Generalized Least Squares
Iteratively Reweighted Least Squares

Daero Kim

Outline

- The general case/ the correlated case
- Generalized Least Squares
- Weighted Least Squares
- Iteratively Reweighted Least Squares
- Summary

The general case

- $\mathrm{y} = \mathrm{X}\beta + \epsilon$ $\qquad$ $\mathrm{var}\,\epsilon = \sigma^2 I$

where

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} \mathbf{x}_1^{\mathrm{T}} \\ \mathbf{x}_2^{\mathrm{T}} \\ \vdots \\ \mathbf{x}_n^{\mathrm{T}} \end{pmatrix} = \begin{pmatrix} x_{11} & \cdots & x_{1p} \\ x_{21} & \cdots & x_{2p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{np} \end{pmatrix}, \quad \boldsymbol{\beta} = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_p \end{pmatrix}, \quad \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}$$

- We have learned that the errors can have non-constant variance or may be correlated.
- Consider variance is correlated
- $\mathrm{var}\,\epsilon = \sigma^2 \Sigma$
- assume σ^2 is unknown, and Σ *is known*

Generalized Least Squares

- Generalized least squares minimizes
- $(y - X\beta)^T \Sigma^{-1} (y - X\beta)$
- Which is solved by
- $\hat{\beta} = (X^T \Sigma^{-1} X)^{-1} X^T \Sigma^{-1} y$
- let $\Sigma = SS^T$
- Where S is a triangular matrix using the Choleski Decomposition

Generalized Least Squares

- We have
- $(y - X\beta)^T S^{-T} S^{-1} (y - X\beta)$
- $= (S^{-1}y - S^{-1}X\beta)^T (S^{-1}y - S^{-1}X\beta)$
- So it is like regressing $S^{-1}X$ on $S^{-1}y$
- $S^{-1}y = S^{-1}X\beta + S^{-1}\epsilon$
- $y' = X'\beta + \epsilon'$

Generalized Least Squares

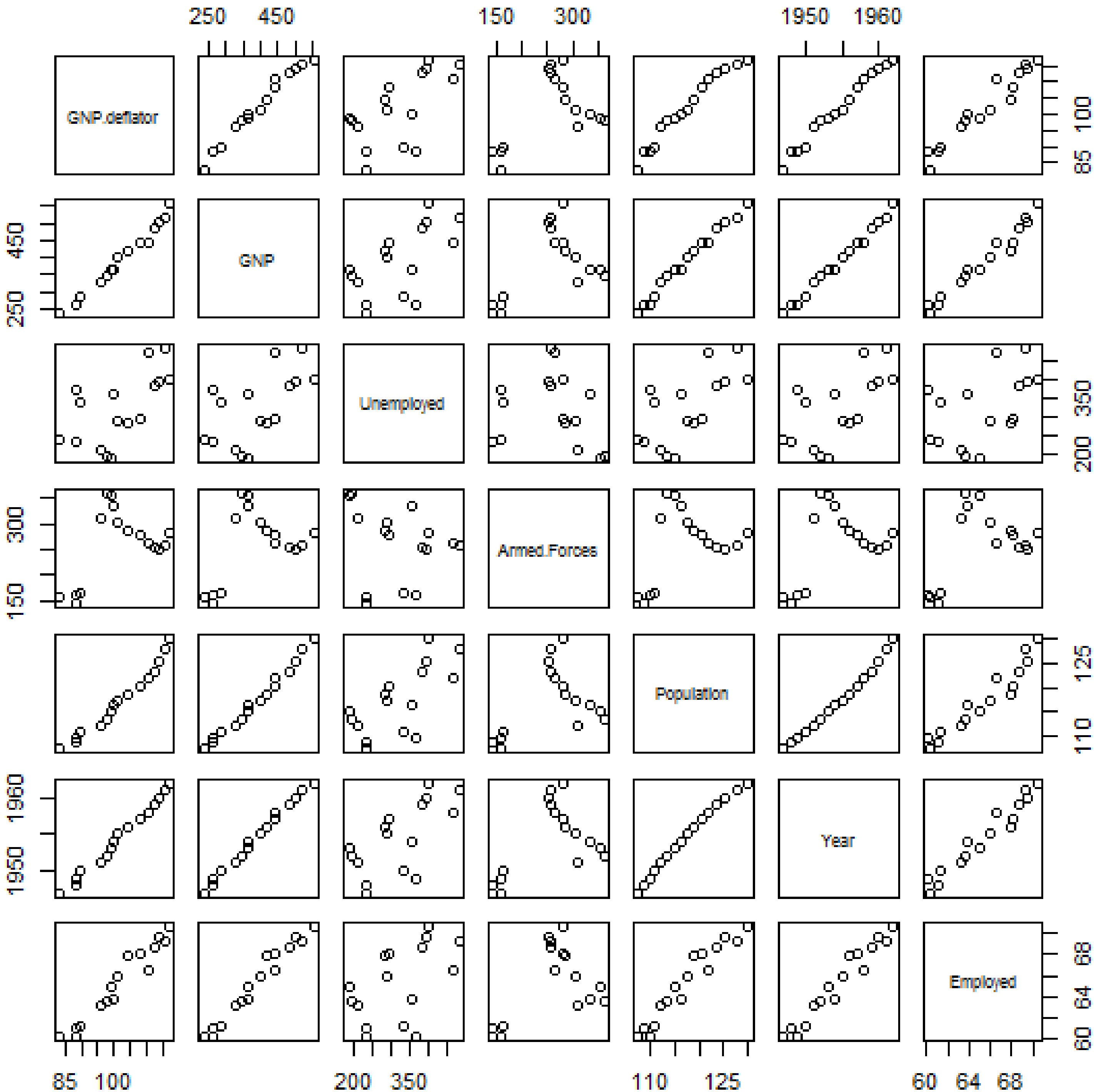
- $y' = X'\beta + \epsilon'$
- Examine the variance of the new errors, ϵ'
- $\mathrm{var}\ \epsilon' = \mathrm{var}(S^{-1}\epsilon) = S^{-1}(\mathrm{var}\epsilon)S^{-T}$

