Chapter 8. Problems with the Error

8.1 Generalized Least Squares:
used when errors are correlated or have unequal variances

Suppose \( Var(\varepsilon) = \sigma^2 \Sigma \), where \( \sigma^2 \) is unknown but \( \Sigma \) is known. Write \( \Sigma = SS^T \) where \( S \) is a triangular matrix using the Choleski decomposition. Then

\[
y = X\beta + \varepsilon
\]

\[
S^{-1}y = S^{-1}X\beta + S^{-1}\varepsilon
\]

Now we find that

\[
Var(\varepsilon') = Var\left(S^{-1}\varepsilon\right) = S^{-1}Var(\varepsilon)S^{-T} = S^{-1}\sigma^2 SS^T S^{-T} = \sigma^2 I
\]

This means that GLS reduces to OLS by a regression of \( y' = S^{-1}y \) on \( S^{-1}X \) which has error \( \varepsilon' \) that’s iid (independent and identically distributed). The sum of squares is now

\[
\left(S^{-1}y - S^{-1}X\beta\right)^T\left(S^{-1}y - S^{-1}X\beta\right) = (y - X\beta)^T S^{-T} S^{-1} (y - X\beta) = (y - X\beta)^T \Sigma^{-1} (y - X\beta)
\]

and this is minimized by

\[
\hat{\beta} = \left(X^T\Sigma^{-1}X\right)^{-1}X^T\Sigma^{-1}y
\]

Also,

\[
Var(\hat{\beta}) = \left(X^T\Sigma^{-1}X\right)^{-1} \sigma^2
\]

Ex 8.1 GLS when errors are autocorrelated (or have unequal variances)

data(globwarm)
head(globwarm)
  nhtemp wusa jasper westgreen chesapeake tornetrask urals mongolia tasman year
1000  0.49  0.85  0.39  -0.23   -0.76  0.85  0.6  1.31  0.55  1995
1001  0.49  0.83  0.37  -0.24   -0.55  0.69  0.6  1.33  0.52  1996
1002  0.48  0.82  0.35  -0.23   -0.32  0.56  0.6  1.33  0.50  1997
1003  0.66  0.82  0.34  -0.21   -0.07  0.44  0.6  1.30  0.49  1998
1004  0.46  0.84  0.32  -0.20   0.17  0.37  0.6  1.24  0.49  1999
1005  0.40  0.88  0.31  -0.19   0.39  0.34  0.6  1.13  0.50  2000
dim(globwarm)
[1] 1001  10

tail(globwarm)
  nhtemp wusa jasper westgreen chesapeake tornetrask urals mongolia tasman year
1995  0.49  0.85  0.39  -0.23   -0.76  0.85  0.6  1.31  0.55  1995
1996  0.49  0.83  0.37  -0.24   -0.55  0.69  0.6  1.33  0.52  1996
1997  0.48  0.82  0.35  -0.23   -0.32  0.56  0.6  1.33  0.50  1997
1998  0.66  0.82  0.34  -0.21   -0.07  0.44  0.6  1.30  0.49  1998
1999  0.46  0.84  0.32  -0.20   0.17  0.37  0.6  1.24  0.49  1999
2000  0.40  0.88  0.31  -0.19   0.39  0.34  0.6  1.13  0.50  2000

modell <-
  lm(nhtemp~wusa+jasper+westgreen+chesapeake+tornetrask+urals+mongolia+tasman,data=globwarm)
library(car)
durbinWatsonTest(modell)
# There is a significant positive autocorrelation.
cor(resid(modell)[-1],resid(modell)[-length(resid(modell))])
# Successive errors are positively correlated.
plot(resid(modell),type="o")

summary(modell)
# WRONG printout because of significant autocorrelation among the residuals.
  Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.2425552  0.0270115 -8.9797 1.972e-15
wusa 0.0773844  0.0429266  1.8027 0.0736475
jasper -0.2287948  0.0781074 -2.9292 0.0039859
westgreen 0.0095839  0.0418405  0.2291 0.8191679
chesapeake -0.0321117  0.0340522 -0.9430 0.3473462
tornetrask 0.0926676  0.0450530  2.0569 0.0416114
urals 0.1853691  0.0914285  2.0275 0.0445674
mongolia 0.0419725  0.0457935  0.9166 0.3609955
tasman 0.1154529  0.0301110  3.8342 0.0001919

