

# Problems with non-IID errors

Analysis of Ecological and Environmental Data

QERM 514

Mark Scheuerell

15 April 2020

# Goals for today

- Understand how generalized least squares can be used when the errors are correlated
- Understand how weighted least squares can be used when the errors have nonconstant variance
- Understand how robust methods can be used when the errors are non-normal or when we have influential observations

# Concerns re: model assumptions

1. Adequacy of the model
2. Independence of errors
3. Non-constant variance
4. Normality of errors

# Possible options

1. Adequacy of the model → possible change in structure
2. Independence of errors → generalized least squares
3. Non-constant variance → weighted least squares
4. Normality of errors → robust methods, transformations

# Generalized least squares

Consider our general model where

$$y_i = \beta_0 + \beta_1 x_{1,i} + \beta_2 x_{2,i} + \cdots + \beta_k x_{k,i} + e_i$$
$$e_i \sim N(0, \sigma^2)$$

which we can write more compactly as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{e}$$
$$\mathbf{e} \sim \text{MVN}(\mathbf{0}, \boldsymbol{\Sigma})$$

What exactly is  $\boldsymbol{\Sigma}$ ?

# An aside on multivariate normals

Consider a vector of random variables  $\mathbf{z}$

The mean of  $\mathbf{z}$  is also a vector, but the variance of  $\mathbf{z}$  is a matrix

$$\mathbf{z} \sim \text{MVN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$
$$\Downarrow$$
$$\mathbf{z} \sim \text{MVN} \left( \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{bmatrix}, \begin{bmatrix} \sigma_1^2 & \gamma_{12} & \cdots & \gamma_{1n} \\ \gamma_{21} & \sigma_2^2 & \cdots & \gamma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{n1} & \gamma_{n2} & \cdots & \sigma_n^2 \end{bmatrix} \right)$$

# An aside on multivariate normals

More specifically for  $\Sigma$

- the diagonal contains the variances  $\sigma_i^2$
- the off-diagonals are the covariances  $\gamma_{ij} = \gamma_{ji}$

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \gamma_{12} & \cdots & \gamma_{1n} \\ \gamma_{21} & \sigma_2^2 & \cdots & \gamma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{n1} & \gamma_{n2} & \cdots & \sigma_n^2 \end{bmatrix}$$

# An aside on multivariate normals

One of our key assumptions in ordinary least squares is that the errors  $\mathbf{e}$  are *independent and identically distributed* (IID)

*Independent* means the covariances are all zero

$$\Sigma = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n^2 \end{bmatrix}$$



# An aside on multivariate normals

*Identically distributed* means the variances are all the same

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma^2 & 0 & \cdots & 0 \\ 0 & \sigma^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma^2 \end{bmatrix} = \sigma^2 \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \sigma^2 \mathbf{I}$$

# Generalized least squares

In cases where the variances are not equal or the covariances are not zero, we can use *generalized least squares* (GLS)

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \gamma_{12} & \cdots & \gamma_{1n} \\ \gamma_{21} & \sigma_2^2 & \cdots & \gamma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{n1} & \gamma_{n2} & \cdots & \sigma_n^2 \end{bmatrix}$$

# Generalized least squares

Let's begin by expressing  $\Sigma$  as a product of  $\sigma^2$  and a matrix  $\mathbf{C}$ , such that

$$\Sigma = \sigma^2 \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{bmatrix}$$

# Generalized least squares

Next we will specify  $\mathbf{C}$  in terms of its [Cholesky decomposition](#)

$$\mathbf{C} = \mathbf{S}\mathbf{S}^\top$$

where  $\mathbf{S}$  is a *lower triangular* matrix

$$\mathbf{S} = \begin{bmatrix} s_{11} & 0 & \cdots & 0 \\ s_{21} & s_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ s_{n1} & s_{n2} & \cdots & s_{nn} \end{bmatrix}$$

# Generalized least squares

You can think of the Cholesky decomposition as a square root transformation for matrices

Consider this example

```
## covariance matrix
CC <- matrix(c(3,4,3,4,8,6,3,6,9), 3, 3)
CC
```

```
##      [,1] [,2] [,3]
## [1,]    3    4    3
## [2,]    4    8    6
## [3,]    3    6    9
```

# Generalized least squares

```
## Cholesky decomposition; `chol()` returns  $t(S)$   
SS <- t(chol(CC))  
round(SS, 2)
```

```
##      [,1] [,2] [,3]  
## [1,] 1.73 0.00 0.00  
## [2,] 2.31 1.63 0.00  
## [3,] 1.73 1.22 2.12
```

```
## reassemble Sigma  
SS %*% t(SS)
```

```
##      [,1] [,2] [,3]  
## [1,]    3    4    3  
## [2,]    4    8    6  
## [3,]    3    6    9
```