  $= S^{-1}\sigma^2 S S^T S^{-T} = \sigma^2 I$
- new variables y' and X' have uncorrelated errors with equal variance
- $\mathrm{var}\ \beta = (X^T \Sigma^{-1} X)^{-1}\ \sigma^2$

To illustrate this

- Longley's regression data was used where the response is **number of people employed**, yearly from 1947 to 1962 and the predictors are GNP implicit price deflator (1954=100), GNP, unemployed, armed forces, population 14 years of age and over, and year. well-known example for a highly collinear regression
- The data originally appeared in Longley (1967)

  J. W. Longley (1967) An appraisal of least-squares programs from the point of view of the user. *Journal of the American Statistical Association*, **62**, 819–841.

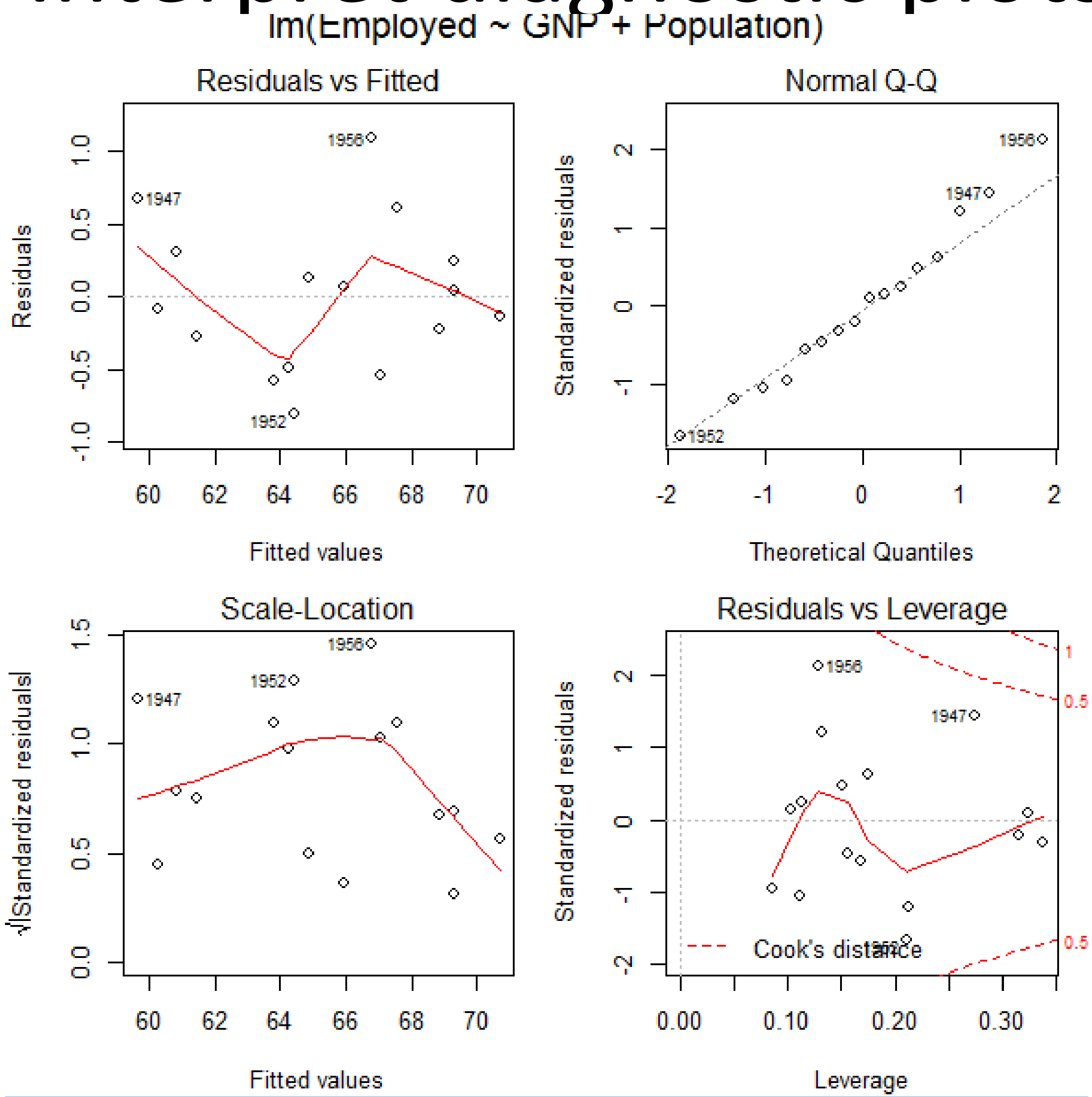


GNP.deflator
GNP
Unemployed
Armed.Forces
Population
Year
Employed
250
450
150
300
1950
1960
85
100
200
350
110
125
60
64
68

Do the simple regression

- > summary(fm1 <- lm(Employed ~ GNP+Population, data = longley))
- Call:
- lm(formula = Employed ~ GNP + Population, data = longley)
- Residuals:
-     Min   1Q   Median   3Q   Max
- -0.80899 -0.33282 -0.02329 0.25895 1.08800
- 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

Interpret diagnostic plots



GLS fitting

