Chapter 7: Problems with the Error

Gunwoong Park

Lecture Note

University of Seoul

What can go wrong with the errors?

Recall we assumed $\epsilon \sim N(0, \sigma^2 I)$

- Unequal variance
- Correlated
- Heavy-tailed

Weighted Least Squares

Errors uncorrelated, but unequal variance, i.e.,

$$\epsilon \sim N(0, \sigma^2 W^{-1})$$

where

$$W^{-1} = diag(1/w_1, \ldots, 1/w_n)$$

Examples:

- Error variance proportional to the response: $w_i = y_i^{-1}$
- y_i is the average of n_i observations: $w_i = n_i$
- y_i is the sum of n_i observations: $w_i = n_i^{-1}$

Estimates

Transformation:

$$\begin{aligned} y_i &\rightarrow \sqrt{w_i} y_i \\ x_i &\rightarrow \sqrt{w_i} x_i \end{aligned}$$

Regress $\sqrt{w_i} y_i$ on $\sqrt{w_i} x_i$. Then

$$\begin{aligned} \hat{\beta} &= (X^T W X)^{-1} X^T W y \\ var(\hat{\beta}) &= (X^T W X)^{-1} \sigma^2 \end{aligned}$$

French Election Example

- French presidential election in 1981
- 10 candidates in the first round, top 2 in the second round
- Who do the votes go to in the second round?

```
> data(fpe)
> fpe
       EI   A   B   C   D  E  F  G  H  J  K  A2  B2  N
Ain   260  51  64  36  23  9  5  4  4  3  3 105 114 17
Alpes  75  14  17   9   9  3  1  2  1  1  1  32  31  5
... ...
## EI: total number of registered voters
## N: difference between 1st and 2nd round totals
```

French Election Model Selection

Model:

$$A2 = \beta_A A + \beta_B B + \beta_C C + \beta_D D + \beta_E E + \beta_F F + \beta_G G + \beta_H H + \beta_J J + \beta_K K + \beta_N N + \epsilon$$

where β_i represents the proportion of votes transferred from candidate i to A_2.

- Constraint

- $0 < \beta_i < 1$ for all i.

French Election Model Selection

```
##Fit a linear model with no intercept
> g <- lm(A2 ~ A+B+C+D+E+F+G+H+J+K+N-1,
    data=fpe, weights=1/EI)
> round(g$coef, 3)
     A      B      C      D      E      F      G
 1.067 -0.105  0.246  0.926  0.249  0.755  1.972
     H      J      K      N
 -0.566  0.612  1.211  0.529

> lm(A2 ~ A+B+C+D+E+F+G+H+J+K+N-1,data=fpe)$coef
     A      B      C      D      E      F      G
 1.075 -0.125  0.257  0.905  0.671  0.783  2.166
     H      J      K      N
-0.854  0.144  0.518  0.558
```

French Election Model Selection

```
## Remove coefficients less than 0
## Set coefficients bigger than 1 to 1
> lm(A2 ~ offset(A+G+K)+C+D+E+F+J+N-1, data=fpe,
        weights=1/EI)$coef
     C       D       E       F      J       N
  0.228   0.970   0.426   0.751  -0.177   0.615

# Now drop J
lm(A2 ~ offset(A+G+K)+C+D+E+F+N-1, data=fpe,
        weights=1/EI)$coef
    C      D     E      F      N
0.226  0.970   0.390  0.744  0.609
```

There exists a package 'mgcv' which automatically enforce all coefficients falling into $[0, 1]$.

- See an example in page 118.

Issue of Finding Weights

In most cases, finding weights is not easy.

