MATH 423/533 - ASSIGNMENT 1
SOLUTIONS

Data stored in three data files on the course website contain x and y variables that are to be used for simple linear regression. The data files are `a1-1.txt`, `a1-2.txt` and `a1-3.txt`.

Code for Assignment 1

```
#Read in data set 1
file1<-"http://www.math.mcgill.ca/dstephens/Regression/Data/a1-1.txt"
data1<-read.table(file1,header=TRUE)
plot(data1$x,data1$y,pch=19)
x1<-data1$x
y<-data1$y
```

(a) Perform a least squares fit of a simple linear regression model (including the intercept) in `R` for each of the three data sets. In particular, for each data set

(i) report the parameter estimates arising from a least squares fit; 2 Marks

(ii) produce a plot of the data with the line of best fit superimposed; 1 Mark

(iii) plot (against the x values) the residuals $e_i, i = 1, \ldots, n$, from the fit; 1 Mark

(iv) comment on the adequacy of the straight line model, based on the residuals plot – that is, comment on whether the assumptions of least squares fitting and how they relate to the residual errors ϵ_i are met by the observed data. 1 Mark

Note: the `R` functions `lm`, `coef` and `residuals` will be useful.

(b) Demonstrate theoretically what happens to the least squares estimates if the predictor is

(i) subjected to a location shift: $x_{i1} \longrightarrow x_{i1} - m$ for some m; 2 Marks

(ii) rescaled: $x_{i1} \longrightarrow lx_{i1}$ for some $l > 0$. 2 Marks

Describe also how the properties of the corresponding estimators change. 1 Mark

---

EXTRA QUESTION FOR STUDENTS IN MATH 533

In the linear model, it is possible to use a different 'best-fit' criterion based on the Euclidean distance between modelled means vector $\mathbf{X}\beta$ and the observed data vector $\mathbf{y}$: that is, we choose β to minimize

$$\|\mathbf{y} - \mathbf{X}\beta\| = \sqrt{\|\mathbf{y} - \mathbf{X}\beta\|^2} = \sqrt{\sum_{i=1}^{n}(y_i - \mathbf{x}_i\beta)^2}.$$

For simple linear regression, derive the two equations that would need to be solved for the elements of $\beta = (\beta_0, \beta_1)^\top$ to find the minimizing values. Explain how the corresponding model assumptions (in terms of the properties of the residual errors and how they define the least squares procedure) differ from those behind least squares.

3 Marks

Another fitting criterion is based on the sum of absolute differences

$$\sum_{i=1}^{n} |y_i - \mathbf{x}_i\beta|$$

Does this criterion lead to estimators with different statistical characteristics to least squares ? Justify your answer.

2 Marks

SOLUTION

```
file1<-"http://www.math.mcgill.ca/dstephens/Regression/Data/a1-1.txt"
data1<-read.table(file1,header=TRUE)
#plot(data1$x,data1$y,pch=19,xlab='x',ylab='y',mar=c(0,1,0,1)); title('Data set 1')
file2<-"http://www.math.mcgill.ca/dstephens/Regression/Data/a1-2.txt"
data2<-read.table(file2,header=TRUE)
#plot(data2$x,data2$y,pch=19,xlab='x',ylab='y',mar=c(0,1,0,1)); title('Data set 2')
file3<-"http://www.math.mcgill.ca/dstephens/Regression/Data/a1-3.txt"
data3<-read.table(file3,header=TRUE)
#plot(data3$x,data3$y,pch=19,xlab='x',ylab='y',mar=c(0,1,0,1)); title('Data set 3')
```

(a) **Data set 1:**

(i) The parameter estimates $(\widehat{\beta}_0, \widehat{\beta}_1)$ are found be

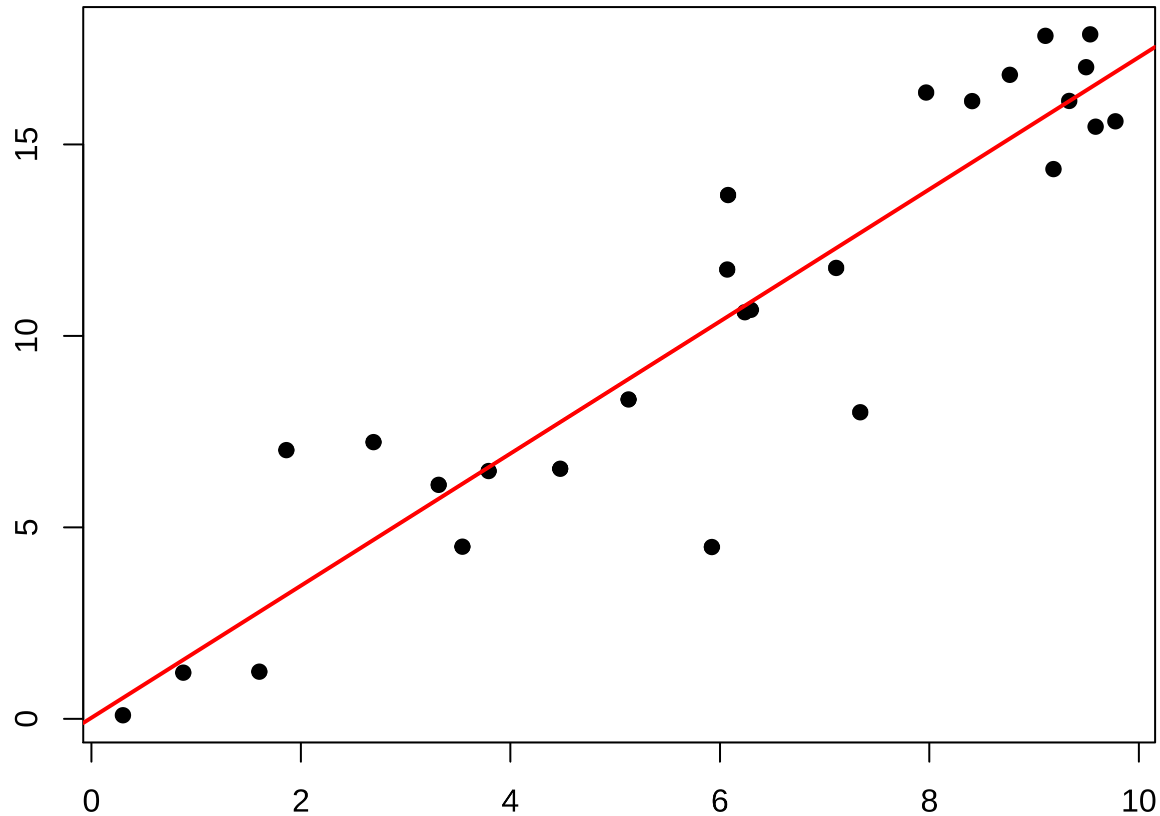
```
fit1<-lm(y~x,data=data1); coef(fit1)

+ (Intercept)           x
+  0.02676552  1.72512091
```