- The nlme library contains a GLS fitting function. We can use it to fit this model:

```
> library(nlme)
> g<-gls(Employed~GNP+Population,
correlation=corAR1(form=~Year),data=longley)
> summary(g)
```

  Generalized least squares fit by REML
- REML can produce unbiased estimates of variance and covariance parameters
- The idea underlying REML estimation was put forward by M. S. Bartlett in 1937

GLS fitting Result

- Model: Employed ~ GNP + Population
  Data: longley

AIC	BIC	logLik
44.66377	47.48852	-17.33188

  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

GLS fitting Result Vs simple lm

- We see that the estimated value of correlation is 0.6441692. However, if we check the 95% confidence intervals for this:

  Correlation structure:

	lower	est.	upper
Phi	-0.44239	0.6441692	0.9644304

> anova(g,fm1)

```
> anova(g,fm1)
    Model df      AIC      BIC    logLik   Test  L.Ratio p-value
g       1  5 44.66377 47.48852 -17.33188
fm1     2  4 46.38923 48.64903 -19.19462 1 vs 2 3.725466  0.0536
```

Weighted least squares

- Sometimes the errors are uncorrelated, but may have unequal variance where the form of the inequality is known.
- Weighted least squares (WLS) can be used in this situation.
  When Σ is diagonal, the errors are uncorrelated but do not necessarily have equal variance.