# Generalized least squares

We can now use our decomposition matrix  $\mathbf{S}$  to transform our standard regression model, such that

$$\begin{aligned} \mathbf{y} &= \mathbf{X}\boldsymbol{\beta} + \mathbf{e} \\ &\Downarrow \\ \mathbf{S}^{-1}\mathbf{y} &= \mathbf{S}^{-1}\mathbf{X}\boldsymbol{\beta} + \mathbf{S}^{-1}\mathbf{e} \\ \mathbf{y}' &= \mathbf{X}'\boldsymbol{\beta} + \mathbf{e}' \end{aligned}$$

and hence

$$\text{Var}(\mathbf{e}') = \text{Var}(\mathbf{S}^{-1}\mathbf{e})$$

# Generalized least squares

We can now solve for  $\text{Var}(\mathbf{e}')$

$$\begin{aligned}\text{Var}(\mathbf{e}') &= \text{Var}(\mathbf{S}^{-1}\mathbf{e}) \\ &= \mathbf{S}^{-1}\text{Var}(\mathbf{e})(\mathbf{S}^{-1})^\top \\ &= \mathbf{S}^{-1}\boldsymbol{\Sigma}(\mathbf{S}^{-1})^\top \\ &= \mathbf{S}^{-1}[\sigma\mathbf{C}](\mathbf{S}^{-1})^\top \\ &= \mathbf{S}^{-1}[\sigma\mathbf{S}\mathbf{S}^\top](\mathbf{S}^{-1})^\top \\ &= \sigma\mathbf{I}\end{aligned}$$

and the errors  $\mathbf{e}'$  are now IID!



# Generalized least squares

The  $SSE$  is then given by

$$\begin{aligned}\mathbf{e}'^\top \mathbf{e}' &= (\mathbf{y}' - \mathbf{X}'\hat{\boldsymbol{\beta}})^\top (\mathbf{y}' - \mathbf{X}'\hat{\boldsymbol{\beta}}) \\ &= (\mathbf{S}^{-1}\mathbf{y} - \mathbf{S}^{-1}\mathbf{X}\hat{\boldsymbol{\beta}})^\top (\mathbf{S}^{-1}\mathbf{y} - \mathbf{S}^{-1}\mathbf{X}\hat{\boldsymbol{\beta}}) \\ &= (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})^\top \mathbf{S}^{-1\top} \mathbf{S}^{-1} (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}) \\ &= (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})^\top \mathbf{C}^{-1} (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})\end{aligned}$$

# Generalized least squares

We can minimize the SSE to find  $\hat{\beta}$

$$\begin{aligned}\hat{\beta} &= \min (\mathbf{y} - \mathbf{X}\beta)^\top \mathbf{C}^{-1} (\mathbf{y} - \mathbf{X}\beta) \\ &= (\mathbf{X}^\top \mathbf{C}^{-1} \mathbf{X}) \mathbf{X}^\top \mathbf{C}^{-1} \mathbf{y}\end{aligned}$$

and from this find that

$$\text{Var}(\hat{\beta}) = \sigma^2 (\mathbf{X}^\top \mathbf{C}^{-1} \mathbf{X})^{-1}$$

# Generalized least squares

This all looks great, but typically we do not know  $\mathbf{C}$

Let's think about situations where the  $e_i$  are not independent

- time series
- spatial data
- grouped (blocked) data

# Autocorrelated data

When modeling data that are collected over time, it's common that the predictor variable(s) will not account for all of the temporal structure in the data

# Autocorrelated data

One option is to explicitly model the errors as an *autoregressive process* where (replacing  $i$  with  $t$ )

$$\begin{aligned}e_t &= \phi e_{t-1} + \delta_t \\ \delta_t &\sim \text{N}(0, \tau^2) \\ &\Downarrow \\ e_t &\sim \text{N}(\phi e_{t-1}, \tau^2)\end{aligned}$$

To do this in **R** we need additional packages not included with the base installation (eg, **nlme**)

# Weighted least squares

Sometimes the errors are *independent but not identically distributed* and the covariance matrix is

$$\Sigma = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n^2 \end{bmatrix}$$

# Weighted least squares

Sometimes the errors are *independent but not identically distributed* and the covariance matrix is

$$\Sigma = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n^2 \end{bmatrix}$$

In these cases we can use a subset of generalized least squares called *weighted least squares*

# Weighted least squares

Similar to GLS, we can express  $\Sigma$  in terms of  $\sigma$  and a matrix  $\mathbf{C}$  with non-diagonal elements equal to 0

$$\Sigma = \sigma^2 \begin{bmatrix} c_1 & 0 & 0 & 0 \\ 0 & c_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & c_n \end{bmatrix} = \sigma^2 \mathbf{C}$$