that is, are equal to $(0.0267655, 1.7251209)$

2 Marks

(ii) Here is the plot

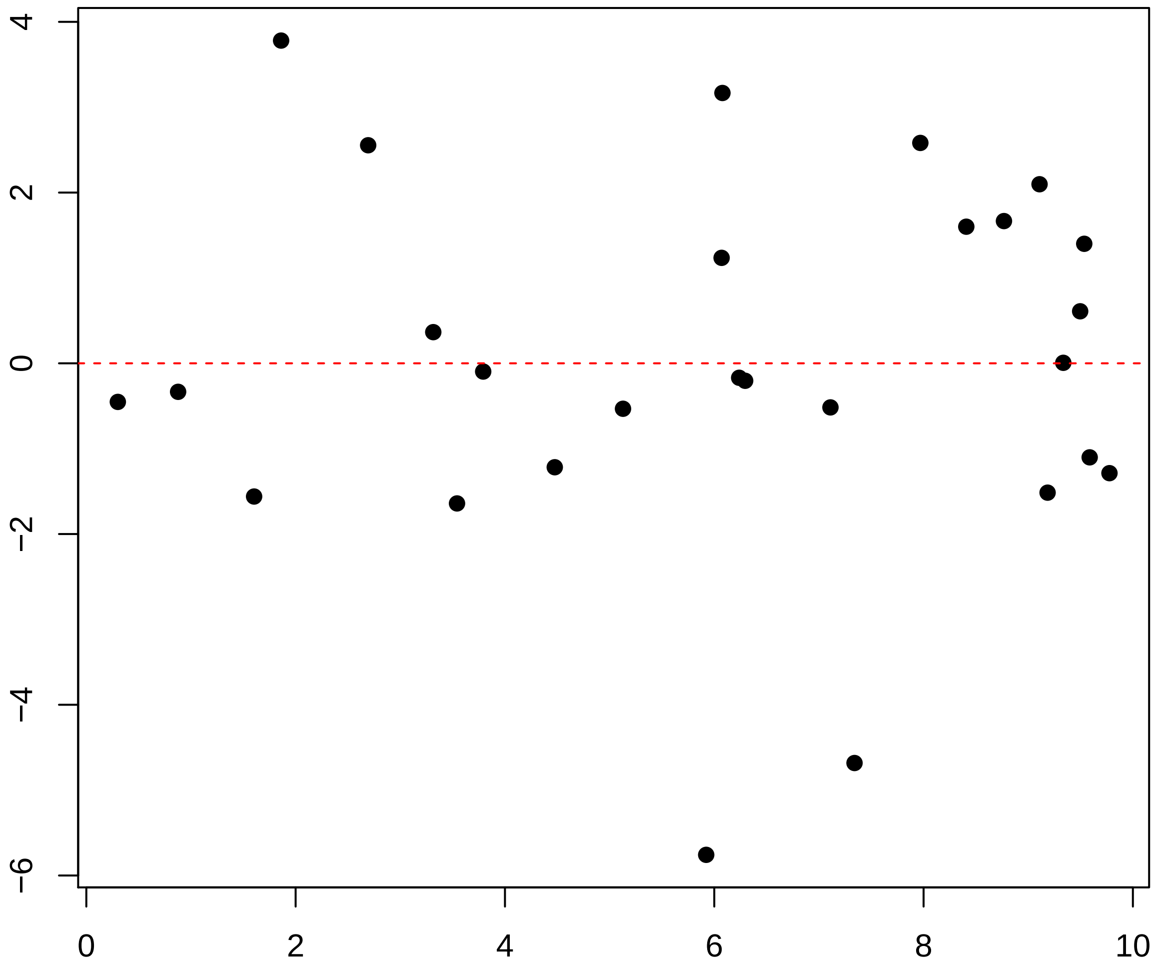
```
par(mar=c(2, 2, .1, .1))
plot(data1$x,data1$y,pch=19,xlab='x',ylab='y')
abline(coef(fit1),col='red',lwd=2)
```



1 Mark

(iii) Here is the residual plot

```r
par(mar=c(2, 2, .1, .1))
plot(data1$x,residuals(fit1),pch=19,xlab='x',ylab='y',mar=c(0,1,0,1))
abline(h=0,col='red',lty=2)
```



1 Mark

(iv) Here the assumptions seem fine; the data set is small, and there may be a couple of outlying points, but there is no strong evidence to suggest that the assumptions of (i) zero mean for all x; (ii) constant variance for all x are not valid here. 1 Mark

Data set 2:

