UNIVERSITY OF MANNHEIM
School of Social Sciences

Advanced Quantitative Methods in Political Science: Maximum Likelihood Estimation and Heteroskedastic Regression

Thomas Gschwend | Oliver Rittmann | Viktoriia Semenova

Week 5 - 16 March 2022

Leftovers from last week:
MLE and Statistical Inference

Properties of the Maximum (i.e. of $\hat{\theta}_{ML}$)

Small Sample Properties

- Invariance to reparameterization
    - Rather than estimating a parameter $\hat{\theta}_{ML}$, one can first estimate a function $g(\hat{\theta}_{ML})$, which is also a ML estimator.
    - In a second step, recover $\hat{\theta}_{ML}$ from $g(\hat{\theta}_{ML})$.
    - Very useful because $g(\hat{\theta}_{ML})$ might be easier derived, or has an more intuitive interpretation (see e.g., King & Browning's 1987 *APSR*)
    - Allows for transformation of parameters (logit transformation of probabilities; logarithmic transformation of variances; Fisher z-transformation of correlations)
- Invariance to sampling plans
    - Information about how data is collected (e.g., sample size) that does *not* affect the likelihood is irrelevant.
    - OK to look at results while deciding how much (further) data to collect.
    - Allowed to pool data (if independent, just add LL to the existing one!) to get more precise estimates
- Minimum Variance Unbiased Estimator (MVUE)
    - A single unbiased estimator with smallest variance (not necessarily linear!).

Properties of the Maximum (i.e. of $\hat{\theta}_{ML}$)

Asymptotic Properties (think of *repeated sampling*, i.e., let $\{\hat{\theta}_N\}$ be a sequence of estimators calculated in the same way from larger and larger samples of size N. For each sample size, $\hat{\theta}_N$ has a *sampling distribution*)

- Consistency
  - From the *Law of Large Numbers*, as $N \rightarrow \infty$, the sampling distribution of $\hat{\theta}_{ML}$ collapses to a spike over the (true) parameter value θ.
- Asymptotic normality
  - From the *Central Limit Theorem*, as $N \rightarrow \infty$, the sampling distribution of $\hat{\theta}_{ML}/se(\hat{\theta}_{ML})$ converges to the normal distribution (Mean?, Variance?).
  - No matter what distribution we assumed in the model for θ itself!
  - Allows us to do hypothesis testing and to construct confidence intervals.
- Asymptotic efficiency
  - Among all consistent, asymptotically normal distributed estimators, $\hat{\theta}_{ML}$ has the smallest variance.
  - $\hat{\theta}_{ML}$ contains as much information as can be packed into a point estimator.

Intro

What should you take home from this class today?

- Log-likelihoods can be approximated around the maximum by a matrix of second derivatives (aka the *Hessian*) that measures the curvature in the neighborhood of the MLE.
- We get standard errors as square roots of diagonal terms of the VarCov matrix.
- We will implement our first MLE estimator in R and also estimate a (heteroskedastic) regression model.

Three Steps to come up with a suitable ML Estimator for your Research Question

1. Formulate a suitable probability model of the data-generating process including assumptions of how Y is distributed (i.e., stochastic component) and a parametrization of stuff that gets estimated (i.e., systematic component).
2. Write down the (log-)likelihood function based on your parametrization and assumptions.
3. Maximize the log-likelihood, analytically (often hard, even impossible) or numerically (use functions in R).

MLE and Standard Errors

Justifying Standard Errors

- The degree of *curvature* of the LL of the normal depends on the second derivative, because (remember from last week?) the LL of the normal is quadratic polynomial around the MLE.
- This is generally not the case, but every (i.e. non-normal) LL can be approximated by a quadratic polynomial around the maximum.
- We take the *second order Taylor series expansion* of the log-likelihood with respect to θ around the maximum $\hat{\theta}$:

$$f(\theta) = lnL(\theta|y) \approx lnL(\hat{\theta}|y) + \left(\frac{\partial lnL(\hat{\theta}|y)}{\partial\theta}\right)'(\theta - \hat{\theta}) + \frac{1}{2}(\theta - \hat{\theta})'\frac{\partial^2 lnL(\hat{\theta}|y)}{\partial\theta\partial\theta'}(\theta - \hat{\theta})$$

- This is fairly general. In fact, any function can be approximated by a quadratic function (see Taylor series demonstration!)

