Lab 9

Bob Douma

19 October 2018

Learning goals

You will learn how to:

1. Estimate the parameters of a model that contains both a continuous and a categorical predictor through maximum likelihood
2. Estimate the parameters of a model with the variance varying as a function of a covariate (`gls`)
3. Apply mixed effect models to data with nested structures

Fitting models containing categorical predictors

1. Take the fifth dataset of the six datasets you have worked with earlier on. Assume that the function was generated by a decreasing exponential function $ae^{(-bx)}$ and estimate the parameters a and b.

```
read.csv("shapes.csv") # and select fifth dataset
# test dataset five for differences between groups
nll0 = function(par,dat){
  a = par[1]
  b = par[2]
  ymean = a*exp(-b*dat$x)
  nll = -sum(dpois(dat$y,lambda=ymean),log=T)
  return(nll)
}

par=c(4,0.2)
opt1 = optim(par=par,fn=nll,dat=dat)
```

2. Adjust the likelihood function such that it can accomodate for different values of b depending on the group an observation belong to. The dataset consist of a column `group` with two levels that indicate to which group an observation belongs. Use the following pseudocode to achieve this and/or check page 305 for in inspiration:

   a. Adapt the likelihood function such that the parameter `b` depends on the group.
   b. Adjust the starting values so it contains multiple starting values for `b`

3. Estimate the parameters a and b when letting b depend on the group. Compare the negative loglikelihood of this model with the model fitted in question 1. Which has a better fit?

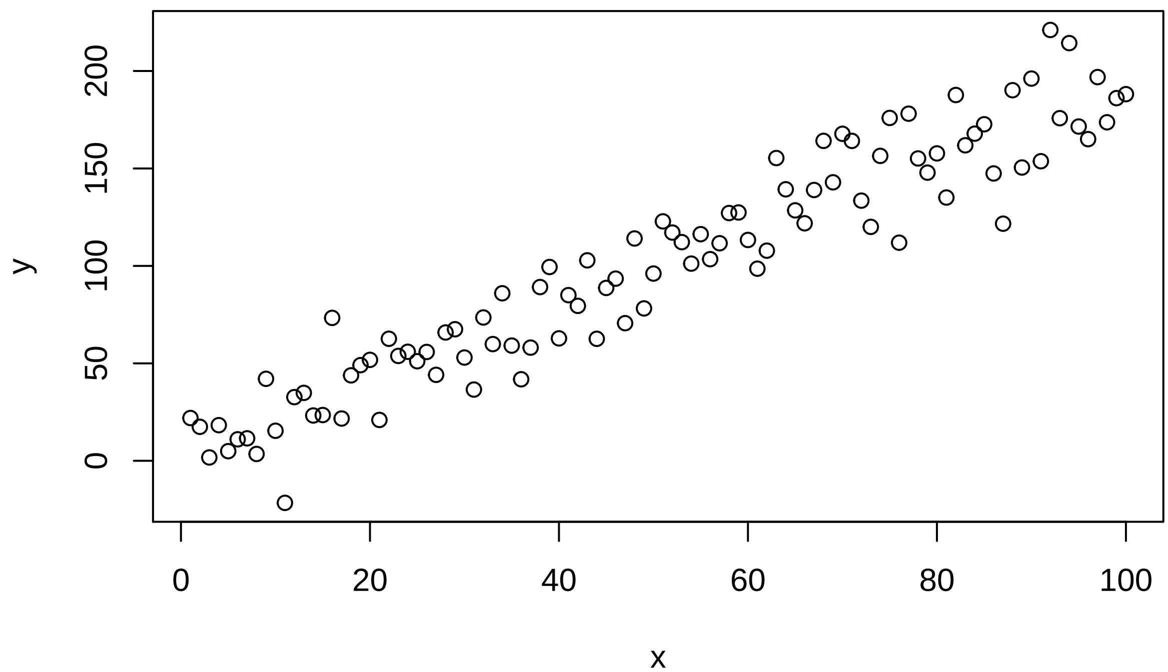
4. Apply model selection techniques (Likelihood ratio test, AIC or BIC) to select the most parsimonious model. Are the models nested? Which model is preferred?