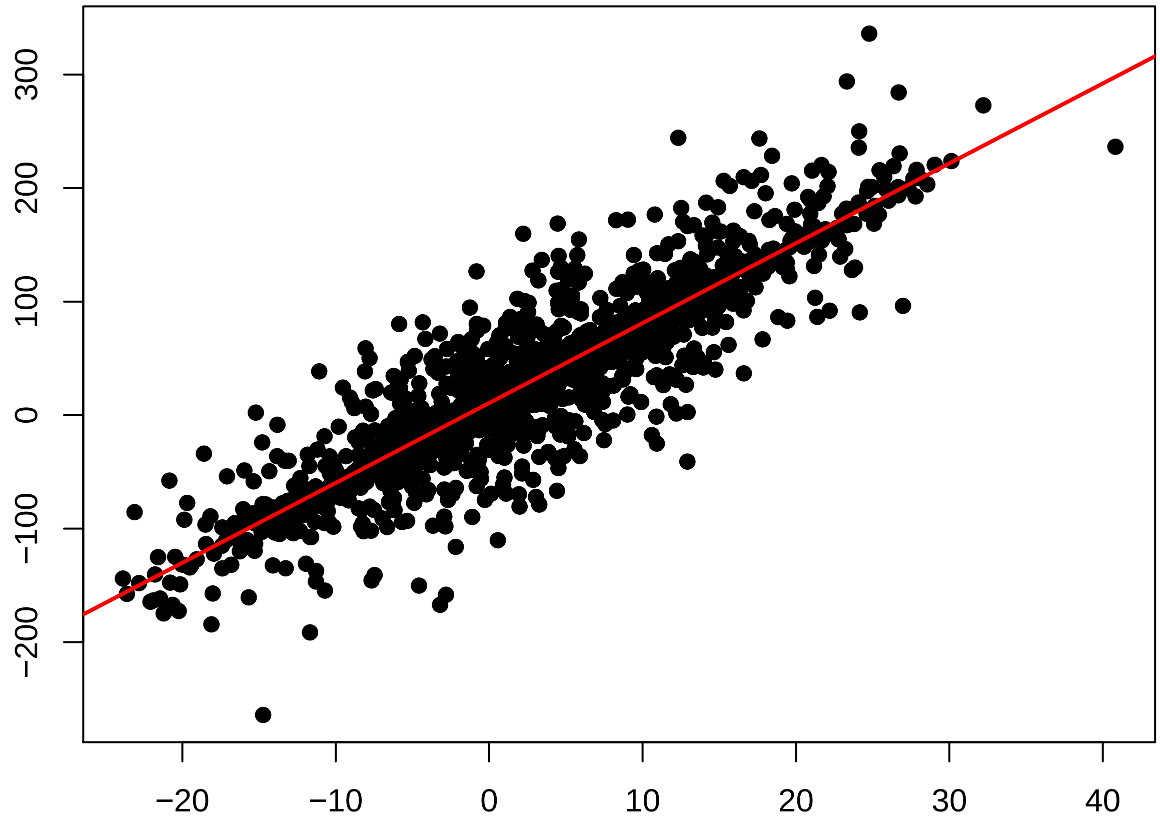
(i) The parameter estimates $(\widehat{\beta}_0, \widehat{\beta}_1)$ are found be

```
fit2<-lm(y~x,data=data2); coef(fit2)

+ (Intercept)           x
+   10.660853    7.037952
```

that is, are equal to $(10.6608534, 7.0379523)$ 2 Marks

(ii) Here is the plot

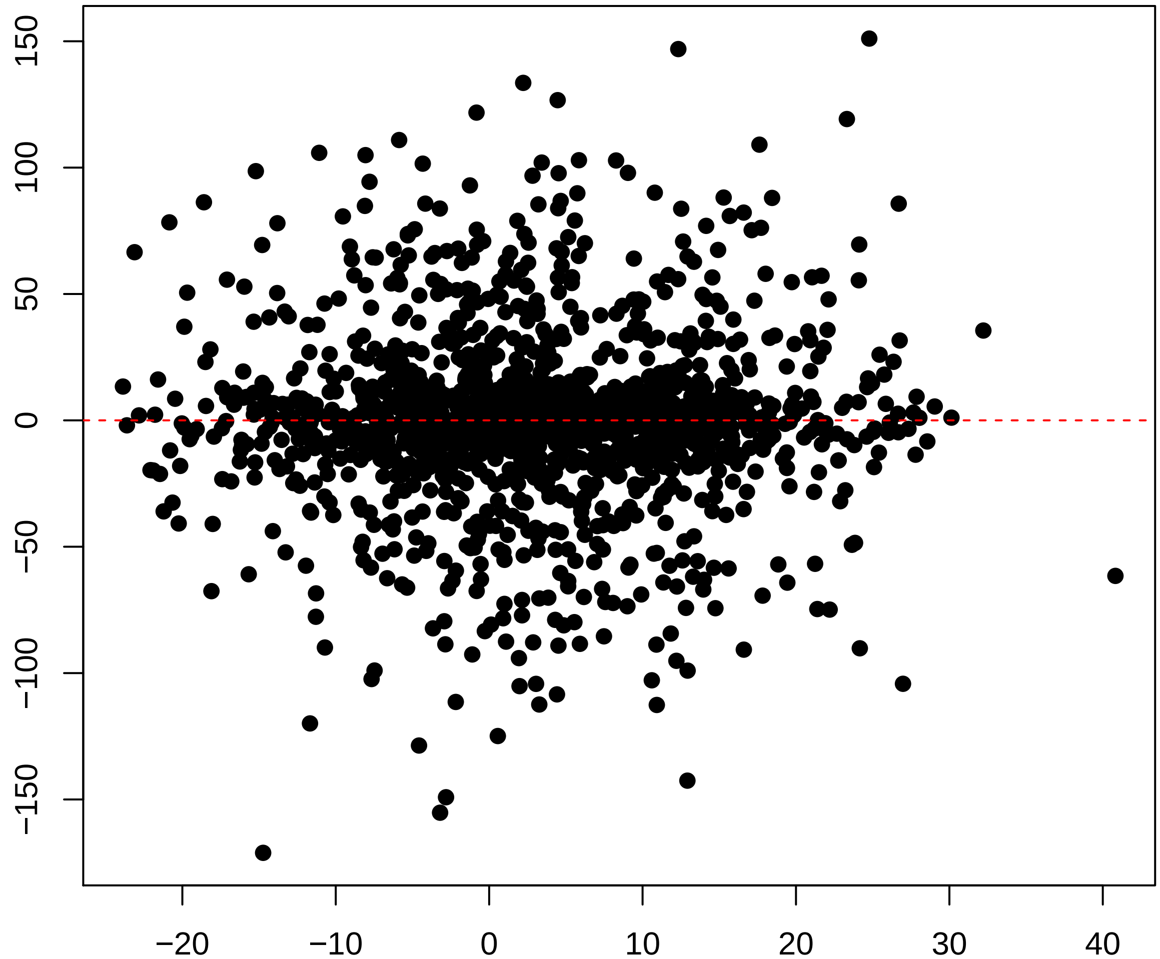
```
par(mar=c(2, 2, .1, .1))
plot(data2$x,data2$y,pch=19,xlab='x',ylab='y')
abline(coef(fit2),col='red',lwd=2)
```