Weighted least squares

- We can write $\Sigma = \mathrm{diag}(1/w_1, \ldots, 1/w_n)$
- Where the w are the weights
- so $S = \mathrm{diag}(\sqrt{(1/w_1)}, \ldots, \sqrt{(1/w_n)})$
- Again regress $S^{-1}X$ on $S^{-1}y$
- Some example cases
  - Errors proportional to a predictor:
  - var $\epsilon_i \propto x_i$ suggests $w_i = x_i^{-1}$
  - *Or* number of observations
  - Y_i are the average of n_i observations then
  - var y_i = var $\epsilon_i = \sigma^2/n_i$ suggests $w_i = n_i$

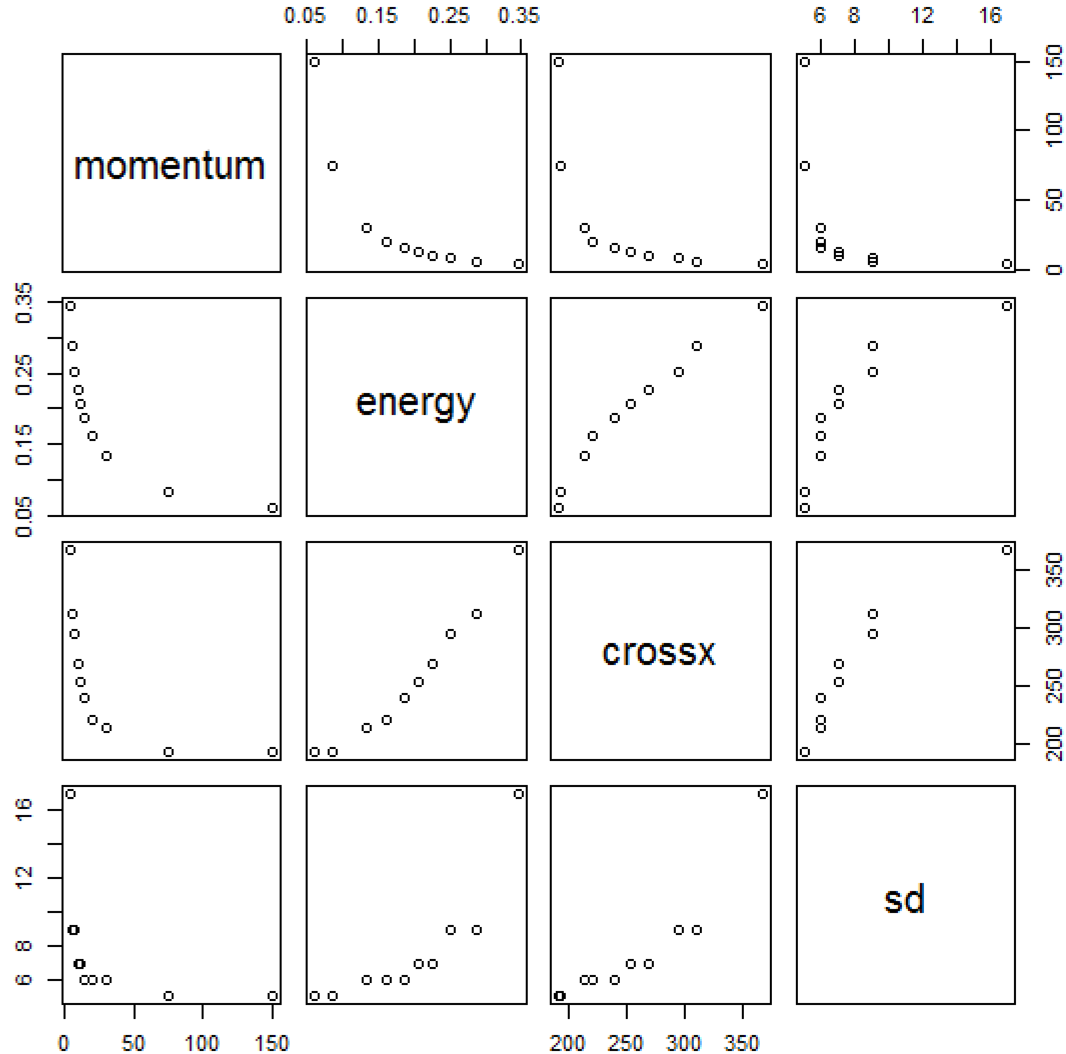
To illustrate this

- We will be using another data from a package called faraway in this example:
- Data obtained from an experiment to study the interaction of certain kinds of elementary particles on collision with proton targets. The experiment was designed to test certain theories about the nature of the strong interaction.
- The cross-section(crossx) variable is believed to be linearly related to the inverse of the energy

  (energy has already been inverted).

