Chapter 8: Problems with Errors

Stats 500, Fall 2015
Brian Thelen, University of Michigan
443 West Hall, bjthelen@umich.edu

Problems with the Error

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

- Unequal variance
- Correlated
- Heavy-tailed

Weighted Least Squares

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

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

where

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

Examples:

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

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

$$\begin{aligned} \hat{\beta} &= (X^T \mathbf{W}\ X)^{-1} X^T \mathbf{W}\ y \\ var(\hat{\beta}) &= (X^T \mathbf{W}\ X)^{-1} \sigma^2 \\ \hat{\sigma}^2 &= \frac{\hat{\epsilon}^T \mathbf{W}\ \hat{\epsilon}}{n - (p+1)} \end{aligned}$$

French Election Example

- French presidential election in 1981
- 10 candidates in 1st round – top 2 in the 2nd round
- Who do the votes go to in the second round?
- Data: (vote totals are in thousands)
  - A – Voters for Mitterand in the first round
  - B – Voters for Giscard in the first round
  - C – Voters for Chirac in the first round
  - $\vdots$
  - K – Voters for party K in the first round
  - A2 – Voters for Mitterand in the second round
  - B2 – Voters for party Giscard in the second round

French Election Example - cont'd

```
> 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
... ...
Vendee 336   61  105   59   19  10  11   6   5   4   3 115 176   8
Yonne  216   44   52   31   24   7   4   4   3   3   2  91  91   8
>
## EI: total number of registered voters
## N: difference between 1st and 2nd round totals
```

```
##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
```

```
## 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
```

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:

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

Generalized Least Squares Continued

Estimates:

$$\begin{aligned}
\hat{\beta} &= (X^T \boldsymbol{\Sigma}^{-1} X)^{-1} X^T \boldsymbol{\Sigma}^{-1} y \\
var(\hat{\beta}) &= (X^T \boldsymbol{\Sigma}^{-1} X)^{-1} \sigma^2 \\
\hat{\sigma}^2 &= \frac{\hat{\epsilon}^T \boldsymbol{\Sigma}^{-1} \hat{\epsilon}}{n - (p+1)}
\end{aligned}$$

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):

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

where δ_i are i.i.d. $N(0, \tau^2)$. This gives

$$cor(\epsilon_i, \epsilon_j) = \rho^{|i-j|}.$$

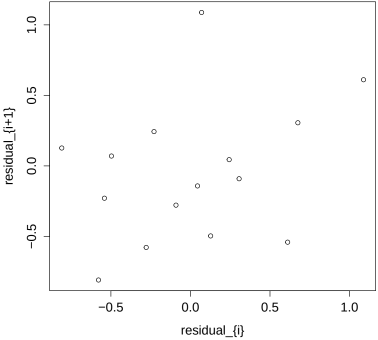
Employment Example

```
> 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
```

- Scatter plot of $\hat{\epsilon}_{i+1}$ vs. $\hat{\epsilon}_i$
- Estimation of correlation (e.g., AR(1) model)

```
## Simple autoregressive correlation structure
## Simple residual scatter plot
plot(g$res[-16],g$res[-1],xlab='residual_{i}',
   + ylab='residual_{i+1}')
## Estimate rho
> rho <- cor(g$res[-1], g$res[-16])
> rho
[1] 0.3104092
> x <- model.matrix(g)
## Compute the correlation matrix
> Sigma <- diag(16)
> Sigma <- rho^abs(row(Sigma) - col(Sigma))
> Sigi <- solve(Sigma)
> xtxi <- solve(t(x) %*% Sigi %*% x)
> beta <- xtxi %*% t(x) %*% Sigi %*% longley$Empl
```




```
> beta
                    [,1]
  (Intercept) 94.8988949
  GNP          0.0673895
  Population  -0.4742741
## Compute the residuals
> res <- longley$Empl - x %*% beta
## rho is changed!
> cor(res[-1], res[-16])
[1] 0.3564162
```

```
## 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
```

```
> intervals(g)
Approximate 95% confidence intervals
 Coefficients:
                  lower          est.        upper
(Intercept) 71.18320440 101.85813280 132.5330612
GNP          0.04915865   0.07207088   0.0949831
Population  -0.88149053  -0.54851350  -0.2155365
 Correlation structure:
         lower      est.     upper
Phi -0.4430373 0.6441692 0.9644866
```

Robust Regression

Main concern: **heavy-tailed** error distribution

1. M-estimation
2. Least trimmed squares

M-estimation

Find β to minimize

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

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

M-estimation Continued

Possible loss functions:

- $L(z) = z^2$ least squares (**LS**)
- $L(z) = |z|$ least absolute deviations (**LAD**)
- **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 $|\hat{\epsilon}_i|$.