- $\omega_i \propto n_i$
- $\omega_i \propto \frac{1}{n_i}$
- $\omega_i \propto x_i$
- $\omega_i \propto \gamma_0 + x_i^{\gamma_1}$

Cars Example

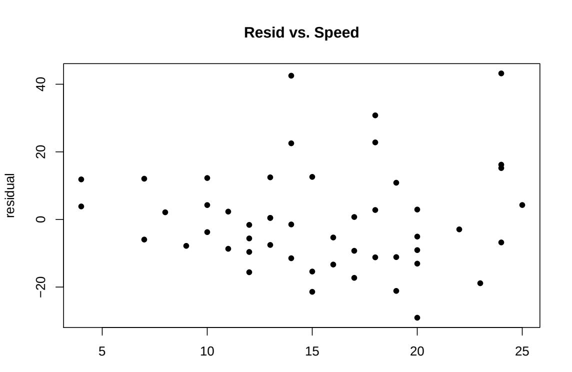
Speed and Stopping Distances of Cars

The data gives the speed of cars and the distances taken to stop.

A data frame with 50 observations on 2 variables.

- speed Speed (mph)

- dist Stopping distance (ft)



Cars Example

```
> require(nlme)
> wlmod = gls(dist~ speed, data = cars,
weight = varConstPower(1, form = ~speed))
> summary(wlmod)

Variance function:
Structure: Constant plus power of variance covariate
Formula: ~speed
Parameter estimates:
const    power
3.160444 1.022368

Coefficients:
Value Std.Error   t-value p-value
(Intercept) -11.085378  4.052378 -2.735524  0.0087
speed         3.484162  0.320237 10.879947  0.0000

Correlation:
(Intr)
speed -0.9
```

Generalized Least Squares (GLS)

In general

$$\epsilon \sim N(0, \sigma^2 \Sigma)$$

Write

$$\Sigma = SS^T$$

where S is a lower triangular matrix (the Cholesky decomposition).
Transformation:

$$\begin{aligned} y &\rightarrow S^{-1}y \\ x &\rightarrow S^{-1}x \end{aligned}$$

Estimates:

$$\begin{aligned} \hat{\beta} &= (X^T \Sigma^{-1} X)^{-1} X^T \Sigma^{-1} y \\ var(\hat{\beta}) &= (X^T \Sigma^{-1} X)^{-1} \sigma^2 \end{aligned}$$

Employment Example

- Employment data from 1947 to 1962
- Response: number of people employed (yearly)
- Predictors: gross national product and population over 14

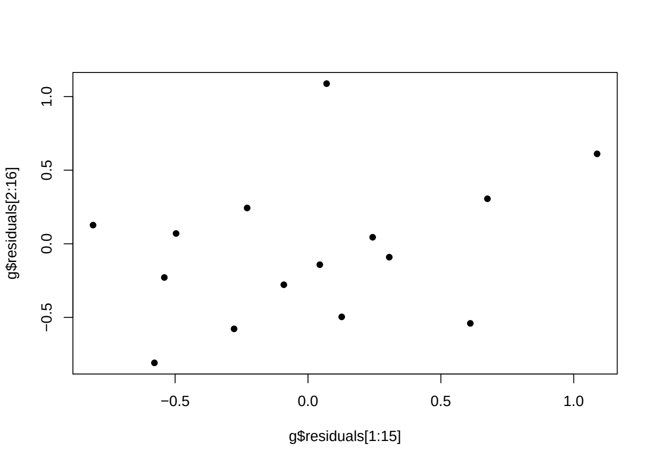
- Data collected over time: errors could be correlated
- One of the simplest correlation structures over time: the autoregressive model – here AR(1):

$$\epsilon_{i+1} = \phi\epsilon_i + \delta_i$$

where δ_i are i.i.d. $N(0, \tau^2)$.

Employment Example

```
> data(longley)
> g <- lm(Employed ~ GNP + Population, longley)
> summary(g)
Coefficients:
            Estimate Std.Error t value Pr(>|t|)
(Intercept) 88.93880  13.78503   6.452 2.16e-05
GNP          0.06317   0.01065   5.933 4.96e-05
Population  -0.40974   0.15214  -2.693   0.0184

Residual standard error:   0.5459 on 13 degrees of freedom
Multiple R-Squared: 0.9791      Adjusted R-squared: 0.9758
F-statistic: 303.9 on 2 and 13 DF       p-value: 1.221e-11
```