Energy and cross-section

```
> data(strongx)
> strongx
   momentum energy crossx sd
1         4  0.345    367 17
2         6  0.287    311  9
3         8  0.251    295  9
4        10  0.225    268  7
5        12  0.207    253  7
6        15  0.186    239  6
7        20  0.161    220  6
8        30  0.132    213  6
9        75  0.084    193  5
10      150  0.060    192  5
```



Fit the regression without weights

```
> gu<- lm(crossx~energy, strongx)
> summary(gu)

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

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
```

Fit the model with weights

```
>  g<-lm(crossx~energy,strongx,weights=sd^-2)
> summary(g)

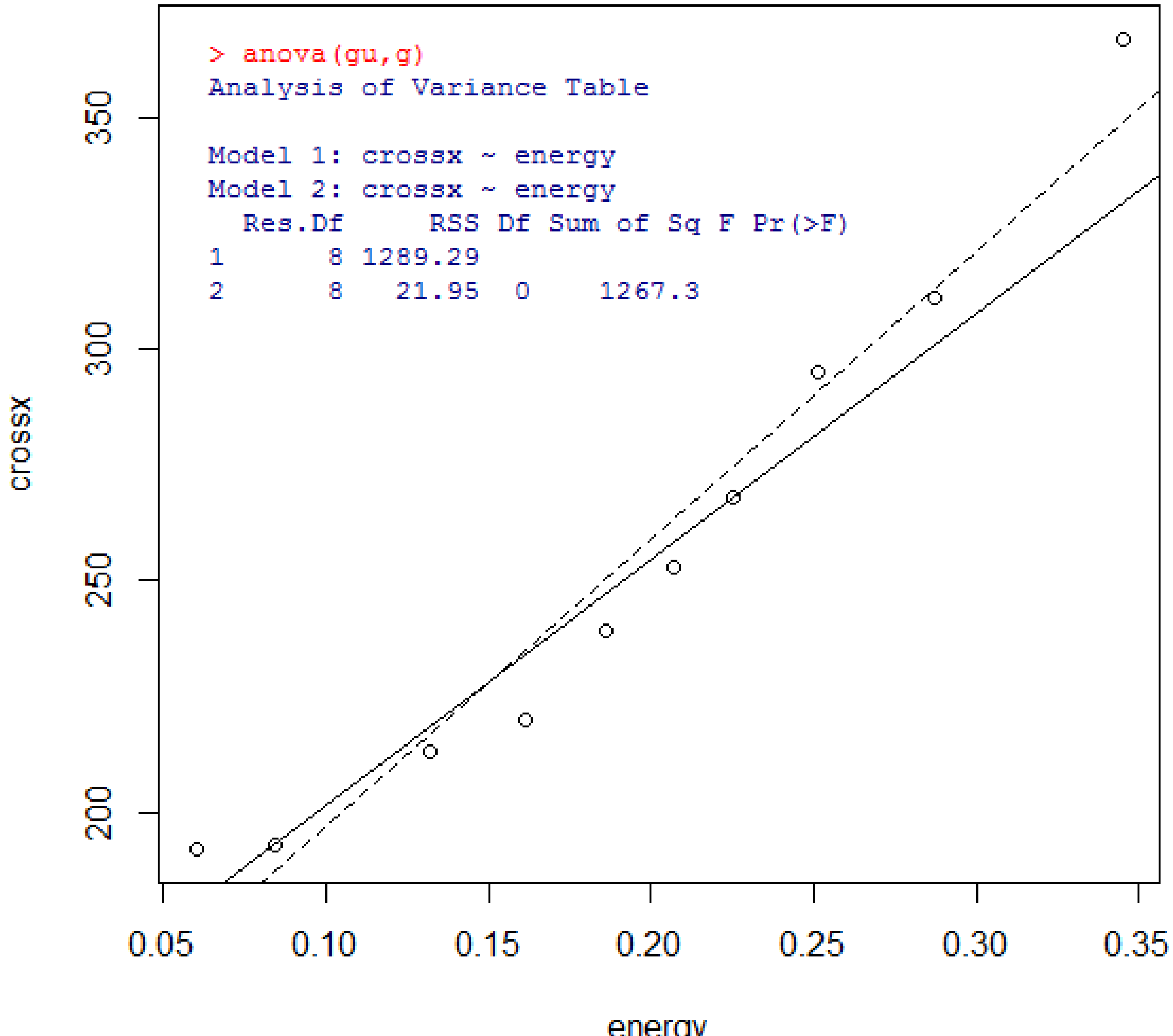
Call:
lm(formula = crossx ~ energy, data = strongx, weights = sd^-2)

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 
```