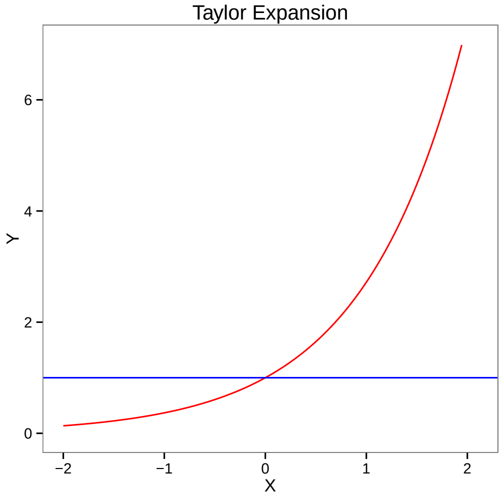

Figure 1: The exponential function, $f(x) = e^x$, and the Taylor series approximation: $x_0 = 0, f_0(x_1) = 1$ (from Wikipedia)

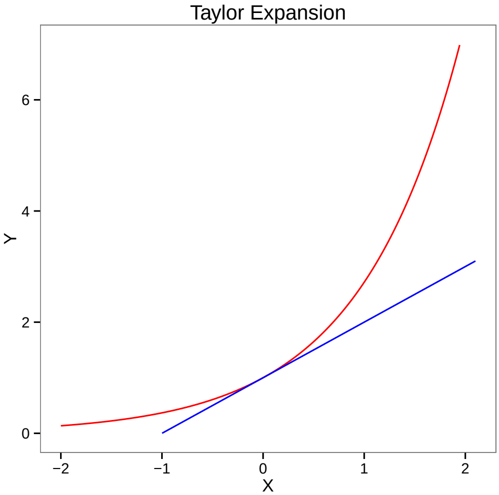

Figure 2: The exponential function, $f(x) = e^x$, and the Taylor series approximation: $x_0 = 0$, $f_1(x_1) = 1 + x_1$ (from Wikipedia)

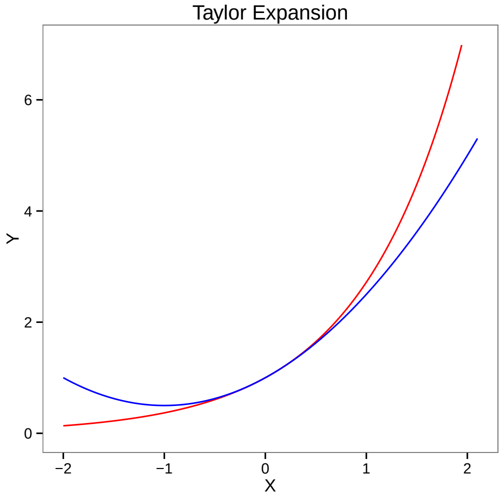

Figure 3: The exponential function, $f(x) = e^x$, and the Taylor series approximation: $x_0 = 0$, $f_2(x_1) = 1 + x_1 + \frac{x_1^2}{2}$ (from Wikipedia)

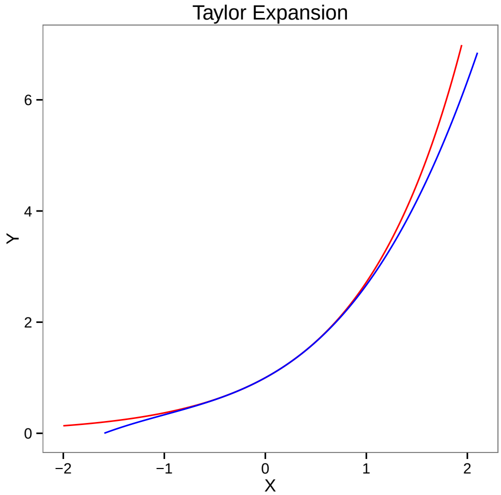

Figure 4: The exponential function, $f(x) = e^x$, and the Taylor series approximation: $x_0 = 0$, $f_3(x_1) = 1 + x_1 + \frac{x_1^2}{2} + \frac{x_1^3}{6}$ (from Wikipedia)