1 Mark

(iii) Here is the residual plot

```r
par(mar=c(2, 2, .1, .1))
plot(data2$x,residuals(fit2),pch=19,xlab='x',ylab='y')
abline(h=0,col='red',lty=2)
```



1 Mark

(iv) The assumptions seem broadly fine; the residuals are apparently zero mean for all x, although the constant variance for all x is a little more questionable, there may be an indication that the variance is smaller near the ends of the x range, but this is not conclusive, as the x values are not uniformly distributed.

1 Mark

Data set 3:

(i) The parameter estimates $(\widehat{\beta}_0, \widehat{\beta}_1)$ are found be

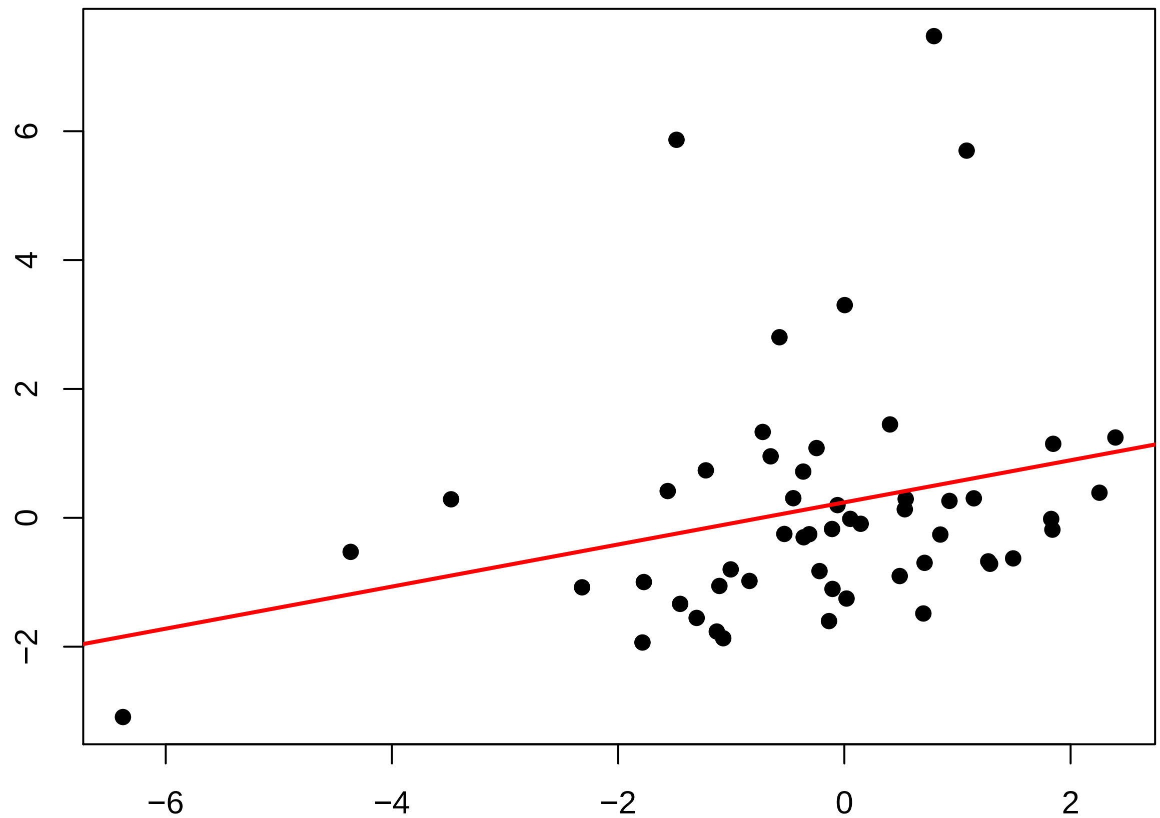
```
fit3<-lm(y~x,data=data3); coef(fit3)

+ (Intercept)           x
+   0.2403328   0.3267628
```