n = 145, p = 9, Residual SE = 0.17577, R-Squared = 0.48
```
require(nlme)
model2 <- gls(nhtemp~wusa+jasper+westgreen+chesapeake+tornetrask+urals+mongolia+tasman,
data=na.omit(globwarm),correlation=corAR1(form=~year))

# without "correlation=corAR1", printout would be the same as "lm" printout
summary(model2)
```

```
Generalized least squares fit by REML
Model: nhtemp ~ wusa + jasper + westgreen + chesapeake + tornetrask + urals + mongolia + tasman
Data: na.omit(globwarm)
AIC       BIC   logLik
-108.2074 -76.16822 65.10371
Correlation Structure: AR(1)
Formula: ~year
Parameter estimate(s):
  Phi
0.7109922

Coefficients:  # CORRECT printout.
             Value  Std.Error   t-value p-value
(Intercept) -0.23010624 0.06702406 -3.433188  0.0008
wusa         0.06673819 0.09877211  0.675678  0.5004
jasper      -0.20244335 0.18802773 -1.076668  0.2835
westgreen   -0.00440299 0.08985321 -0.049002  0.9610
chesapeake  -0.00735289 0.07349791 -0.100042  0.9205
tornetrask   0.03835169 0.09482515  0.404446  0.6865
urals        0.24142199 0.22871028  1.055580  0.2930
mongolia     0.05694978 0.10489786  0.542907  0.5881
tasman       0.12034918 0.07456983  1.613913  0.1089

Correlation:
           (Intr) wusa   jasper wstgrn chespk trntrs urals  mongol
wusa       -0.517
jasper     -0.058 -0.299
westgreen  0.330 -0.533  0.121
chesapeake 0.090 -0.314  0.230  0.147
tornetrask -0.430  0.499 -0.197 -0.328 -0.441
urals     -0.110 -0.142 -0.265  0.075 -0.064 -0.346
mongolia   0.459 -0.437 -0.205  0.217  0.449 -0.343 -0.371
tasman     0.037 -0.322  0.065  0.134  0.116 -0.434  0.416 -0.017

Residual standard error: 0.204572
Degrees of freedom: 145 total; 136 residual
```

```
Ex 8.1.1 One more example of GLS (Generalized Least Squares)
```
```
data(longley)
dim(longley)
[1] 16  7
```
longley
<table>
<thead>
<tr>
<th>GNP.deflator</th>
<th>GNP</th>
<th>Unemployed</th>
<th>Armed.Forces</th>
<th>Population</th>
<th>Year</th>
<th>Employed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1947</td>
<td>83.0</td>
<td>234.289</td>
<td>235.6</td>
<td>159.0</td>
<td>107.608</td>
<td>1947</td>
</tr>
<tr>
<td>1948</td>
<td>88.5</td>
<td>259.426</td>
<td>232.5</td>
<td>145.6</td>
<td>108.632</td>
<td>1948</td>
</tr>
<tr>
<td>1949</td>
<td>88.2</td>
<td>258.054</td>
<td>368.2</td>
<td>161.6</td>
<td>109.773</td>
<td>1949</td>
</tr>
</tbody>
</table>

<< Omitted to save paper >>>

```r
model1 <- lm(Employed ~ GNP + Population, longley)
summary(model1)
```

```
#WRONG printout

Estimate  Std. Error t value  Pr(>|t|)
(Intercept) 88.938798  13.785027  6.4518 2.160e-05
GNP          0.063172   0.010647  5.9332 4.958e-05
Population  -0.409743   0.152137 -2.6933   0.01843

n = 16, p = 3, Residual SE = 0.54592, R-Squared = 0.98
```

```r
cor(longley$GNP, longley$Pop)
```

```
[1] 0.9910901
```

```r
cor(residuals(model1)[-1], residuals(model1)[-16])
```

```
#Autocorrelation of successive residuals

[1] 0.3104092
```

```r
durbinWatsonTest(model1)
```

```
lag      Autocorrelation  D-W Statistic  p-value
1       0.2878177      1.301484   0.044
```

```
#Significant positive autocorrelation
```

```r
model2 <- gls(Employed ~ GNP + Population, correlation=corAR1(form=~Year), data=longley)
summary(model2)
```

```
Generalized least squares fit by REML
Model: Employed ~ GNP + Population

Correlation Structure: AR(1)
Formula: ~Year
Parameter estimate(s):
  Phi
0.6441692

Coefficients:  Value  Std.Error  t-value  p-value
(Intercept)  101.85813  14.198932  7.173647  0.0000
GNP           0.07207  0.010606  6.795485  0.0000
Population  -0.54851  0.154130 -3.558778  0.0035

Correlation:
  (Intr)  GNP
GNP      0.943
Population -0.997 -0.966

Residual standard error: 0.689207
Degrees of freedom: 16 total; 13 residual
```

```r
intervals(model2)
```

```
Approximate 95% confidence intervals

Coefficients:  lower    est.    upper
(Intercept)  71.18320  101.85813 132.53306
GNP          0.049159   0.0720708  0.0949831
Population  -0.881491  -0.548513  -0.2155365

Correlation structure:
  lower    est.    upper
Phi -0.4432383  0.6441692  0.9645041
```

```
# Autocorrelation is no longer significant.
```

