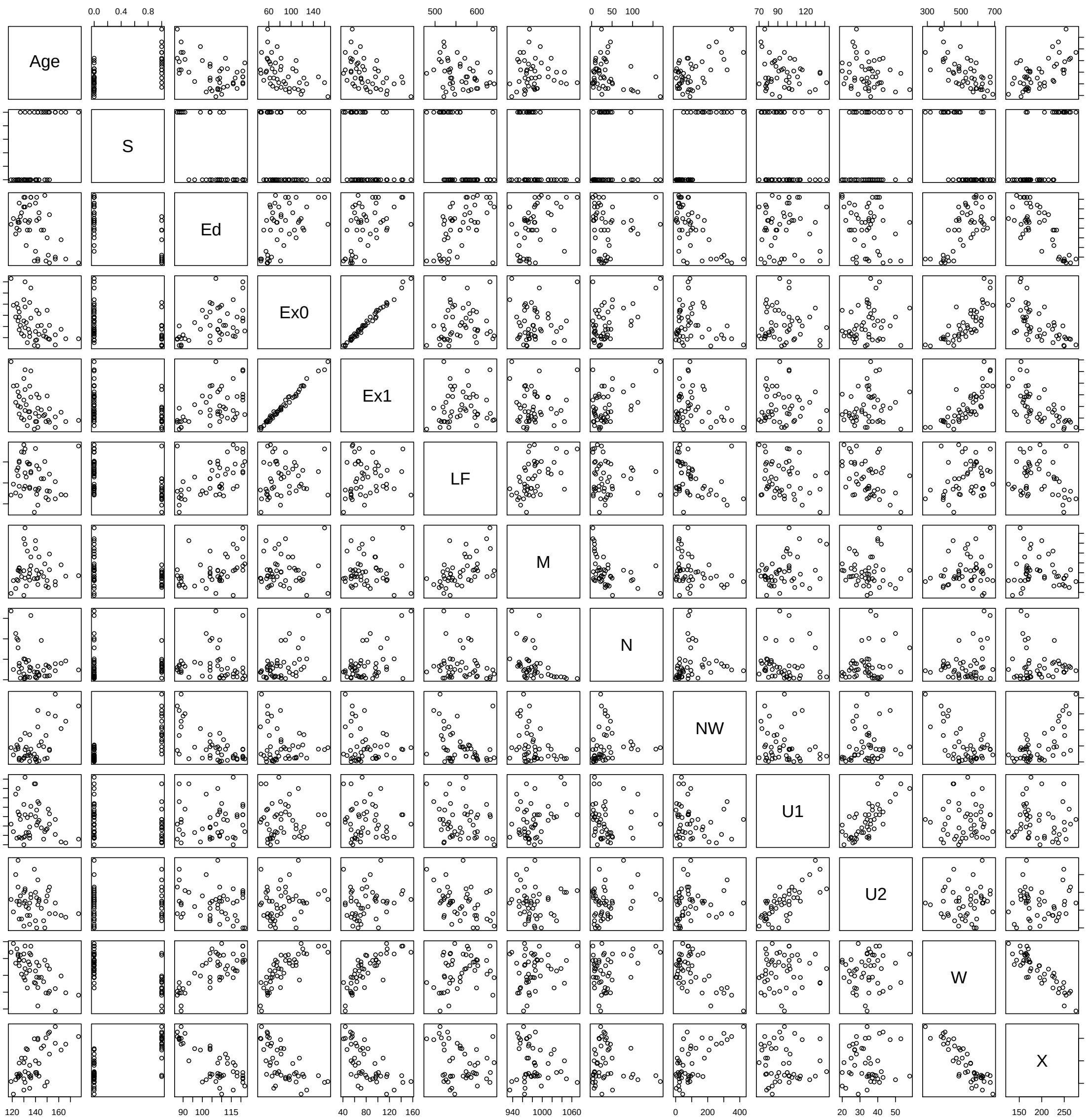

Figure 26: uscrime scatterplots for all pairs of predictors

  - calculate correlations between them

```
> (uscrime.cor <- cor(uscrime.data[2:ncol(uscrime.data)]))
```

```
            Age           S          Ed         Ex0         Ex1         LF
Age  1.00000000  0.58435534 -0.53023964 -0.50573690 -0.51317336 -0.1609488
S    0.58435534  1.00000000 -0.70274132 -0.37263633 -0.37616753 -0.5054695
Ed  -0.53023964 -0.70274132  1.00000000  0.48295213  0.49940958  0.5611780
Ex0 -0.50573690 -0.37263633  0.48295213  1.00000000  0.99358648  0.1214932
Ex1 -0.51317336 -0.37616753  0.49940958  0.99358648  1.00000000  0.1063496
LF  -0.16094882 -0.50546948  0.56117795  0.12149320  0.10634960  1.0000000
M   -0.02867993 -0.31473291  0.43691492  0.03376027  0.02284250  0.5135588
N   -0.28063762 -0.04991832 -0.01722740  0.52628358  0.51378940 -0.1236722
NW   0.59319826  0.76710262 -0.66488190 -0.21370878 -0.21876821 -0.3412144
U1  -0.22438060 -0.17241931  0.01810345 -0.04369761 -0.05171199 -0.2293997
U2  -0.24484339  0.07169289 -0.21568155  0.18509304  0.16922422 -0.4207625
W   -0.67005506 -0.63694543  0.73599704  0.78722528  0.79426205  0.2946323
X    0.63921138  0.73718106 -0.76865789 -0.63050025 -0.64815183 -0.2698865
              M           N          NW          U1          U2           W
Age -0.02867993 -0.28063762  0.59319826 -0.22438060 -0.24484339 -0.67005506
S   -0.31473291 -0.04991832  0.76710262 -0.17241931  0.07169289 -0.63694543
Ed   0.43691492 -0.01722740 -0.66488190  0.01810345 -0.21568155  0.73599704
Ex0  0.03376027  0.52628358 -0.21370878 -0.04369761  0.18509304  0.78722528
Ex1  0.02284250  0.51378940 -0.21876821 -0.05171199  0.16922422  0.79426205
LF   0.51355879 -0.12367222 -0.34121444 -0.22939968 -0.42076249  0.29463231
M    1.00000000 -0.41062750 -0.32730454  0.35189190 -0.01869169  0.17960864
N   -0.41062750  1.00000000  0.09515301 -0.03811995  0.27042159  0.30826271
NW  -0.32730454  0.09515301  1.00000000 -0.15645002  0.08090829 -0.59010707
U1   0.35189190 -0.03811995 -0.15645002  1.00000000  0.74592482  0.04485720
U2  -0.01869169  0.27042159  0.08090829  0.74592482  1.00000000  0.09207166
W    0.17960864  0.30826271 -0.59010707  0.04485720  0.09207166  1.00000000
X   -0.16708869 -0.12629357  0.67731286 -0.06383218  0.01567818 -0.88399728
              X
Age  0.63921138
S    0.73718106
Ed  -0.76865789
Ex0 -0.63050025
Ex1 -0.64815183
LF  -0.26988646
M   -0.16708869
N   -0.12629357
NW   0.67731286
U1  -0.06383218
U2   0.01567818
W   -0.88399728
X    1.00000000
```

We can find pairs with cor ≥ 0.7:

```
> as.vector(apply(which(uscrime.cor>=0.7 & upper.tri(uscrime.cor),arr.ind=T),1,
+                                  function(pair){paste(colnames(uscrime.cor)[pair[1]],colnames(uscrim