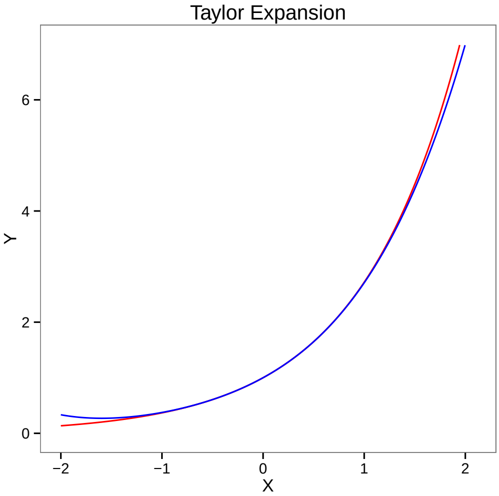

Figure 5: The exponential function, $f(x) = e^x$, and the Taylor series approximation: $x_0 = 0$, $f_4(x_1) = 1 + x_1 + \frac{x_1^2}{2} + \frac{x_1^3}{6} + \frac{x_1^4}{24}$ (from Wikipedia)

Standard Errors

- Instead of plotting the entire likelihood function, we can summarize the curvature in the neighborhood of the maximum with the *Fisher Information Matrix* denoted by $\mathcal{I}(\hat{\theta}|y)$.
- The *information* in the data (i.e., degree of curvature) can be estimated as negative expectation in terms of the second derivative (the so-called *Hessian Matrix*) of the log-likelihood with respect to θ evaluated at $\hat{\theta}$

$$\mathcal{I}(\hat{\theta}|y) = -E\left(\frac{\partial^2 lnL(\theta|y)}{\partial\theta\partial\theta'}\right) = -H(\hat{\theta})$$

- If θ is a single parameter than the larger $\mathcal{I}(\hat{\theta}|y)$, the more curved the log-likelihood and, thus, the more information in the data to estimate $\hat{\theta}$. Hence, we expect to get more precise estimates (i.e., smaller standard errors).

Standard Errors

The *Hessian Matrix* $H(\theta)$ reflects the degree of curvature of the second-order approximation of the log-likelihood, i.e.,

$$H(\theta) = \left(\frac{\partial^2 lnL(\theta|y)}{\partial\theta\partial\theta'}\right) \quad = \quad \begin{pmatrix} \frac{\partial^2 lnL(\theta)}{\partial\theta_0\partial\theta_0'} & \frac{\partial^2 lnL(\theta)}{\partial\theta_0\partial\theta_1'} & \cdots \\ \frac{\partial^2 lnL(\theta)}{\partial\theta_1\partial\theta_0'} & \frac{\partial^2 lnL(\theta)}{\partial\theta_1\partial\theta_1'} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

This is a square, symmetric matrix.

Given that we think about having more precision when having more information (about the curvature around the maximum), the variance-covariance matrix $Var(\hat{\theta})$ is inversely related to the Information Matrix $\mathcal{I}(\theta|y)$, which is the negative of the expected value of the Hessian.

Standard Errors

Thus,

$$Var(\hat{\theta}) = [\mathcal{I}(\theta|y)]^{-1} = [-E\left(H(\theta)\right)]^{-1} = \left[-E\left(\frac{\partial^2 lnL(\theta|y)}{\partial\theta\partial\theta'}\right)\right]^{-1}$$

Because of the expected value operator we have to estimate this matrix. This can be done, for instance, through

$$\widehat{Var}(\hat{\theta}) = \begin{pmatrix} -\frac{\partial^2 lnL(\hat{\theta})}{\partial\theta_0\partial\theta_0'} & -\frac{\partial^2 lnL(\hat{\theta})}{\partial\theta_0\partial\theta_1'} & \cdots \\ -\frac{\partial^2 lnL(\hat{\theta})}{\partial\theta_1\partial\theta_0'} & -\frac{\partial^2 lnL(\hat{\theta})}{\partial\theta_1\partial\theta_1'} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}^{-1} = \left(-H(\hat{\theta})\right)^{-1}$$

As you can see, we can read off the standard errors of $\hat{\theta}_{ML}$ from the square roots of the diagonal elements of this matrix. Thus, $\widehat{Var}(\hat{\theta})$ can be estimated as a function of the matrix of second derivatives. Remember (last week), these are only correct asymptotically!