8.2 Weighted Least Squares: “rule of thumb” weight =inverse of variance

Sometimes the errors are uncorrelated but have unequal variance. That is, \( \Sigma = diag\left( \frac{1}{w_1}, \ldots, \frac{1}{w_p} \right) \), i.e.,

\[
S = diag\left( \frac{1}{\sqrt{w_1}}, \ldots, \frac{1}{\sqrt{w_p}} \right).
\]

So we regress \( \sqrt{w_i} y_i \) on \( \sqrt{w_i} x_i \). Cases with low variability get a high weight, high variability a low weight. Some examples:
1. Errors proportional to a predictor: $Var(e_i) \propto x_i$ suggests $w_i = \frac{1}{x_i}$.

2. When $y_i$ are the averages of $n_i$ observations, then $Var(y_i) = \frac{\sigma^2}{n_i}$, which suggests $w_i = n_i$.

**Ex 8.2 WLS (Weighted Least Squares)**

```r
library(faraway)
data(fpe)
dim(fpe)
[1] 24 14
fpe[1:3,]
   EI  A  B  C  D  E  F  G  H  J  K  A2  B2  N
  Ain  260 51  36  23  9  5  4  4  3  3 105 114  17
  Alpes 75  14  17  9  3  1  2  1  1  1  32  31  5
  Ariege107  27  18  13  2  2  2  1  1  1  57  33  6

N = Total # of voters in the second round – Total # of voters in the first round

model1 <- lm(A2 ~ A+B+C+D+E+F+G+H+J+K+N-1, data=fpe, weight=1/EI)  #with WEIGHT
model2 <- lm(A2 ~ A+B+C+D+E+F+G+H+J+K+N-1, data=fpe)  #without WEIGHT
round(coef(model1),3)
  A      B      C      D      E      F      G      H      J      K      N
  1.067 -0.105  0.246  0.926  0.249  0.755  1.972 -0.566  0.612  1.211  0.529
round(coef(model2),3)
  A      B      C      D      E      F      G      H      J      K      N
  1.075 -0.125  0.257  0.905  0.671  0.783  2.166 -0.854  0.144  0.518  0.558

plot(1:11, coef(model1), xlim=c(0,12),ylim=c(-1,2.5), type="n", xaxt="n",
xlab="departments",ylab="Coefficients of two models")
axis(1,at=1:11,labels=c("A","B","C","D","E","F","G","H","J","K","N"))
points(1:11, coef(model1), pch=16, col="blue")
points(1:11, coef(model2), pch=1, col="black")
legend(locator(1),pch=c(1,16),col=c("black","blue"),c("with weight","w/o weight"))

With weight, there are substantial changes in estimates for some of the lesser candidates.

round(lm(A2 ~ A+B+C+D+E+F+G+H+J+K+N-1, data=fpe, weight=53/EI)$coef,3)
  A      B      C      D      E      F      G      H      J      K      N
  1.067 -0.105  0.246  0.926  0.249  0.755  1.972 -0.566  0.612  1.211  0.529

1/EI or 53/EI makes no difference in estimated coefficients.

Let’s set those coefficients greater than 1.0 as 1 using “offset”.

round(lm(A2 ~ offset(A+G+K)+C+D+E+F+N-1, data=fpe, weight=1/EI)$coef,3)
  C      D      E      F      N
  0.226  0.970  0.390  0.744  0.609

**Interpretation:** 22.6% of the voters who voted for “C” voted for Mitterand in the second round.

Textbook introduces another way of estimating the coefficients like this:

```r
require(mgcv)  # Mixed GAM Computation vehicle
```
Ex 8.2.1 One more example of WLS (Weighted Least Squares)