# Weighted least squares

We saw earlier that we could fit a GLS model with OLS if we could express the variance in the transformed errors as a function of the Cholesky decomposition of  $\mathbf{C} = \mathbf{S}\mathbf{S}^T$ , where

$$\begin{aligned}\text{Var}(\mathbf{e}') &= \text{Var}(\mathbf{S}^{-1}\mathbf{e}) \\ &= \sigma\mathbf{I}\end{aligned}$$

This suggest a *weighting* of  $\mathbf{e}$  proportional to  $\mathbf{S}^{-1}$

# Weighted least squares

Let's define our variance multiplier  $\mathbf{S}^{-1}$  as

$$\mathbf{S}^{-1} = \begin{bmatrix} \frac{1}{\sqrt{w_1}} & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{w_2}} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \frac{1}{\sqrt{w_n}} \end{bmatrix}$$

# Weighted least squares

From this we can define a weights matrix  $\mathbf{W}$  as

$$\begin{aligned}\mathbf{W} &= \mathbf{S}\mathbf{S}^\top \\ &= \begin{bmatrix} w_1 & 0 & 0 & 0 \\ 0 & w_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & w_3 \end{bmatrix}\end{aligned}$$

How do we choose the weights?

# Weighted least squares

Errors proportional to the predictor

In general, the weights  $w_i$  should reflect differences in the variance of the errors  $\epsilon_i$

In many ecological applications, we find that the variance is proportional to a predictor

$$\text{Var}(\epsilon_i) = x_i \sigma^2$$

This suggests  $w_i = \frac{1}{x_i}$

# Weighted least squares

Observations are averages

It's not uncommon that the observations  $y_i$  are actually averages of several pieces of raw data

In that case

$$\text{Var}(\epsilon_i) = \frac{\sigma^2}{n_i}$$

This suggests  $w_i = n_i$

# Weighted least squares

Observations are sums

Similarly, the observations  $y_i$  might be sums of several pieces of raw data

In that case

$$\text{Var}(\epsilon_i) = n_i \sigma^2$$

This suggests  $w_i = \frac{1}{n_i}$

# Robust regression

We saw in the last lecture that non-normal errors & unusual observations can affect model fits

- heteroscedastic errors where  $\text{Var}(e_i) \propto n_i$
- outliers that do not come from the data generating process

# Robust regression

We saw in the last lecture that non-normal errors & unusual observations can affect model fits

- heteroscedastic errors where  $\text{Var}(e_i) \propto n_i$
- outliers that do not come from the data generating process

In these case we can use so-called *robust regression*



# Robust regression

## M-estimation

Recall that our goal in ordinary least squares is to minimize the error sum-of-squares ( $SSE$ )

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

The objective function is the squared differences between the data and their estimates

# Robust regression

## M-estimation

An alternative is to minimize a different function

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

$\Downarrow$

$$SSE = \sum_{i=1}^n f(z)$$

# Robust regression

## M-estimation

One possibility for  $f(z)$  is the *least absolute deviation* (LAD)