Variance-Covariance Matrix of a Linear Regression Model

We start with the normal equations and look at the a *gradient* vector

$$\frac{\partial lnL}{\partial\theta} = \begin{pmatrix} \frac{\partial lnL}{\partial\beta} \\ \frac{\partial lnL}{\partial\sigma^2} \end{pmatrix} = \begin{pmatrix} \frac{X'(y-X\beta)}{\sigma^2} \\ -\frac{N}{2\sigma^2} + \frac{(y-X\beta)'(y-X\beta)}{2\sigma^4} \end{pmatrix}$$

Next we take the derivative of each element of the gradient vector wrt β and σ^2.

$$\frac{\partial^2 lnL}{\partial\beta\partial\beta'} = \frac{\partial(\frac{X'(y-X\beta)}{\sigma^2})}{\partial\beta} = -\frac{X'X}{\sigma^2}$$

$$\frac{\partial^2 lnL}{\partial\beta\partial\sigma^2} = \frac{\partial(\frac{X'(y-X\beta)}{\sigma^2})}{\partial\sigma^2} = -\frac{X'\epsilon}{\sigma^4}$$

$$\frac{\partial^2 lnL}{\partial\sigma^2\partial\sigma^2} = \frac{\partial(-\frac{N}{2\sigma^2} + \frac{(y-X\beta)'(y-X\beta)}{2\sigma^4})}{\partial\sigma^2} = \frac{N}{2\sigma^4} - \frac{\epsilon'\epsilon}{\sigma^6}$$

Variance-Covariance Matrix of a Linear Regression Model

Subsequently, we can calculate the remaining entries of the *Hessian Matrix* as

$$H(\theta) = \left(\frac{\partial^2 lnL}{\partial\theta\partial\theta'}\right) = \begin{pmatrix} -\frac{X'X}{\sigma^2} & -\frac{X'\epsilon}{\sigma^4} \\ -\frac{X'\epsilon}{\sigma^4} & \frac{N}{2\sigma^4} - \frac{\epsilon'\epsilon}{\sigma^6} \end{pmatrix}$$

Taking the expectation yields ...

$$E(H(\theta)) = E(\left(\frac{\partial^2 lnL}{\partial\theta\partial\theta'}\right)) = \begin{pmatrix} -E(\frac{X'X}{\sigma^2}) & -E(\frac{X'\epsilon}{\sigma^4}) \\ -E(\frac{X'\epsilon}{\sigma^4}) & E(\frac{N}{2\sigma^4} - \frac{\epsilon'\epsilon}{\sigma^6}) \end{pmatrix} = \begin{pmatrix} -\frac{X'X}{\sigma^2} & 0 \\ 0 & -\frac{N}{2\sigma^4} \end{pmatrix}$$

Thus, the variance-covariance matrix is

$$Var(\hat{\theta}) = [-E\left(H(\theta)\right)]^{-1} = \begin{pmatrix} \sigma^2(X'X)^{-1} & 0 \\ 0 & \frac{2\sigma^4}{N} \end{pmatrix}$$

and can be estimated through ...

$$\widehat{Var}(\hat{\theta}) = \left[-E\left(H(\hat{\theta})\right)\right]^{-1} = \begin{pmatrix} \hat{\sigma}^2(X'X)^{-1} & 0 \\ 0 & \frac{2\hat{\sigma}^4}{N} \end{pmatrix} = \begin{pmatrix} Var(\hat{\beta}_{OLS}) & 0 \\ 0 & Var(\hat{\sigma}^2_{MLE}) \end{pmatrix}$$

Bootstrapping

- OK, I understand that we can use the LL to get standard errors.
- But what should I do in small samples, all standard errors in MLE are only correct asymptotically?
- Use bootstrapping!
  - A bootstrap provides a way to perform a statistical inference by re-sampling (i.e. drawing potentially infinite - or at least a really large number of samples) from the data you have.
  - Thus, assuming the data you have is equivalent to the population you wanna draw inferences to, bootstrap produces multiple samples (obviously by replacement) from the current population.
  - Based on every drawn sample calculate an estimate. Thus, you get a *bootstrap sampling distribution* of θ.
  - The se($\hat{\theta}_{ML}$) would be the standard deviation of this distribution.
  - Use the relevant percentiles to construct confidence intervals.