```r
data(cars)
dim(cars)
[1] 50  2
head(cars)
  speed dist
1     4    2
2     4   10
3     7    4
4     7   22
5     8   16
6     9   10

mod11 <- lm(dist~speed,data=cars)  #without WEIGHT, i.e., usual "lm"
par(mfrow=c(1,2))
plot(dist~speed,data=cars)
plot(mod11,1)

Residuals show "megaphone" pattern (i.e., unequal variance). One solution is transformation of the variables (like Box-Cox transformation to stabilize the variance, see next chapter), and another solution is WLS.

summary(mod11)
## A tibble: 2 x 5
## 1 (Intercept)    speed     t value  Pr(>|t|)  
##   -17.58   3.93       -2.60  0.0123

n = 50, p = 2, Residual SE = 15.3796, R-Squared = 0.65

mod12 <- lm(dist~speed,weight=1/(speed^2),data=cars)  #with WEIGHT=1/x^2
summary(mod12)
## A tibble: 2 x 5
## 1 (Intercept)    speed     t value  Pr(>|t|)  
##   -9.57    3.37       11.63 1.44e-15

n = 50, p = 2, Residual SE = 0.9947, R-Squared = 0.74

mod13 <- gls(dist~speed,data=cars,weight=varConstPower(1,form=~speed))  #GLS
summary(mod13)
## A tibble: 2 x 5
## 1 (Intercept)    speed     t value  Pr(>|t|)  
##   -9.57    3.37       11.63 1.44e-15

n = 50, p = 2, Residual SE = 0.9947, R-Squared = 0.74
```
### 8.3 Testing for Lack of Fit

#### Ex 8.3 Testing for lack of fit

```r
data(corrosion)
dim(corrosion)
[1] 13  2
corrosion[1:3,]
  Fe  loss
1 0.01 127.6
2 0.48 124.0
3 0.71 110.8

modell <- lm(loss ~ Fe, corrosion)
summary(modell)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  129.787    1.403  92.522  < 2e-16 ***
  Fe        -24.020    1.280 -18.773 1.06e-09 ***
---
```

---

*This means SD = 3.16 + x^1.022, i.e., SD is proportional to x*

---

**Variance function:**
- **Structure:** Constant plus power of variance covariate
- **Formula:** ~speed

**Parameter estimates:**
- **const** = 3.1604
- **power** = 1.0224

**Coefficients:**

<table>
<thead>
<tr>
<th></th>
<th>Value</th>
<th>Std.Error</th>
<th>t-value</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>-11.085378</td>
<td>4.052378</td>
<td>-2.735524</td>
<td>0.0087</td>
</tr>
<tr>
<td>speed</td>
<td>3.484162</td>
<td>0.320237</td>
<td>10.879947</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

**Correlation:**
- (Intercept) speed -0.9

**Residual standard error:** 0.7636833
**Degrees of freedom:** 50 total; 48 residual

```r
plot(dist~speed,data=cars)
abline(model1,lty=1,col=1)
abline(model2,lty=2,col=2)
abline(model3,lty=3,col=4)
legend(locator(1),lty=1:3,col=c(1,2,4),c("usual lm","weight=1/(x^2)","gls"))
```
Residual standard error: 3.058 on 11 degrees of freedom
Multiple R-squared: 0.9697, Adjusted R-squared: 0.967
F-statistic: 352.3 on 1 and 11 DF, p-value: 1.055e-09

```r
par(mfrow=c(1,2))
plot(loss ~ Fe, data=corrosion, xlab="Iron content",ylab="Weight loss")
abline(coef(model1))
plot(loss ~ Fe, data=corrosion, xlab="Iron content",ylab="Weight loss")
abline(coef(model1))
points(corrosion$Fe, fitted(model2),pch=15)
```

```r
model2 <- lm(loss ~ factor(Fe), corrosion)
anova(model1, model2)
Analysis of Variance Table

<table>
<thead>
<tr>
<th>Model</th>
<th>Df</th>
<th>Sum Sq</th>
<th>Mean Sq</th>
<th>F value</th>
<th>Pr(&gt;F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>11</td>
<td>102.850</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>11.782</td>
<td>91.069</td>
<td>9.2756</td>
<td>0.008623 **</td>
</tr>
</tbody>
</table>