Graphical comparison



Weighted least square shown in solid. Unweighted is dashed

Iteratively Reweighted Least Squares

- In cases, where the form of the variance of ϵ is not completely known, we may model Σ using a small number of parameters. For example, $\mathrm{var}\ \epsilon_{\mathrm{i}} = \gamma_0 + \gamma_1 x_1$
- The IRWLS fitting Algorithm:
- 1. Start with initial w_{i} , for example $\mathrm{w}_{\mathrm{i}} = 1$
  2. Use least squares to estimate β.
  3. Use the residuals to estimate γ, by regressing x on $\hat{\epsilon}^2$.
  4. Recompute the weights and go to 2

Start with initial w_i to estimate β, $w_i = 1$

```
> a <- rep(1, 10)
> a
 [1] 1 1 1 1 1 1 1 1 1 1
> g<-lm(crossx~energy,strongx,weights=a)
> summary(g)

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

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
```

Use least squares to estimate γ, updates weight and reestimate β.

```
>  for (i in 1:20) {
+ eta <- lmod$fit
+                    lmod$res
+ mod<-lm(lmod$res^2~ crossx,strongx)
+ gamma0<-summary(mod)$coefficients[1]
+ gamma1<-summary(mod)$coefficients[2]
+ w <-abs(1/(gamma0+gamma1*crossx))#change weights
+  lmod <- lm(crossx~energy,strongx,weights=w)
+  cat(i, coef(lmod), "\n")
+ }
1 135.3471 617.7262
2 135.9976 614.0009
3 137.2072 607.0521
4 139.4577 594.049
5 143.8359 568.4552
```