Likelihood ratio for nested models

- L^* is the likelihood value of the *unrestricted* model.
- L_R^* is the likelihood value of the (nested) *restricted* model.
- Thus, $L^* \geq L_R^*$, i.e. $\frac{L_R^*}{L^*} \leq 1$.
- Substantively, the likelihood ratio is a ratio of two probabilities (aka *risk ratio*):

$$\begin{aligned} \frac{L(\theta_1|y)}{L(\theta_2|y)} &= \frac{k(y)}{k(y)} \frac{P(y|\theta_1)}{P(y|\theta_2)} \\ &= \frac{P(y|\theta_1)}{P(y|\theta_2)} \end{aligned}$$

- Statistically, let $R = -2ln(\frac{L_R^*}{L^*}) = 2(lnL^* - lnL_R^*)$, then – under H_0 of no difference between the two models – R is asymptotically χ^2 distributed, with the degree of freedom equal to the number of restrictions.

Wald Test

- Is $\hat{\theta}_j$ systematically different from a theoretical θ^*?
- Generalized version of a t-test.
- Let $\hat{\theta}_j$ the jth element of $\hat{\theta}$, $\hat{\sigma}_j$ its standard error, the square root of the jth diagonal element of the variance-covariance matrix. Then,

  $$\mathcal{W} = \frac{\hat{\theta}_j - \theta^*}{\hat{\sigma}_j}$$

  is asymptotically standard normal distributed, assuming $H_0 : \theta_j = \theta^*$.
- For the formal Wald test, we can instead also use that $\mathcal{W}^2 \sim \chi^2(1)$.

Score Test

- The *score test* is aka *Lagrange multiplier test*.
- Again, if the null is valid, i.e. $H_0 : \theta_j = \theta^*$, then the restricted estimator should be near the point that maximizes the log-likelihood.
- Therefore, the respective slope should be near zero.
- Given that the *score* $S(\theta_j)$ is the slope of the log-likelihood at θ_j, it can be shown that the score statistic S with

$$\mathcal{S} = \frac{S(\theta_j)}{\sqrt{\mathcal{I}(\theta_j)}}$$

  is asymptotically standard normal distributed, assuming $H_0 : \theta_j = \theta^*$.

Overview: Statistical Inference using ML

(Figure is taken from Fox, Appendix D)

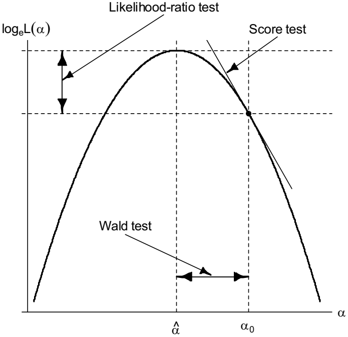

Figure D.19: Tests of the hypothesis H_0: $\alpha = \alpha_0$: The likelihood-ratio test compares $\log_e L(\widehat{\alpha})$ to $\log_e L(\alpha_0)$; the Wald test compares $\widehat{\alpha}$ to α_0; and the score test examines the slope of $\log_e L(\alpha)$ at $\alpha = \alpha_0$.

To sum-up

- If the ML model is correct, then $\hat{\theta}_{ML}$ is a consistent point estimate of θ.
- As the number of observations become large, ...
  - ...the sampling distribution of $\hat{\theta}_{ML}$ becomes normal.
  - ...the log-likelihood becomes quadratic.
  - ...the assumed second-order approximation of the log-likelihood improves.
- There are also several numerical algorithms (e.g. *Newton-Raphson, BHHH, Method of Scoring, L-BFGS-B, BFGS, simulated-annealing*) to find a maximum and estimate the variance-covariance matrix.