that is, are equal to $(0.2403328, 0.3267628)$

2 Marks

(ii) Here is the plot

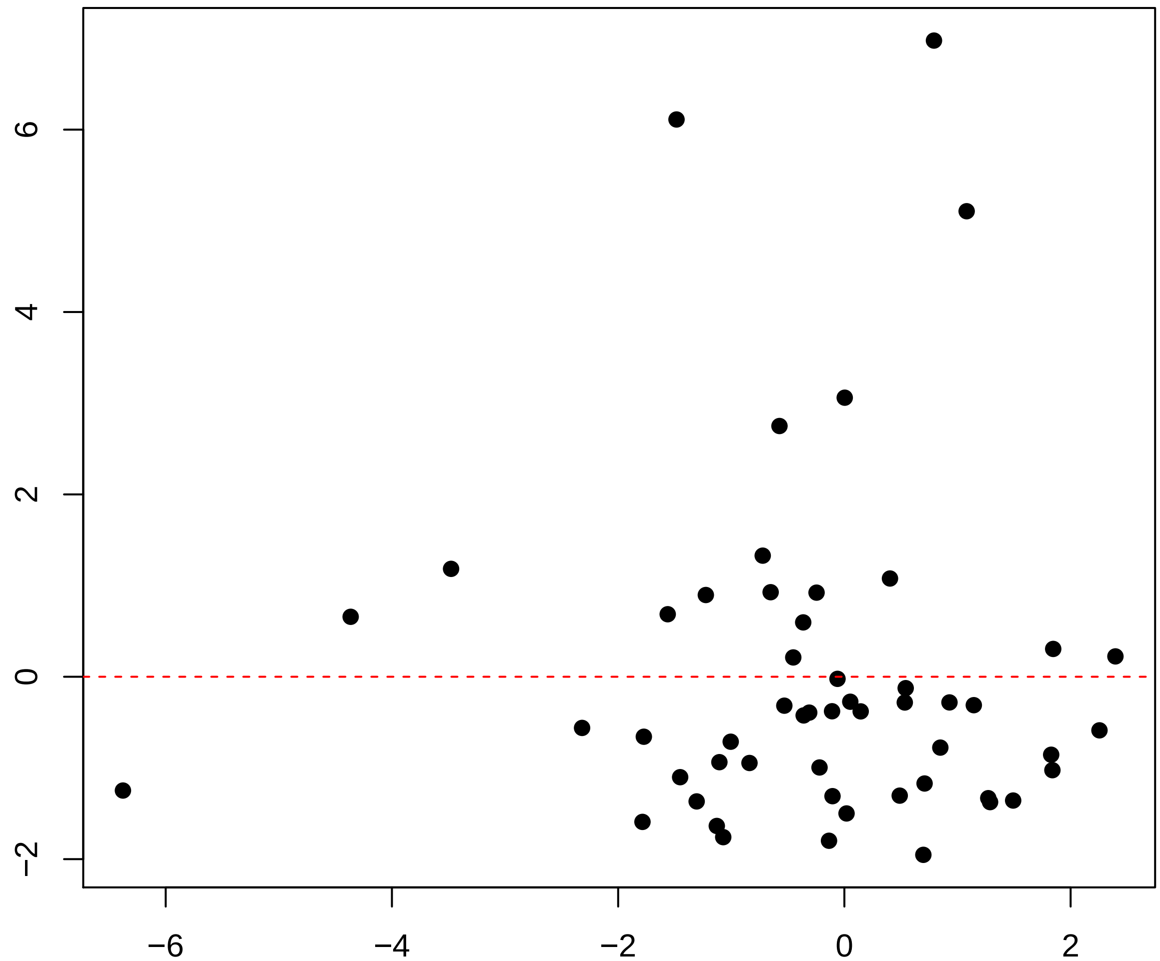
```
par(mar=c(2, 2, .1, .1))
plot(data3$x,data3$y,pch=19,xlab='x',ylab='y')
abline(coef(fit3),col='red',lwd=2)
```



1 Mark

(iii) Here is the residual plot

```
par(mar=c(2, 2, .1, .1))
plot(data3$x,residuals(fit3),pch=19,xlab='x',ylab='y',mar=c(0,1,0,1))
abline(h=0,col='red',lty=2)
```



1 Mark

(iv) The assumptions seem broadly fine, apart from several outliers, which in a real analysis would require some further analysis. It is plausible that the residuals are zero mean for all x; the constant variance assumption is more questionable, but it is not easy to conclude a non-constant variance.

1 Mark

(b) We have from the general theory that if the mean model is correctly specified, the least squares estimators are **unbiased** with variance $\sigma^2(\mathbf{X}^\top\mathbf{X})^{-1}$.

(i) if the predictor is subjected to a location shift by some quantity m, then the original model

$$Y_i = \beta_0 + \beta_1 x_{i1} + \epsilon_i$$

becomes in terms of the new predictor $x_{i1}^{\text{new}} = x_{i1} - m$

$$Y_i = \beta_0^{\text{new}} + \beta_1^{\text{new}} x_{i1}^{\text{new}} + \epsilon_i = \beta_0^{\text{new}} + \beta_1^{\text{new}}(x_{i1} - m) + \epsilon_i = Y_i = (\beta_0^{\text{new}} - m\beta_1^{\text{new}}) + \beta_1^{\text{new}} x_{i1} + \epsilon_i$$

which means that we must have $\beta_1^{\text{new}} = \beta_1$ and

$$\beta_0 = \beta_0^{\text{new}} - m\beta_1^{\text{new}} \qquad \Longleftrightarrow \qquad \beta_0^{\text{new}} = \beta_0 + \beta_1 m$$

that is

$$\begin{bmatrix} \beta_0^{\text{new}} \\ \beta_1^{\text{new}} \end{bmatrix} = \begin{bmatrix} 1 & m \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} = \mathbf{M} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}$$

say, that is, represents a linear reparameterization. Therefore we know that the least squares estimates in the new parameterization are related to those in the original parameterization by the relation $\widehat{\beta}^{\text{new}} = \mathbf{M}\widehat{\beta}$. This leaves the slope estimate unchanged, but the intercept estimate becomes $\widehat{\beta}_0^{\text{new}} = \widehat{\beta}_0 + \widehat{\beta}_1 m$. The estimator variance-covariance matrix is computed using the new design matrix $\mathbf{X}^{\text{new}}$, specifically

$$\frac{1}{nS_{xx}^{\text{new}}} \begin{bmatrix} \sum\limits_{i=1}^n (x_{i1} - m)^2 & -\sum\limits_{i=1}^n (x_{i1} - m) \\ -\sum\limits_{i=1}^n (x_{i1} - m) & n \end{bmatrix}$$