g$residuals[2:16]
1.0
0.5
0.0
−0.5
−0.5
0.0
0.5
1.0
g$residuals[1:15]

```
## Fit GLS with AR(1) structure
> library(nlme)
> g <- gls(Employed ~ GNP + Population,
   correlation=corAR1(form=~Year), data=longley)
> summary(g)
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  <.0001
GNP         0.07207  0.010606  6.795485  <.0001
Population -0.54851  0.154130 -3.558778  0.0035

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

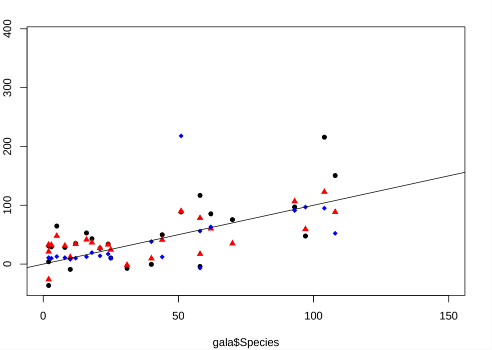
ANOVA Test

Lack of Fit: ANOVA test

How well does a model fit the data?

- If the model is correct, then $\hat{\sigma} \approx \sigma$.
- Otherwise, $\hat{\sigma} \gg \sigma$.

ANOVA test



Simulation Study: Polynomial Model

```
> lm_red = lm(y ~ x)
> lm_full =lm(y ~ x + I(x^2))
> anova(lm_red,lm_full)

Analysis of Variance Table
Model 1: y ~ x
Model 2: y ~ x + I(x^2)
Res.Df     RSS Df Sum of Sq       F    Pr(>F)
1     98 2273759
2     97      91  1   2273668 2426999 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> lm_red = lm(y ~ x + I(x^2))
> lm_full =lm(y ~ x + I(x^2) + I(x^3))
> anova(lm_red,lm_full)
Analysis of Variance Table

Model 1: y ~ x + I(x^2)
Model 2: y ~ x + I(x^2) + I(x^3)
Res.Df    RSS Df Sum of Sq     F Pr(>F)
1     97 90.872
2     96 88.859  1    2.0132 2.175 0.1435
```

Simulation Study: Polynomial Model

```
Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept)   99.614     15.411   6.464 3.99e-09 ***
x             -3.986      1.498  -2.661  0.00911 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 152.3 on 98 degrees of freedom
Multiple R-squared:  0.06737,Adjusted R-squared:  0.05786
F-statistic:  7.08 on 1 and 98 DF,  p-value: 0.009111

Coefficients:
Estimate Std. Error  t value Pr(>|t|)
(Intercept) -0.0288700  0.1169616   -0.247    0.806
x            0.0104287  0.0098592    1.058    0.293
I(x^2)       1.0006549  0.0006423 1557.883   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9679 on 97 degrees of freedom
Multiple R-squared:      1,Adjusted R-squared:      1
F-statistic: 1.301e+06 on 2 and 97 DF,  p-value: < 2.2e-16
```

Simulation Study