Fitting models with the heterogenous variance

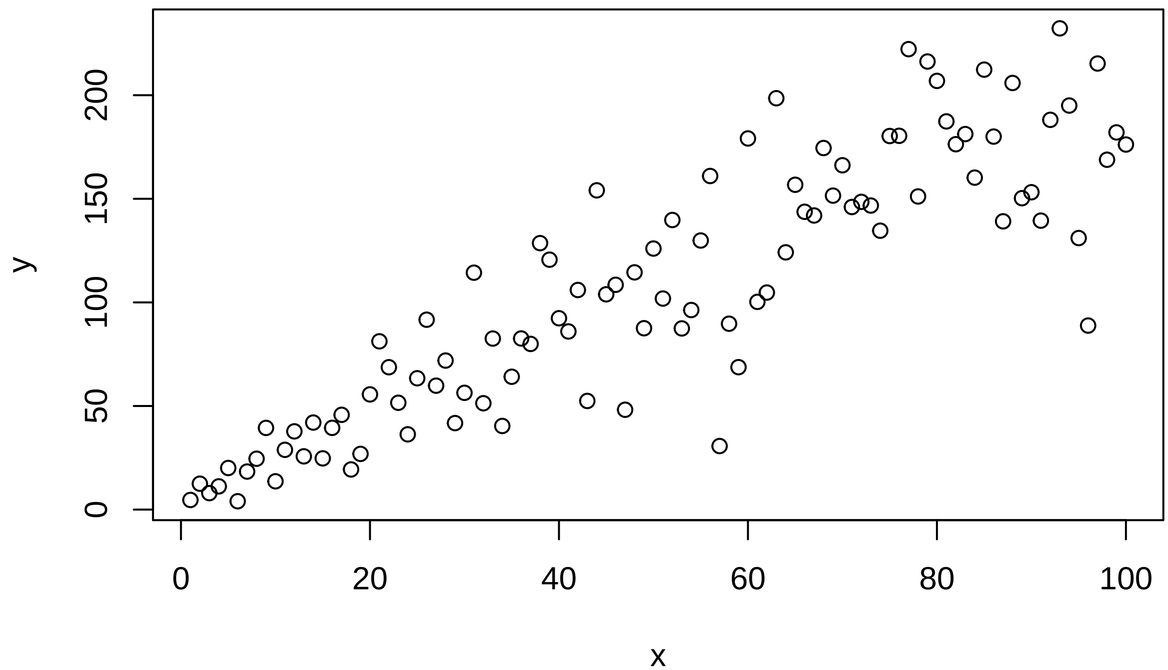
It is common to observe in your data that the variance of your data is not constant along x. In the above example we observed a decreasing variance with increasing x. In case of the Poisson distribution, the relationship between the mean and the variance is fixed through λ. For other probability distributios with ≥ 2 parameters the variance can be modelled (somewhat) independent of each other. An classic example is the normal distribution with mu equal to the mean of the distribution and σ to the square root of the variance. Traditionally, when the variance increases with x log transformations are applied. However, the variance can also be modelled explictly. The purpose of this excercise is to show you give you insight how this can done (by showing a simple but wrong approach), followed by the correct, but canned approach `gls` in `R`.

To illustrate the principle, we generate data from scratch first with constant variance, then we heterogenous variance:

```
x = seq(1,100,length.out = 100)
y = 2 + 2 * x + rnorm(100,0,20)
plot(y~x)
```




```
x = seq(1,100,length.out = 100)
y = 2 + 2 * x + rnorm(100,0,sqrt(20*x^1))
plot(y~x)
```




```
dat = data.frame(x,y)
```

1. Estimate the paramaters of the model through applying the Bolker approach. Make two models, one with constant variance and another one with the variance as a function of x
2. Compare the AIC of both models. Which model is preferred?
3. Now we will apply the function `gls` (from the package `MASS`). Do the values of the `gls.2` correspond to the values that were obtained through the Bolker approach? Look careful at the residual standard error, the equation how `varPower` is fitted and how the data was generated. To fit the models you can type:

```
library(nlme)
gls.1 = gls(y~x, data=dat,method="ML")
summary(gls.1)
```

```
## Generalized least squares fit by maximum likelihood
##   Model: y ~ x
##   Data: dat
##        AIC      BIC    logLik
##   966.7254 974.5409 -480.3627
##
## Coefficients:
##                 Value Std.Error   t-value p-value
## (Intercept) 11.353011  6.006673  1.890067  0.0617
## x            1.896334  0.103264 18.363860  0.0000
##
##  Correlation:
##   (Intr)
```

```
## x -0.868
##
## Standardized residuals:
##         Min          Q1         Med          Q3         Max
## -3.54406952 -0.57811572 -0.01205459  0.51316824  2.29435089
##
## Residual standard error: 29.5088
## Degrees of freedom: 100 total; 98 residual
```

```r
# to specify the variance as a function of x we can use different functions
# (see chapter 4 of Zuur for details) or see ?varClasses
gls.2 = gls(y~x, weights=varPower(form=~x), data=dat,method="ML")
summary(gls.2)
```

```
## Generalized least squares fit by maximum likelihood
##   Model: y ~ x
##   Data: dat
##        AIC      BIC    logLik
##   931.8929 942.3136 -461.9465
##
## Variance function:
##  Structure: Power of variance covariate
##  Formula: ~x
##  Parameter estimates:
##     power
## 0.6538365
##
## Coefficients:
##                Value Std.Error   t-value p-value
## (Intercept) 4.507032 1.5318911  2.942136  0.0041
## x           2.042948 0.0705296 28.965842  0.0000
##
##  Correlation:
##   (Intr)
## x -0.515
##
## Standardized residuals:
##         Min          Q1         Med          Q3         Max
## -2.82154601 -0.72424140  0.01145194  0.59130639  2.20912386
##
## Residual standard error: 2.275649
## Degrees of freedom: 100 total; 98 residual
```

```r
AIC(gls.1,gls.2)
```

```
##       df      AIC
## gls.1  3 966.7254
## gls.2  4 931.8929
```

4. The variance estimates are biased because they are estimated with *maximum likelihood* and not *restricted maximum likelihood* (choose `method=REML` instead). With restricted maximumlikelihood you account for the fact that you need to the mean of the data to estimate its variance (i.e. $\sigma^2 = \sum_{i=1}^{N}(y_i - \mu)^2$), but that the mean itself is an estimate from the data (i.e. $\bar{y}$) and not the population parameter itself (i.e. μ). Compare the AIC of both models when using `REML`. Which model fits the data better?

```r
x = rnorm(50000,2,4)

optim.norm = function(pars,x){
  mean = pars[1]
  sd = pars[2]
  nll = -sum(dnorm(x,mean,sd,log=T))
}

par=c(mean=1,sd=2)
opt1 = optim(par,optim.norm,x=x)


var.ml = (opt1$par[2])^2
sum((x-mean(x))^2)/length(x)
```

```
## [1] 15.82133
```

```r
sum((x-mean(x))^2)/(length(x)-1)
```

```
## [1] 15.82164
```

```r
var(x)
```

```
## [1] 15.82164
```

```r
(var.ml-var(x))/var(x)*100
```

```
##           sd
## 0.0007237689
```

Fitting models to nested data.