First, we note that

$$S_{xx}^{\text{new}} = \sum_{i=1}^n (x_{i1}^{\text{new}} - \overline{x}_1^{\text{new}})^2 \equiv \sum_{i=1}^n (x_{i1} - \overline{x}_1)^2 = S_{xx}$$

so the variance of estimator $\widehat{\beta}_1$ is unchanged at $n\sigma^2/S_{xx}$. To compute the variance-covariance in matrix form, we use the relationship between new and original parameters; we have the new variance-covariance matrix as

$$\sigma^2 \mathbf{M}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{M}^\top$$

2 Marks

(ii) if the predictor is subjected to a rescaling by l, then the original model becomes in terms of the new predictor $x_{i1}^{\text{new}} = l x_{i1}$

$$Y_i = \beta_0^{\text{new}} + \beta_1^{\text{new}} x_{i1}^{\text{new}} + \epsilon_i = \beta_0^{\text{new}} + \beta_1^{\text{new}}(l x_{i1}) + \epsilon_i$$

that is

$$Y_i = \beta_0^{\text{new}} + l\beta_1^{\text{new}} x_{i1} + \epsilon_i$$

which means that we must have $\beta_0^{\text{new}} = \beta_0$ and

$$\beta_1 = l\beta_1^{\text{new}} \qquad \Longleftrightarrow \qquad \beta_1^{\text{new}} = \frac{1}{l}\beta_1$$

that is

$$\begin{bmatrix} \beta_0^{\text{new}} \\ \beta_1^{\text{new}} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1/l \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} = \mathbf{L} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}$$

say, that is, represents a linear reparameterization. As before we have that $\widehat{\beta}^{\text{new}} = \mathbf{L}\widehat{\beta}$, and that the new variance-covariance matrix is

$$\sigma^2 \mathbf{L}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{L}^\top$$

2 Marks

The estimators are still unbiased for the corresponding estimands, but the variance is modified in the way described above.

1 Mark

Here are the results verified for Data set 2: here we choose $m = 2$ and $l = 4$ for illustration.

```
fit.orig<-lm(y~x,data=data2)
fit.m<-lm(y~I(x-2),data=data2)
fit.l<-lm(y~I(4*x),data=data2)
coef(fit.orig)

+ (Intercept)           x
+   10.660853    7.037952

coef(fit.m)

+ (Intercept)    I(x - 2)
+   24.736758    7.037952

coef(fit.orig)[1] + 2*coef(fit.orig)[2]

+ (Intercept)
+    24.73676

coef(fit.l)

+ (Intercept)    I(4 * x)
+   10.660853    1.759488

coef(fit.orig)[2]/4

+        x
+ 1.759488
```

The estimates match up as predicted. For the variance-covariance matrix results, using the `R` function `vcov`:

```
vcov(fit.orig)

+              (Intercept)            x
+ (Intercept)  0.96348192 -0.027230380
+ x           -0.02723038  0.008561911

vcov(fit.m)

+              (Intercept)     I(x - 2)
+ (Intercept)  0.88880805 -0.010106557
+ I(x - 2)    -0.01010656  0.008561911

M<-matrix(c(1,0,2,1),2,2)
M %*% vcov(fit.orig) %*% t(M)

+              [,1]         [,2]
+ [1,]  0.88880805 -0.010106557
+ [2,] -0.01010656  0.008561911

vcov(fit.l)

+               (Intercept)      I(4 * x)
+ (Intercept)  0.963481922 -0.0068075950
+ I(4 * x)    -0.006807595  0.0005351195

L<-matrix(c(1,0,0,1/4),2,2)
L %*% vcov(fit.orig) %*% t(L)

+               [,1]          [,2]
+ [1,]  0.963481922 -0.0068075950
+ [2,] -0.006807595  0.0005351195
```

EXTRA QUESTION FOR STUDENTS IN MATH 533

If we write for arbitrary β

$$S(\beta) = \sum_{i=1}^{n} (y_i - \mathbf{x}_i\beta)^2 \tag{1}$$

then the new criterion is to estimate β by minimizing $\sqrt{S(\beta)}$; this is a monotone increasing and continuous transform of $S(\beta)$, so therefore we have automatically that it is minimized at the same β values, that is, the least squares estimates – there is no change in the estimate, and the assumptions behind the new criteria are therefore identical to those behind least squares (that is, each data pair (x_i, y_i) contributes to the fit in an identical fashion, additively and exchangeably). We have for simple linear regression

$$\begin{aligned}
\frac{\partial S(\beta)}{\partial \beta_0} &= -\frac{1}{2}\frac{1}{\sqrt{S(\beta)}}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_{i1}) \\
\frac{\partial S(\beta)}{\partial \beta_1} &= -\frac{1}{2}\frac{1}{\sqrt{S(\beta)}}\sum_{i=1}^{n}x_{i1}(y_i - \beta_0 - \beta_1 x_{i1})
\end{aligned}$$

and equating to zero gives the two equations. 3 Marks

For the criterion

$$\sum_{i=1}^{n} |y_i - \mathbf{x}_i\beta|$$

note that this can be written in the form

$$\sum_{i=1}^{n} w(y_i, x_i; \beta)(y_i - \mathbf{x}_i\beta)^2 \tag{2}$$

where

$$w(y_i, x_i; \beta) = \frac{1}{|y_i - \mathbf{x}_i\beta|} \qquad i = 1, \ldots, n$$

Comparing (2) with (1) we see that in general the least squares solution is not a minimizer of (2) as the "weights" $w(y_i, x_i; \beta)$ are not in general equal. Thus the estimators are **not** (almost surely) equal to the least squares estimators for finite n.