```
Coefficients:
Estimate Std. Error  t value Pr(>|t|)
(Intercept) -7.916e-02  1.212e-01   -0.653    0.515
x           -9.183e-03  1.652e-02   -0.556    0.580
I(x^2)       1.001e+00  7.363e-04 1359.836   <2e-16 ***
I(x^3)       6.495e-05  4.404e-05    1.475    0.144
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9621 on 96 degrees of freedom
Multiple R-squared:      1,Adjusted R-squared:      1
F-statistic: 8.78e+05 on 3 and 96 DF,  p-value: < 2.2e-16
```

Robust Regression

Main concern: heavy-tailed error distribution

- M-estimation
- Least trimmed squares

M-estimation

Definition:

- Find β to minimize

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

$L(\cdot)$ is called the loss function.

Possible loss functions:

- $L(z) = z^2$ least squares (LS)

$$\beta = \arg\min \sum_{i=1}^{n} (y_i - x_i^T \beta)^2$$

- $L(z) =$ negative log likelihood.

M-estimation Continued

Possible loss functions:

- least absolute deviations (LAD), $L(z) = |z|$

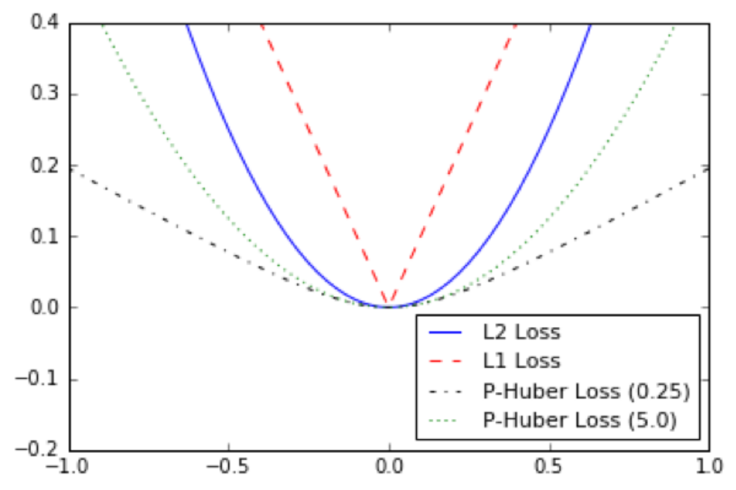
$$\beta = \arg\min \sum_{i=1}^{n} |y_i - x_i^T \beta|$$

- Huber's method

$$L(z) = \begin{cases} z^2/2 & \text{if } |z| \le c \\ c|z| - c^2/2 & \text{otherwise} \end{cases}$$

c should be a robust estimate of σ, e.g., the median of $|\hat{\epsilon}_i|$.

LS vs LAD vs Huber



Gala Example

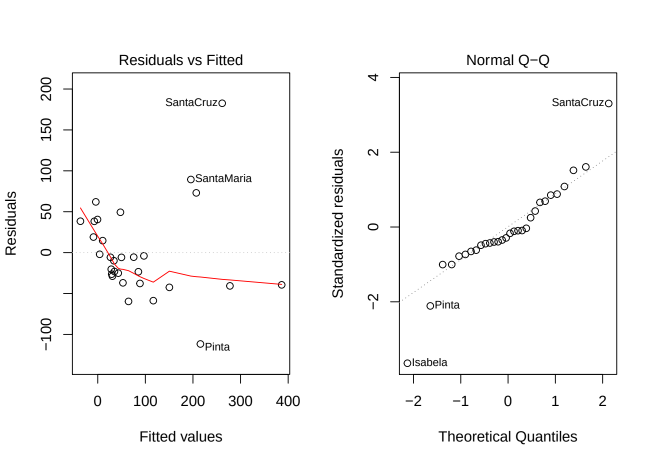
Recall from Ch. 2: Number of species of tortoise on the various Galapagos islands

- Response: number of species of tortoise
- Predictors: number of endemic species, area of the island, highest elevation of the island, distance from the nearest island, distance from Santa Cruz Island, area of the adjacent island

Gala Example: LSE

```
> data(gala)
## Least squares
> g <- lm(Species ~ Area + Elevation + Nearest
    + Scruz + Adjacent, data=gala)
> summary(g)
Coefficients:
            Estimate Std.Error t value Pr(>|t|)
(Intercept) 7.068221 19.154198   0.369 0.715351
Area       -0.023938  0.022422  -1.068 0.296318
Elevation   0.319465  0.053663   5.953 3.82e-06
Nearest     0.009144  1.054136   0.009 0.993151
Scruz      -0.240524  0.215402  -1.117 0.275208
Adjacent   -0.074805  0.017700  -4.226 0.000297