We conclude that there is a significant lack of fit. Beware – this method works only when you have multiple readings per x like this example.

```r
model11 <- lm(loss ~ Fe+I(Fe^2)+I(Fe^3), corrosion)
anova(model11)
Analysis of Variance Table

Response: loss
Df Sum Sq Mean Sq F value  Pr(>F)
Fe 1 3293.8 3293.8 658.0120 1.002e-09 ***
I(Fe^2) 1  2.8   2.8   0.5522  0.4764
I(Fe^3) 1 55.0  55.0  10.9947  0.0090 **
Residuals 9 45.0  5.0

If you want to overlay the cubic model on the plot:
plot(loss~Fe)
x <- seq(0,2,0.01)
y <- predict(model11, data.frame(Fe=x))
lines(x,y, lty=2, col=4)
```

6th degree polynomial can go through 7 data points perfectly, but very little is achieved!

```r
model3 <- lm(loss ~ Fe+I(Fe^2)+I(Fe^3)+I(Fe^4)+I(Fe^5)+I(Fe^6),corrosion)
plot(loss~Fe, corrosion, ylim=c(60,130))
points(corrosion$Fe, fitted(model2),pch=15)
grid <- seq(0,2,length=50)
lines(grid,predict(model3, data.frame(Fe=grid)))
```
8.4 Robust Regression
Used when the errors are not normally distributed.

M-estimates choose $\beta$ to minimize:

$$\sum_{i=1}^{n} f(y_i - x_i^T \beta)$$

Some possible choices for function $\rho$ are:
1). $f = x^2$ is the least squares regression.
2). $f = |x|$ is the LAD (least absolute deviation) regression. It’s also called the $L_1$ regression.
3). $f(x) = \begin{cases} \frac{x^2}{2} & \text{if } |x| \leq c \\ c |x| - \frac{c^2}{2} & \text{otherwise} \end{cases}$ is called Huber’s method and $c$ is a robust estimator of $\sigma$. A value proportional to the median of $|\hat{e}|$ is suitable.

The normal equation $X^T (y - X\hat{\beta}) = 0$ can be written in algebraic form as

$$\sum_{i=1}^{n} w_i x_{ij} \left( y_i - \sum_{j=1}^{p} x_{ij} \beta_j \right) = 0, \text{ for } j = 1, 2, \ldots, p \quad (*)$$

Differentiate (*) with respect to $\beta_j$ and set it to zero, we get

$$\sum_{i=1}^{n} f' \left( y_i - \sum_{j=1}^{p} x_{ij} \beta_j \right) x_{ij} = 0, \text{ for } j = 1, 2, \ldots, p$$

Let $u_i = \left( y_i - \sum_{j=1}^{p} x_{ij} \beta_j \right)$, and write

$$\sum_{i=1}^{n} \frac{f'(u_i)}{u_i} x_{ij} \left( y_i - \sum_{j=1}^{p} x_{ij} \beta_j \right) = \sum_{i=1}^{n} w(u_i) x_{ij} \left( y_i - \sum_{j=1}^{p} x_{ij} \beta_j \right) = 0, \text{ for } j = 1, 2, \ldots, p,$$

where $w(u) = \frac{f'(u)}{u}$. We find our choices of $\rho$ that:

1). LS: $w(u) = \frac{f'(u)}{u}$ is constant.
2). LAD: \( w(u) = \frac{1}{|u|} \).

3). Huber: \( w(u) = \begin{cases} \frac{1}{c} & \text{if } |u| \leq c \\ |u| & \text{otherwise} \end{cases} \)

**Ex 8.4.1 M-estimation**

```r
data(gala)
print(dim(gala))

# gala[1:3,]
Species Endemics Area Elevation Nearest Scruz Adjacent
Baltra 58 23 25.09 346 0.6 0.6 1.84
Bartolome 31 21 1.24 109 0.6 26.3 572.33
Caldwell 3 3 0.21 114 2.8 58.7 0.78

model1 <- lm(Species ~ Area+Elevation+Nearest+Scruz+Adjacent, gala)
summary(model1)

# To check AIC values for the 3 models
lapply(list(model1,model2,model3),AIC)
[[1]]
[1] 339.07
[[2]]
[1] 343.0158
[[3]]
[1] 326.6757
```

[2] LTS (Least Trimmed Squares)