2 Marks

The form of (2) suggests an iterative procedure for finding the minimizing values:

1. Initialize the iteration at step $r = 0$ to the values $\beta^{(0)}$ (say, the least squares estimates); compute the corresponding weights

   $$w_i^{(0)} = \frac{1}{|y_i - \mathbf{x}_i\beta^{(0)}|} \qquad i = 1, \ldots, n$$

   where the weight is set to zero if the denominator is zero.

2. For $r = 0, 1, 2, \ldots$:

   (i) Solve the minimization problem

   $$\sum_{i=1}^{n} w_i^{(r)}(y_i - \mathbf{x}_i\beta)^2 = (\mathbf{y} - \mathbf{X}\beta)^\top \mathbf{W}^{(r)}(\mathbf{y} - \mathbf{X}\beta) \tag{3}$$

   say, where $\mathbf{W}^{(r)} = \text{diag}(w_1^{(r)}, \ldots, w_n^{(r)})$, to obtain the new values $\beta^{(1)}$; this can be achieved analytically as

   $$\beta^{(r+1)} = (\mathbf{X}^\top\mathbf{W}^{(r)}\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{W}^{(r)}\mathbf{y}$$

(ii) Compute new weights

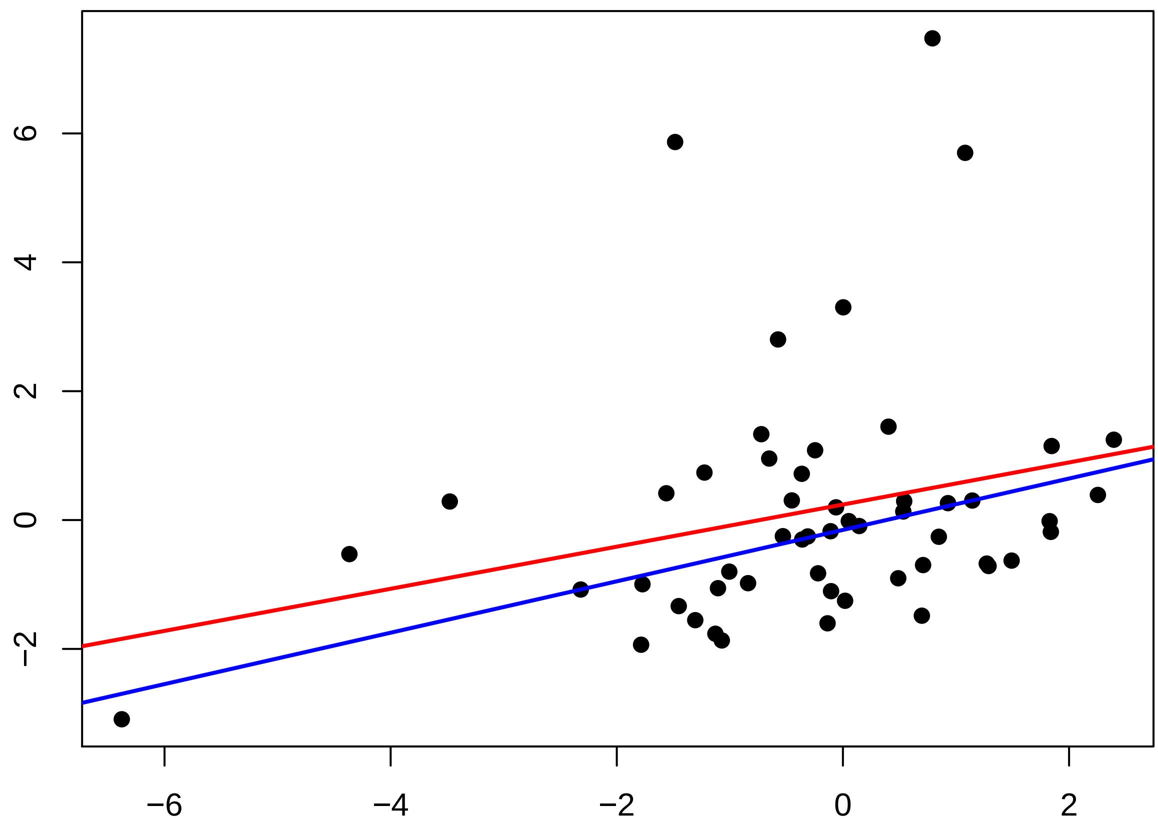
$$w_i^{(r+1)} = \frac{1}{|y_i - \mathbf{x}_i \beta^{(r+1)}|} \qquad i = 1, \ldots, n$$

where the weight is set to zero if the denominator is zero.

3. Repeat 2. until convergence of the sequence $\beta^{(r)}, r = 1, 2, \ldots$.

The following code gives some numerical solutions (for Data set 3):

```
#Numerical methods for solution
#Define a function for use in 'optim'
least.l1<-function(bev,Xm,yv){

        fval<-sum(abs(yv-Xm%*%bev))

        return(fval)
}

be.start<-coef(lm(y~x,data3))
l1.fit<-optim(be.start,fn=least.l1,Xm=cbind(1,data3$x),yv=data3$y)
par(mar=c(2, 2, .1, .1))
plot(data3$x,data3$y,pch=19,xlab=expression(x[1]),ylab='y')
abline(coef(lm(y~x,data=data3)),col='red',lwd=2)
abline(l1.fit$par,lwd=2,col='blue')
```



Using the `quantreg` library in `R`

```
library(quantreg)

+ Loading required package:  SparseM
+ Loading required package:  methods
+
+ Attaching package:  'SparseM'
```

```
+ The following object is masked from 'package:base':
+
+     backsolve
```

```
fit.l1<-rq(y~x,0.5,data=data3)
summary(fit.l1)
```