Numerical Optimization

- Newton-Raphson works well and quickly for simple functions with global maxima
- Method of Scoring, BHHH and simulated-annealing can be better alternatives when likelihood is complex
- Some practical tips
    - The likelihood can have local maxima or saddle points with which numerical algorithms have a hard time (because they “think” its a global maximum).
    - Use different starting values. They should not matter if a global maximum is detected.
    - Use OLS instead to find first reasonable parameter values.
    - Graph LL by fixing all parameters (but 1 or 2) at reasonable values and graph the rest to eyeball maximum in order to find good starting values.
- When encountering convergence problems, …
    - …you may delete missing values explicitly and try again.
    - …rescale the variables so that they are measured (ideally) on the same scale
    - …try another numerical algorithm

Implementation in R

Implementation in R

- Let's estimate a linear regression model via maximum likelihood instead of using ordinary least squares
- *Step 1:* Assume the following model:

$$\begin{array}{lcll} Y_i & \sim & f_N(y_i|\mu_i, \sigma^2) & \text{stochastic} \\ \mu_i & = & X\beta \; (= \beta_0 + \beta_1 x_i) & \text{systematic} \end{array}$$

- The parameters we are going to estimate using the above parameterization are $\theta = (\mu_i, \sigma^2) = (\beta_0, \beta_1, \sigma^2)$
- We further assume that y_i is iid.

Implementation in R

- *Step 2:* Using our assumptions about the model and the chosen parameterization of the systematic component, we can set up the likelihood function as follows:

$$L(\beta, \sigma^2|y) = (2\pi\sigma^2)^{-N/2}\ exp\left[-\frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i - \beta_0 - \beta_1 x_i)^2\right]$$

- Then (although this is optional) we can take the log of the likelihood function, because it simplifies the next step (i.e. maximization):

$$\begin{aligned} logL(\beta, \sigma^2|y) &= -\frac{N}{2}log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i - \beta_0 - \beta_1 x_i)^2 \\ &= -\frac{N}{2}log(2\pi) - \frac{N}{2}log(\sigma^2) - \frac{1}{2}\sum_{i=1}^{N}\frac{(y_i - \beta_0 - \beta_1 x_i)^2}{\sigma^2} \\ &= -\frac{N}{2}log(\sigma^2) - \frac{1}{2}\sum_{i=1}^{N}\frac{(y_i - \beta_0 - \beta_1 x_i)^2}{\sigma^2} \end{aligned}$$

Implementation in R

- Now, let’s write the log-likelihood as a R-function `lm.lik`:

```
 lm.lik <- function(theta, y, x) {
  beta0 <-  theta[1]
  beta1 <-  theta[2]
  gamma <-  theta[3]
# Parametrize sigma2 to be non-negative
  sigma2 <- exp(gamma)
# Residual
  e <- y - beta0 - beta1*x
# Log lik function for one observation
  logl <- -1/2*log(sigma2) - 1/2*(e^2/(sigma2))
# Log lik function is sum over N observations
  logl <- sum(logl)
  return(logl)
 }
```

Implementation in R

- Here is a slightly more general code of the same likelihood:

```
lm.lik1 <-function(theta,y,X){
     N<-nrow(X) # number of observations
     k<-ncol(X) # number of parameters
 # Supstring paramters theta
     beta<-theta[1:k]
     gamma<-theta[k+1]
 # Parametrize sigma2 to be non-negative
     sigma2 <- exp(gamma)
 # Residual
     e<- y-(X%*%beta)
 # Log lik function fover N observations
     logl <- - 1/2*N*log(sigma2)-1/2*((t(e)%*%e)/(sigma2))
 return(logl)
   }
```

Implementation in R

- *Step 3*: Maximize the log-likelihood numerically. Of course, we could do it analytically (see last week). Now we let the computer do all the work for us.
- R provides a tool named `optim()` which maximizes arbitrary functions numerically if we specify `control=list(fnscale=-1)` (`optim()` tries to minimize by default).
- To maximize our likelihood function, we need to feed `optim()` with a set of starting values (the `optim(stval, ...)`'s first guesses for the parameters).

```
      stval <- c(1,1,1)
```

- Then we simply call `optim()` to maximize a likelihood function (`fn=lm.lik`), with particular starting values (`stval`) and data (`y=y, x=x`)

```
res <- optim(stval, fn=lm.lik, control=list(fnscale=-1),
             y=y, x=x, hessian=TRUE)
> res$par
[1] 49.708304  1.125821 10.378797
> sqrt(diag(solve(-1 * res$hessian)))
[1] 1.6249732 0.4578586 3.7924240
```

- Take some data and see how our $\hat{\theta}_{ML}$ compares to $\hat{\theta}_{OLS}$!