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

# Robust regression

## M-estimation

Another possibility is *Huber's method* where

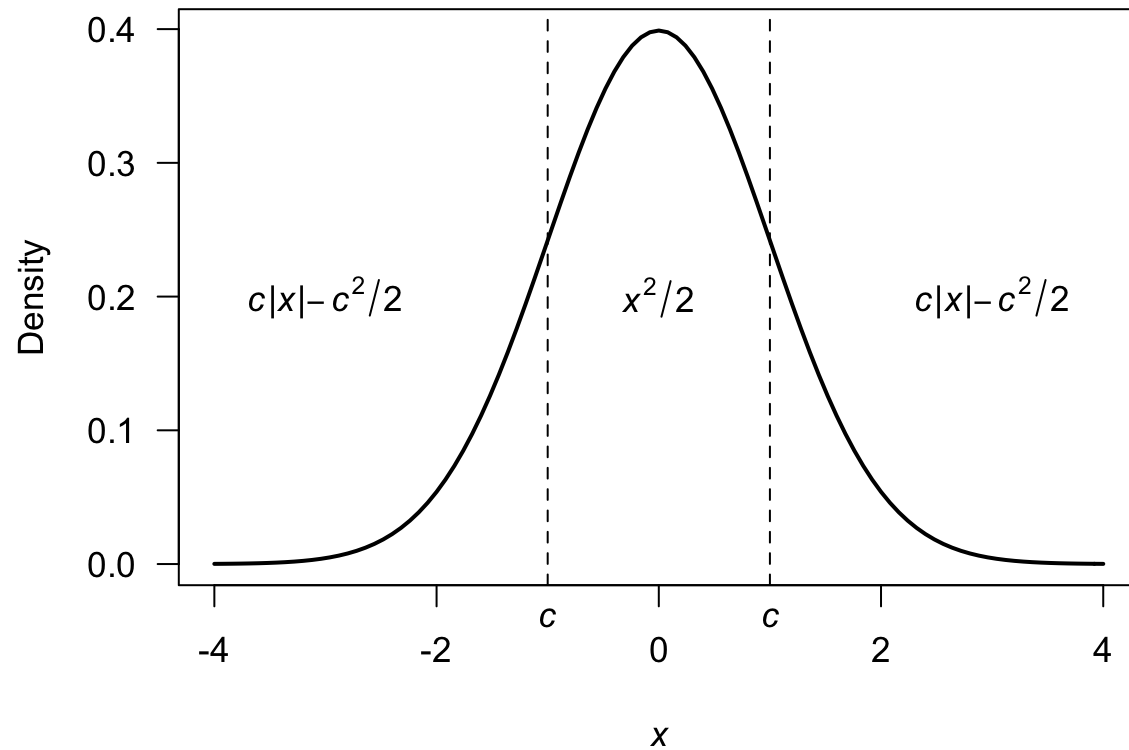
$$SSE = \sum_{i=1}^n f(z)$$

$$f(z) = \begin{cases} \frac{z^2}{2} & \text{if } |z| \leq c \\ c|z| - \frac{c^2}{2} & \text{otherwise} \end{cases}$$

and  $c = \hat{\sigma} \propto \text{Median}(|\hat{\epsilon}|)$

# Robust regression

M-estimation via Huber's method



# Robust regression

M-estimation via Huber's method

Note the following:

- M-estimation does not address points with large leverage
- it says nothing about which predictors to include
- it says nothing about which transformations to make

# Robust regression

## Least trimmed squares

M-estimation will fail if the large errors are numerous and extreme in value

*Least trimmed squares* (LTS) is a resistant regression method that deals well with this situation

# Robust regression

## Least trimmed squares

LTS minimizes the sum of squares of the  $q$  smallest residuals

$$SSE = \sum_{i=1}^n e_i^2 \rightarrow SSE_q = \sum_{i=1}^q e_{(i)}^2$$

and  $(i)$  indicates the residuals are sorted in ascending order

The default is  $q = \lfloor n/2 \rfloor + \lfloor (k + 1)/2 \rfloor$

where  $\lfloor \cdot \rfloor$  is the floor function



# Robust regression

## Least trimmed squares

In practice, we can easily fit LTS models in R with `MASS::ltsreg()` but it does not provide estimates of the parameter uncertainty

We can, however, estimate it via bootstrapping

# Robust regression

## Least trimmed squares bootstrapping procedure

1. Fit your model to the data
2. Calculate  $\mathbf{e} = \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}$
3. Do the following *many* times:
  - Generate  $\mathbf{e}^*$  by sampling *with replacement* from  $\mathbf{e}$
  - Calculate  $\mathbf{y}^* = \mathbf{X}\hat{\boldsymbol{\beta}} + \mathbf{e}^*$
  - Estimate  $\hat{\boldsymbol{\beta}}^*$  from  $\mathbf{X}$  &  $\mathbf{y}^*$ )
4. Select the  $\frac{\alpha}{2}$  and  $(1 - \frac{\alpha}{2})$  percentiles from the saved  $\hat{\boldsymbol{\beta}}^*$

# Summary

1. Robust methods protect against long-tailed errors, but they cannot overcome problems with the choice of model and its variance structure

# Summary

1. Robust methods protect against long-tailed errors, but they cannot overcome problems with the choice of model and its variance structure
2. Robust methods give  $\hat{\beta}$  without the associated inferential methods, but we can use bootstrapping to overcome this

# Summary

1. Robust methods protect against long-tailed errors, but they cannot overcome problems with the choice of model and its variance structure
2. Robust methods give  $\hat{\beta}$  without the associated inferential methods, but we can use bootstrapping to overcome this
3. Robust methods can be used to confirm least squares estimates; it's worth checking if they deviate from one another

# Summary

1. Robust methods protect against long-tailed errors, but they cannot overcome problems with the choice of model and its variance structure
2. Robust methods give  $\hat{\beta}$  without the associated inferential methods, but we can use bootstrapping to overcome this
3. Robust methods can be used to confirm least squares estimates; it's worth checking if they deviate from one another
4. Robust methods are useful when data need to be fit automatically without human intervention, which is rare in ecology