Continue until convergence.

```
16 142.8921 531.344
17 142.8921 531.344
18 142.8921 531.344
19 142.8921 531.344
20 142.8921 531.344
> summary(lmod)

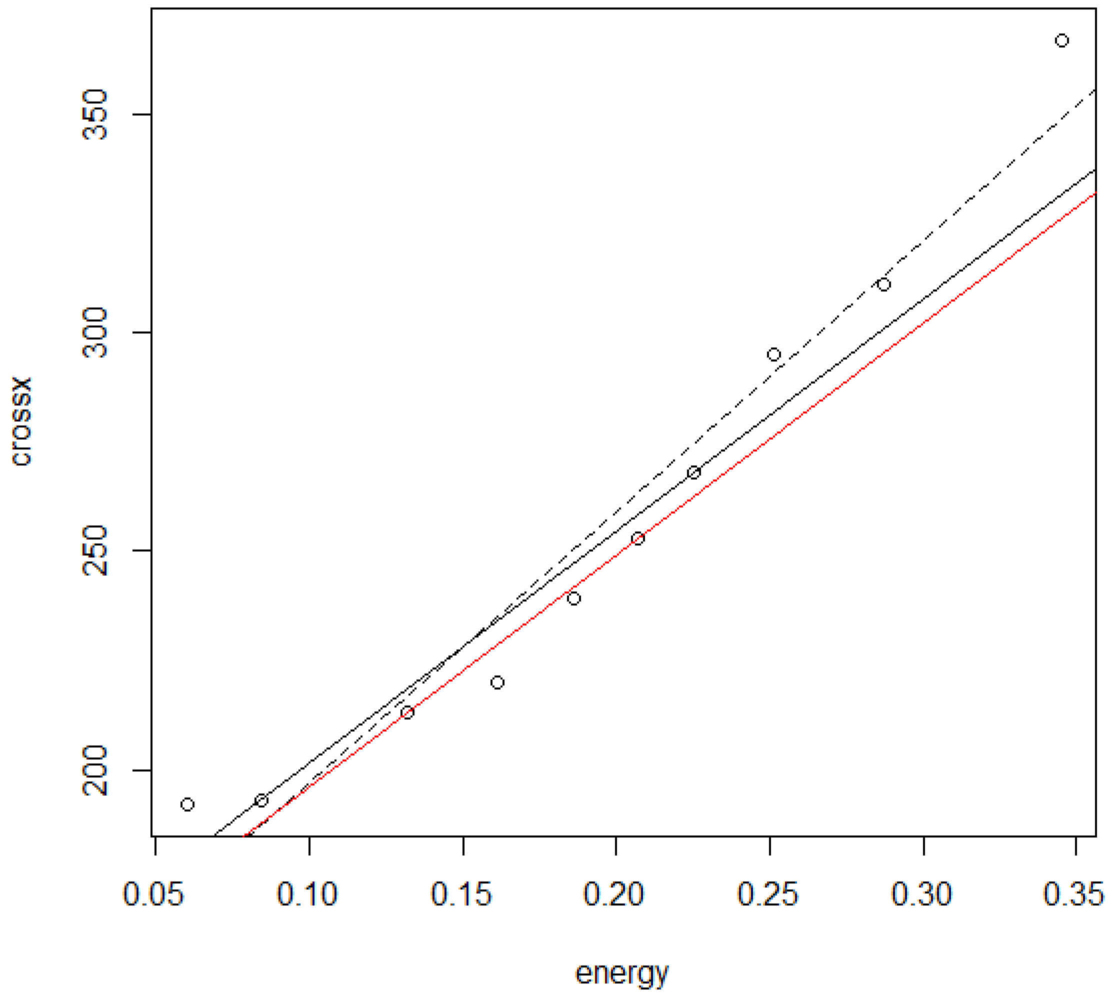
Call:
lm(formula = crossx ~ energy, data = strongx, weights = w)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.44014 -0.00437  0.37332  0.76038  1.39141 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  142.892      9.246   15.45 3.06e-07 ***
energy       531.344     62.002    8.57 2.65e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9362 on 8 degrees of freedom
Multiple R-squared: 0.9018,     Adjusted R-squared: 0.8895 
F-statistic: 73.44 on 1 and 8 DF,  p-value: 2.652e-05
```

Graphical comparison



IRWS shown in red-solid, WLS shown in solid. Unweighted is dashed

Summary

- IRWLS may be useful when the errors are uncorrelated, but have unequal variance where the form of the inequality is **unknown.**
- some concerns about this such as how is subsequent inference about β affected? Also how many degrees of freedom do we have? more complex and time-consuming?
- More details may be found in Carroll and Ruppert (1988).
- An alternative approach is to model the variance and jointly estimate the regression and weighting parameters using likelihood based method. This can be implemented in R using the gls() function in the nlme library.

Thank you

Thank you

Thank you

More on the general case without using GLS

- <u>**Fit a linear model**</u>
  > data(longley)
  > g <- lm(Employed ~GNP + Population, data=longley)
  > summary(g, cor=T)
- 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 *

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
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

The general case

- Successive errors could be correlated.
- Estimate the correlation using a simple autoregressive form:
- $\epsilon_{i+1} = \rho\epsilon_i + \delta_i$ where $\delta_i \sim N(0,\tau^2)$
- > cor(g$res[-1],g$res[-16])
  [1] 0.3104092
- We now construct the Σ matrix and compute the GLS estimate of β along with its standard errors.

The general case

- > x <- model.matrix(g)
  > Sigma <- diag(16)
  > Sigma <- 0.31041^ 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
```

The general case

- Our initial estimate of the correlation is 0.31 but once we fit our GLS model, need to re-estimate

```
> cor(res[-1],res[-16])
[1] 0.3564162
```

  and then recompute the model again with the correlation = 0.3564162. This process would be iterated until convergence.

The general case

- Compare with the model where the errors are assumed to be correlated.
  One way is to regress $S^{-1}y$ on $S^{-1}x$:
- ```
  > sm <- chol(Sigma)
  > smi <- solve(t(sm))
  > sx <- smi %*% x
  > sy <- smi %*% longley$Employed
  > lm(sy~sx -1)$coef
  sx(Intercept)     sxGNP        sxPopulation
    94.8988949    0.0673895  -0.4742741
  ```
  ```