Residual standard error:   60.98 on 24 degrees of freedom
Multiple R-Squared: 0.7658     Adjusted R-squared: 0.7171
F-statistic:  15.7 on 5 and 24 DF      p-value: 6.838e-07
```

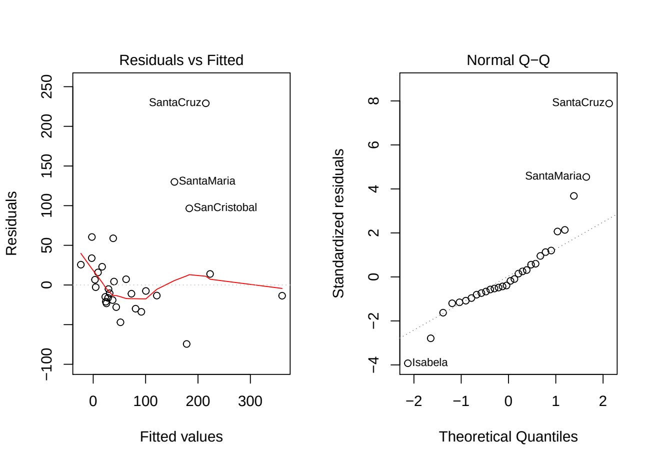
Gala Example: LSE Diagnostics



Gala Example: Huber

```
## Huber's method
> library(MASS)
> ghuber <- rlm(Species ~ Area + Elevation + Nearest
    + Scruz + Adjacent, data=gala)
> summary(ghuber)
Coefficients:
            Value   Std.Error t value
(Intercept)  6.3611 12.3897    0.5134
Area        -0.0061  0.0145   -0.4214
Elevation    0.2476  0.0347    7.1320
Nearest      0.3592  0.6819    0.5267
Scruz       -0.1952  0.1393   -1.4013
Adjacent    -0.0546  0.0114   -4.7648
Residual standard error:  29.73 on 24 degrees of freedom
```

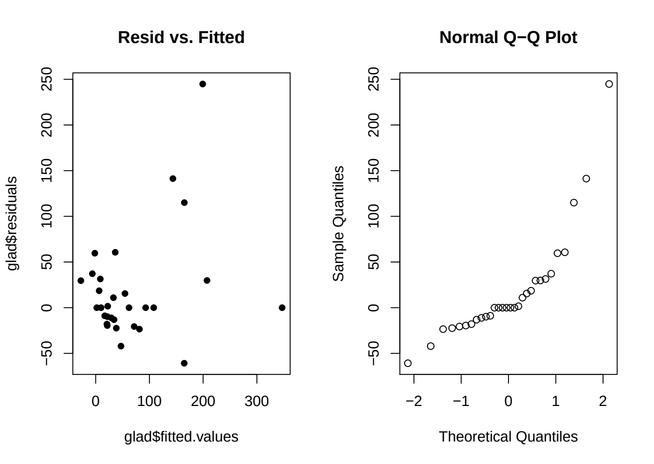
Gala Example: Huber Diagnostics



Gala Example: Least Absolute Deviations

```
## Least absolute deviations
> library(quantreg)
> glad <- rq(Species ~ Area + Elevation + Nearest
    + Scruz + Adjacent, data=gala)
> summary(glad)
Coefficients:
            coefficients lower bd  upper bd
(Intercept)   1.31445    -19.87777  24.37411
Area         -0.00306     -0.03185   0.52800
Elevation     0.23211      0.12453   0.50196
Nearest       0.16366     -3.16339   2.98896
Scruz        -0.12314     -0.47987   0.13476
Adjacent     -0.05185     -0.10458   0.01739
```

Gala Example: LAD Diagnostics



Least Trimmed Squares (LTS)

LTS is a robust statistical method that fits a function to a set of data whilst not being unduly affected by the presence of outliers.

Minimize:

$$\sum_{i=1}^{m} \hat{\epsilon}_{(i)}^2$$

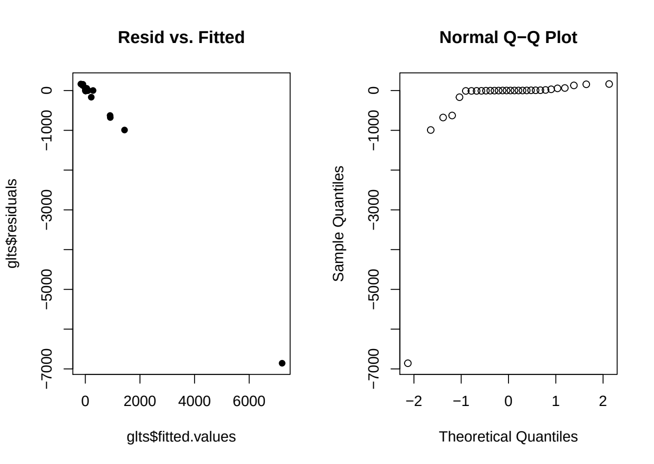
where $m < n$ and (i) indicates sorting.
Default m: $\lfloor n/2 \rfloor + \lfloor (p+1)/2 \rfloor$

– ignores largest residuals

Gala Example: LTS