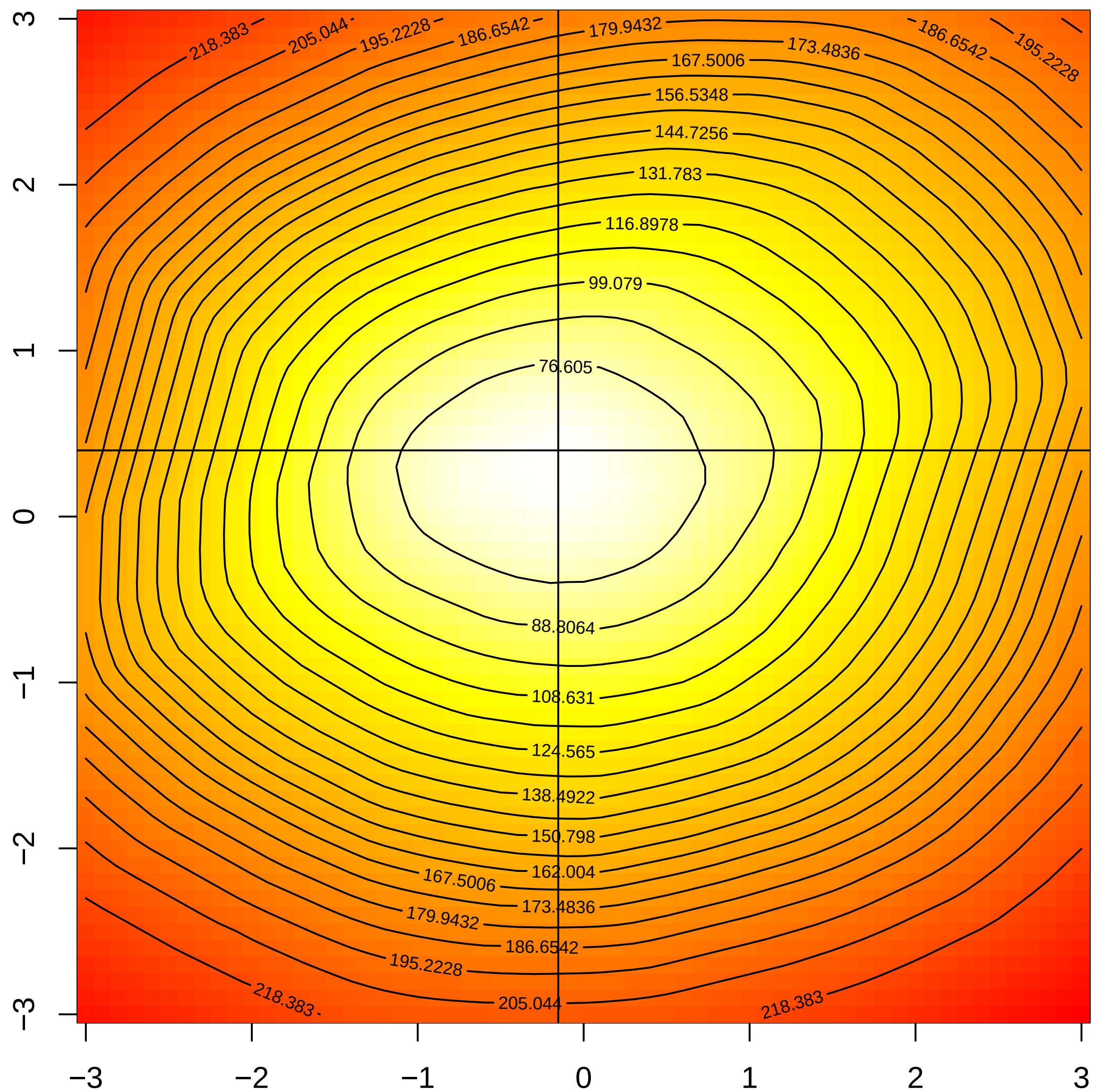
```
+
+ Call: rq(formula = y ~ x, tau = 0.5, data = data3)
+
+ tau: [1] 0.5
+
+ Coefficients:
+               coefficients lower bd upper bd
+ (Intercept) -0.15321     -0.51563 -0.03863
+ x            0.39879      0.07487  0.46292
```

Produce a surface plot

```
be0<-seq(-3,3,by=0.1)
be1<-seq(-3,3,by=0.1)

Resid.func.l1<-function(b0,b1,xv,yv){

        return((abs(yv-(b0+b1*xv))))


}
Resid.func.l1(0,1,data3$x[1],data3$y[1])
```

```
+ [1] 2.001
```

```
Smat<-outer(be0,be1,'*')*0
for(i in 1:nrow(data3)){

        Smat<-Smat+outer(be0,be1,Resid.func.l1,xv=data3$x[i],yv=data3$y[i])

}

zval<-quantile(Smat,prob=c(1:19)/20)
par(mar=c(2, 2, .1, .1))
image(be0,be1,Smat,col=rev(heat.colors(200)),xlab=expression(beta[0]),ylab=expression(beta[1])
contour(be0,be1,Smat,add=T,levels=zval)
abline(v=coef(fit.l1)[1],h=coef(fit.l1)[2])
```



Using an iterative procedure (100 steps)

```
be.old<-coef(lm(y~x,data3))
nsteps<-100
yvec<-data3$y
xvec<-data3$x
e.old<-yvec-cbind(1,xvec) %*% be.old
for(istep in 1:nsteps){
        wval<-1/abs(e.old)
        be.old<-coef(lm(yvec~xvec,weights=wval,subset=(wval < Inf)))
        e.old<-yvec-cbind(1,xvec) %*% be.old
}
print(be.old)

+ (Intercept)        xvec
+   -0.1532125   0.3987872

print(coef(fit.l1))

+ (Intercept)           x
+   -0.1532125   0.3987872
```