Heteroskedastic Regression

Heteroskedastic Regression

- Now, what if we instead relax the homoskedasticity assumption?
- *Step 1:* Assume the following model:

$$
\begin{array}{lcll}
Y_i & \sim & f_N(y_i|\mu_i, \sigma_i^2) & \text{stochastic} \\
\mu_i & = & X\beta \ (= \beta_0 + \beta_1 x_i) & \text{systematic} \\
\sigma_i^2 & = & exp(\gamma Z) \ \ (= exp(\gamma_0 + \gamma_1 z_i)) & \text{systematic}
\end{array}
$$

- The parameters we are going to estimate using the above parametrization of the model's systematic component are $\theta = (\beta_0, \beta_1, \gamma_0, \gamma_1)$
- We further assume that the y_i are independently distributed.
- Thus, we get the following log-likelihood function:

$$
\begin{aligned}
logL(\theta|y) &= -\frac{N}{2} log(2\pi) - \frac{1}{2}\sum_{i=1}^{N} log(\sigma_i^2) \ - \frac{1}{2}\sum_{i=1}^{N} \frac{(y_i - \beta_0 - \beta_1 x_i)^2}{\sigma_i^2} \\
&= - \frac{1}{2}\sum_{i=1}^{N} (\gamma_0 + \gamma_1 z_i) \ - \frac{1}{2}\sum_{i=1}^{N} \frac{(y_i - \beta_0 - \beta_1 x_i)^2}{exp(\gamma_0 + \gamma_1 z_i)}
\end{aligned}
$$

Heteroskedastic Regression - Implementation in R

- Lets write the LL as a R-function `hetero.lik`, but this time with four arguments (θ, y, x, z):

```
hetero.lik <- function(theta, y, x, z) {
    beta0 <-  theta[1]
    beta1 <-  theta[2]
    gamma0 <- theta[3]               # This line is new
    gamma1 <- theta[4]               # This line is new
# Residual
    e <- y - beta0 - beta1*x
# Variance parameterization
    sigma2 = exp(gamma0 + gamma1*z)  # This line is new
# Log lik function for one observation
    logl <- -1/2*log(sigma2) - 1/2*(e^2/(sigma2))
# Log lik function is sum over N observations
    logl <- sum(logl)
    return(logl)
  }
```

- Note, we need to feed `optim()` with four starting values!

Heteroskedastic Regression - Implementation in R

```
  # start values for maximization algorithm  - now we need 4 values
  stval <- c(0,0,0,0)

  # maximize the likelihood function numerically using optim()

res2 <- optim(stval,            # starting values
     fn=hetero.lik,             # the likelihood function
     control=list(fnscale=-1),  # maximize rather than minimize funct
     y=y, x=x, z=z,             # the data
     hessian=TRUE)              # return numerical Hessian

cat("MLE Betas\n", res2$par[1:2], "\n\n")
cat("MLE Gammas\n", (res2$par[3:4]), "\n\n")
cat("Hessian\n")
print(res2$hessian)
cat("\n MLE St. Errors\n", sqrt(diag(solve(-1*res2$hessian))), "\n\n")
```

Quo vadis AQM

Infrastructure of "Advanced Quantitative Methods" Course

Three steps to come up with a suitable ML Estimator for your research question

1. Formulate a suitable probability model of the data-generating process including assumptions of how Y is distributed (i.e., stochastic component) and a parametrization of stuff that gets estimated (i.e., systematic component).
2. Write down the (log-)likelihood function based on your parametrization and assumptions.
3. Maximize the Log-Likelihood, analytically (often hard, even impossible) or numerically (use functions in R).

There are two more things we need to talk about this semester:

- Interpretation of estimation results through simulating quantities of interest (*you have seen this last semester as well as in the lab*)
- How to check whether the assumed model does fit the data? (*Coming soon!*)

Then, we can apply this infrastructure to any existing model or come-up with our own model.