```

```
[1] "Ex0 Ex1" "S NW"    "U1 U2"   "Ed W"    "Ex0 W"   "Ex1 W"   "S X"
```

– calculate variance inflation factors for every predictor

```
> library(car)
> vif(uscrime.model)
```

```
      Age         S        Ed       Ex0       Ex1        LF         M         N
 2.698021  4.876751  5.049442 94.633118 98.637233  3.677557  3.658444  2.324326
       NW        U1        U2         W         X
 4.123274  5.938264  4.997617  9.968958  8.409449
```

- Choose the strongest correlated pair of predictors and remove one of them from the model. Compare the new model with the initial one. How does collinearity of predictors affect a model?
  We can see that the strongest correlation and the biggest VIF comes from Ex0 and Ex1. After removing Ex1 (the biggest VIF):

```
> uscrime_no_ex1.model <- lm(R~.,uscrime.data[,!colnames(uscrime.data) %in% c("Ex1")])
> summary(uscrime_no_ex1.model)
```

```
Call:
lm(formula = R ~ ., data = uscrime.data[, !colnames(uscrime.data) %in%
```

```
    c("Ex1")])

Residuals:
   Min     1Q Median     3Q    Max 
-38.76 -13.59   1.09  13.25  48.93 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -7.041e+02  1.529e+02  -4.604 5.58e-05 ***
Age          1.064e+00  4.165e-01   2.556 0.015226 *  
S           -7.875e+00  1.475e+01  -0.534 0.596823    
Ed           1.722e+00  6.286e-01   2.739 0.009752 ** 
Ex0          1.010e+00  2.436e-01   4.145 0.000213 ***
LF          -1.718e-02  1.464e-01  -0.117 0.907297    
M            1.630e-01  2.079e-01   0.784 0.438418    
N           -3.886e-02  1.282e-01  -0.303 0.763604    
NW          -1.299e-04  6.200e-02  -0.002 0.998340    
U1          -5.848e-01  4.319e-01  -1.354 0.184674    
U2           1.819e+00  8.465e-01   2.149 0.038833 *  
W            1.351e-01  1.047e-01   1.290 0.205711    
X            8.040e-01  2.320e-01   3.465 0.001453 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 21.72 on 34 degrees of freedom
Multiple R-squared:  0.7669,	Adjusted R-squared:  0.6846 
F-statistic:  9.32 on 12 and 34 DF,  p-value: 1.351e-07

> vif(uscrime_no_ex1.model)

     Age        S       Ed      Ex0       LF        M        N       NW 
2.670798 4.864533 4.822240 5.109739 3.414298 3.657629 2.321929 3.963407 
      U1       U2        W        X 
5.912063 4.982983 9.955587 8.354136 
```

Now we can see that all VIFs are smaller that 10 and there is one more significant coefficient i.e. Ex0 (the one that was strongly correlated with removed Ex1). The standard errors of cooeficients are larger for model with collinear predictors because of not stable solution of regression equation. We can also see the positive correlation between predictors Ex0 and Ex1 is mirrored by high negative corelation between their correnponding coefficients.