```
## Least trimmed squares
> library(MASS)
> glts <- ltsreg(Species ~ Area + Elevation +
       Nearest + Scruz + Adjacent, data=gala)
> round(glts$coef, 3)
(Intercept)   Area   Elevation  Nearest   Scruz  Adjacent
  8.975      1.544    0.024      0.803   -0.117   -0.196
## Another try
> glts <- ltsreg(Species ~ Area + Elevation +
      Nearest + Scruz + Adjacent, data=gala)
> round(glts$coef, 3)
(Intercept)   Area   Elevation  Nearest   Scruz  Adjacent
  9.321      1.512    0.032      0.559   -0.091   -0.196

  ## Exact solution - takes longer
  > glts <- ltsreg(Species ~ Area + Elevation +
  Nearest + Scruz + Adjacent, data=gala, nsamp = "exact")

  > glts$coef
  (Intercept)        Area   Elevation     Nearest
  9.38114511    1.54365847  0.02412458  0.81110889
  Scruz    Adjacent
  -0.11773219 -0.19792333
```

Gala Example: LTS Diagnostics



Bootstrap

- We don't have the standard errors for the LTS regression coefficients.
- When we have no theory to compute SEs, can use **bootstrap**
- Fundamental idea: pretend the observed data is the population
- Resample observed data, create multiple samples
- From each sample, estimate parameters and assess variability

Bootstrap Continued

Simulation world:

- Generate ϵ from the known error distribution
- Form $y = X\beta + \epsilon$ from the known β
- Compute $\hat{\beta}$

– useful for testing new methodology

Bootstrap world:

- Sampling with replacement from $\hat{\epsilon}_1, \ldots, \hat{\epsilon}_n \Rightarrow \epsilon^*$
- Form $y^* = X\hat{\beta} + \epsilon^*$
- Compute $\hat{\beta}^*$ from (X, y^*)

– useful for assessing estimator uncertainty on real data when no theory is available

Gala Example: Inference

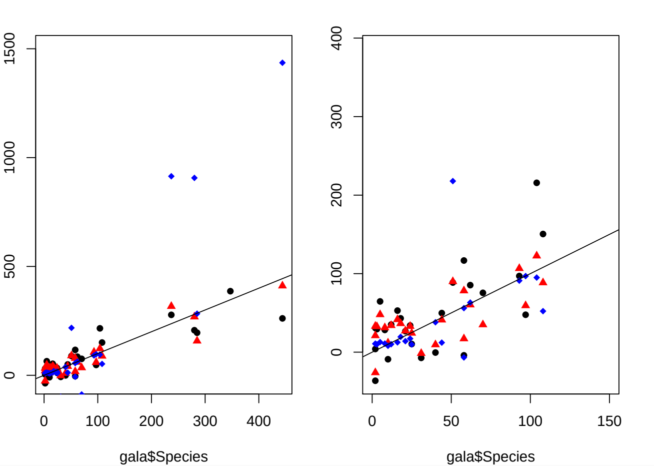
```
# extract matrix of predictors for ltsreg
> x <- gala[,3:7]
## bootstrap 1000 times
> bcoef <- matrix(0, nrow=1000, ncol=6)
> for (i in 1:1000) {
+   newy <-  glts$fit + glts$resid[sample(30, rep=T)]
+   bcoef[i,] <- ltsreg(x, newy, nsamp="best")$coef
+ }