**Ex 8.4.2 LTS**
library(MASS)
model4 <- ltsreg(Species ~ Area+Elevation+Nearest+Scruz+Adjacent, gala)
coef(model4)
(Intercept)        Area   Elevation     Nearest       Scruz    Adjacent
9.07854040  1.53475430  0.02366307  0.69826301 -0.08978973 -0.19718969

model5 <- ltsreg(Species ~ Area+Elevation+Nearest+Scruz+Adjacent, gala)
coef(model5)  #Each time, estimates are different. To avoid this, use ="exact".
(Intercept)        Area   Elevation     Nearest       Scruz    Adjacent
13.73855481  1.70787730 -0.02456592  1.01845584 -0.12825251  0.03807795

model4 <- ltsreg(Species ~ Area+Elevation+Nearest+Scruz+Adjacent, gala, nsamp="exact")
coef(model4)
(Intercept)        Area   Elevation     Nearest       Scruz    Adjacent
9.3814511  1.54365847  0.02412458  0.81110889 -0.11773219 -0.19792333

[2.1] Bootstrap

Ex 8.4.3 Bootstrap

model4 <- ltsreg(Species ~ Area+Elevation+Nearest+Scruz+Adjacent, gala)
sample(10, rep=T)
[1] 7 5 7 5 8 1 9 4 6 6

residuals(model4)[sample(30, rep=T)]
Coamano   Baltra    Onslow      Genovesa  Daphne.Major
166.1310986  0.2247260 -6.7686780  8.2059795  0.3615683
Daphne.Major Fernandina Marchena SanSalvador Champion
0.3615683  0.2247260 -162.7632162 -667.6758766  16.5050407
SantaCruz    Wolf  Espanola  Espanola  Gardner2
-968.3914288  4.7552796  0.2247260  0.2247260 -9.3414899
Rabida      Tortuga  Pinta    Baltra    Marchena
154.7763130  3.1807144  2.6637281  0.2247260  0.2247260
Pinta  Espanola  Pinta    Pinzon    Coamano
2.6637281  0.2247260  2.6637281  53.0793101  166.1310986
SantaFe    SantaMaria  Espanola  Darwin    Isabela
0.2247260  0.2247260  0.2247260  0.2247260 -6705.8807897

x <- model.matrix(~Area+Elevation+Nearest+Scruz+Adjacent, gala)[,-1]
bcoef <- matrix(0, 1000, 6)
for (i in 1:1000) {
    newy <- predict(model4) + residuals(model4)[sample(30, rep=T)]
    brg <- ltsreg(x, newy, nsamp="best")
    bcoef[i,] <- brg$coef}
quantile(bcoef[,2],c(0.025,0.975))
2.5%    97.5%
1.477025  1.609167

plot(density(bcoef[,2]),xlab="Coefficient of Area",main="Bootstrap Method")
abline(v=quantile(bcoef[,2],c(0.025,0.975)), lty=2, col="blue")

We conclude that there is a significant Area effect. That's different from the conclusion of Model 1. By the way, we get a similar conclusion from the "lm" fit if we exclude the island of Isabela. The reason why this case caught our attention is because of its large value of the Cook's distance, i.e., very influential point.

* First, here is the reason why "Isabela" is such an influential point.

model1 <- lm(Species ~ Area+Elevation+Nearest+Scruz+Adjacent, gala)
influencePlot(model1, id.method="identify")

StudRes   Hat  CookD
Isabela  -5.333694  0.9685321 8.250797
model5 <- lm(Species ~ Area+Elevation+Nearest+Scruz+Adjacent, gala, + subset=(row.names(gala) != "Isabela"))
summary(model5)

Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept) 22.58614   13.40191   1.685  0.10545
Area         0.29574    0.06186   4.781 8.04e-05 ***
Elevation    0.14039    0.04970   2.824  0.00961 **
Nearest     -0.25518    0.72168  -0.354  0.72686
Scruz       -0.09010    0.14980  -0.602  0.55339
Adjacent    -0.06503    0.01223  -5.318 2.12e-05 ***
---
Residual standard error: 41.65 on 23 degrees of freedom
Multiple R-squared:  0.8714, Adjusted R-squared:  0.8434
F-statistic: 31.17 on 5 and 23 DF,  p-value: 1.617e-09

Ex 8.4.4 lm, rlm, rq & ltsreg

data(star)
attach(star)
plot(light ~ temp, star)
model1 <- lm(light ~ temp)
model2 <- rlm(light ~ temp)
model3 <- ltsreg(light ~ temp, nsamp="exact")
model4 <- rq(light ~ temp)

abline(coef(model1))
abline(coef(model2), lty=2, col="red")
abline(coef(model3), lty=3, col="purple")
abline(coef(model4), lty=4, col="blue")
legend(locator(1),lty=1:4,col=c(1:2,"purple","blue"),c("lm","rlm","ltsreg","rq"))

Only "ltsreg" managed to exclude four points on the far left and capture the bulk of the data.