Gala Example

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

- 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

```
> 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
```

```
## 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
```

```
## 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
```

Least Trimmed Squares (LTS)

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

```
## Least trimmed squares
> library(MASS)
> glts <- ltsreg(Species ~ Area + Elevation +
        Nearest + Scruz + Adjacent, data=gala)
> round(glts$coef, 3)
(Intercept)    Area   Elevation  Nearest   Scruz  Adjacent
  8.975       1.544    0.024       0.803   -0.117   -0.196
## Another try with set seed
> set.seed(123)
> glts <- ltsreg(Species ~ Area + Elevation +
+                  Nearest + Scruz + Adjacent, data=gala)
> round(glts$coef, 3)
(Intercept)     Area   Elevation  Nearest  Scruz  Adjacent
     12.507    1.545    0.017       0.523   -0.094   -0.143
## Exact solution - takes longer
> glts <- ltsreg(Species ~ Area + Elevation +
       Nearest + Scruz + Adjacent, data=gala, nsamp="exact")
> round(glts$coef, 3)
(Intercept) Area   Elevation  Nearest   Scruz  Adjacent
   9.381    1.544    0.024      0.811    -0.118   -0.198
```

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 Continued

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^*)

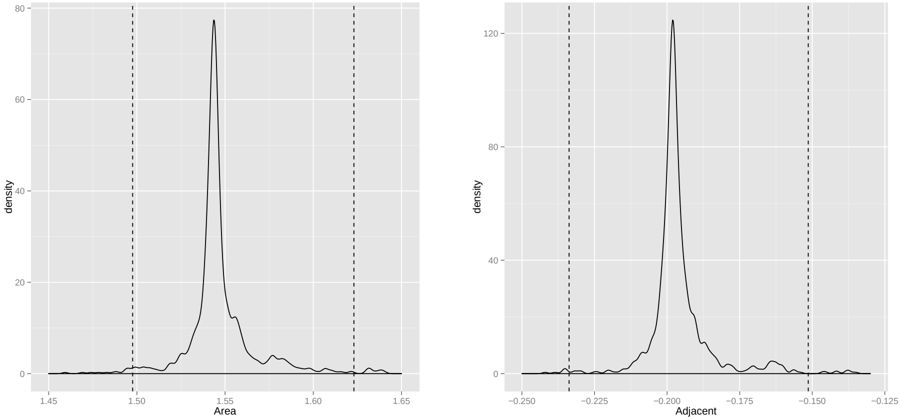
– useful for assessing estimator uncertainty on real data when no theory is available

Gala Example

```
# 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 Parameters
> colnames(bcoef) = names(coef(glts))
> apply(bcoef,2,function(x),quantile(x,c(0.025,0.975)))
Error: unexpected ',' in "apply(bcoef,2,function(x),"
> apply(bcoef,2,function(x) quantile(x,c(0.025,0.975)))
      (Intercept)     Area  Elevation   Nearest       Scruz    A
2.5%     1.917772 1.494069 -0.01461920 0.1588385 -0.26238063 -0.
97.5%   21.467200 1.606935  0.07333018 1.9147331  0.09821907 -0.
```

So which fit is better?

Histogram of bootstrap estimates



```
> library(ggplot2)
> bcoef <- data.frame(bcoef)
> p1 <- ggplot(bcoef, aes(x = Area)) + geom_density()
+                                 + xlim(1.45,1.65)
> p1 + geom_vline(xintercept=c(1.4976, 1.6230),
+                              linetype="dashed")
> p2 <- ggplot(bcoef, aes(x = Adjacent)) + geom_density()
+                                      + xlim(-0.25, -0.13)
> p2 + geom_vline(xintercept=c(-0.23375, -0.15138),
+                                     linetype="dashed")
```

```
## LS model w/o Isabela (the most influential point)
> gi <- lm(formula(g), data=gala,
           subset=(row.names(gala) != 'Isabela'))
> summary(gi)
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
```

Remarks

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

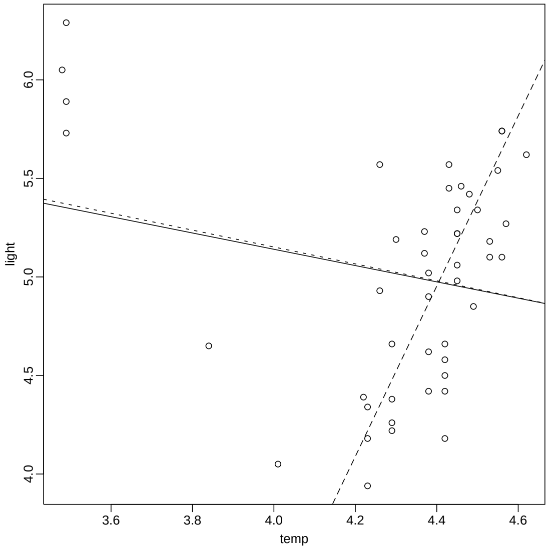
  Former more informative, but time-consuming; latter quick and suitable for large datasets.
- M-estimation failed to identify "Isabela"

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 Continued



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)