## 95% C.I. for Area
> quantile(bcoef[,2], c(0.025, 0.975))
    2.5%    97.5%
1.486674 1.689529
```

Gala Example

```
> g <- lm(Species ~ Area + Elevation + Nearest + Scruz + Adjacent, data=gala)
> which.max( cooks.distance(g) )
Isabela
16

## LS model w/o Isabela (the most influential point)
> gi <- lm(formula(g), data=gala,
subset=(row.names(gala) != 'Isabela'))
> gi$coef
(Intercept)        Area   Elevation     Nearest       Scruz    Adjacent
22.58614473  0.29574351  0.14039023 -0.25518223 -0.09010457 -0.06503051

> g$coef
(Intercept)         Area    Elevation      Nearest        Scruz     Adjacent
7.068220709 -0.023938338  0.319464761  0.009143961 -0.240524230 -0.074804832

> glts$coef
(Intercept)         Area    Elevation      Nearest        Scruz     Adjacent
13.492574616  1.563886712  0.009783274  0.697117983 -0.116777344 -0.197276535
```

Gala Example

```
plot(gala$Species, g$fitted.values, pch = 16, ylim = c(-25,1500))
points(gala$Species[-16], gi$fitted.values, pch = 17, col="red")
points(gala$Species,glts$fitted.values, pch = 18, col="blue")
```



Remarks

- Two routes to the same goal:
  - ▷ Regression diagnostics in conjunction with LS
  - ▷ Robust methods

  Former more informative, but time-consuming; latter quick and suitable for large datasets.
- *M*-estimation failed to identify "Isabela”

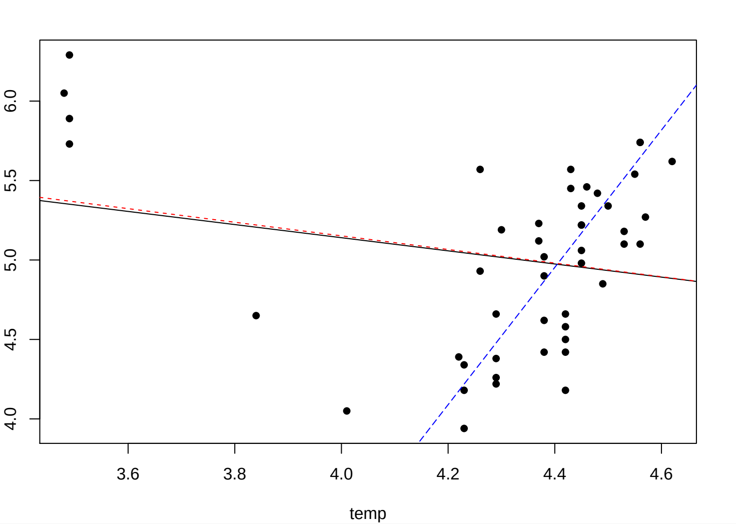
All models are **wrong**!

Star Example

- 47 stars in the star cluster CYG OB1
- Response: log of the light intensity
- Predictor: log of the surface temperature

```
## Compare LS, Huber and LTS
> data(star)
> plot(light ~ temp, data=star, xlab="temp", ylab="light")
> starls <- lm(light ~ temp, star)
> abline(starls$coef)
> starhuber <- rlm(light ~ temp, star)
> abline(starhuber$coef, lty=2)
> starlts <- ltsreg(light ~ temp, star, nsamp="exact")
> abline(starlts$coef, lty=5)
```

Star Example: LS vs Huber vs LTS



Summary: Robust methods

- Protect against outliers and heavy tails... but not misspecified structure (model or error)
- Theory not available for standard errors – need bootstrap
- If robust and LS fits are very different, cause to worry
- Useful when automatic fitting is needed (no human intervention)