Universiteit Utrecht

Erik-Jan van Kesteren
Methodology & Statistics

Bayesian Linear Models
The **blim** Package

```
> library(devtools)
> install_github("vankesteren/blim")
> library(blim)
```

How to blim

Anyone who can use the lm function in R can use blim! This is how to make a **blimfit** object:

```
> fit <- blim(formula = y~x1+x2,
              data = dataset)
```

It is easy to get your results:

```
> summary(fit)
```

The blimfit object

A blimfit object contains:

1. The **trace**: a matrix of all the sampled parameters
2. The **summary**: parameter estimates and credible intervals
3. The **priors**: the prior information about the parameters
4. The **data**: matrix representations of the input data

The example

Birthweight data. 189 babies were weighed. Their mothers were checked for **smoking** during pregnancy and history of **hypertension**.

	No Smoking	Smoking
No HT	µ1	µ2
HT	µ3	µ4

Informative Hypothesis

µ1 > µ2 > µ3 > µ4

The Model

I specify an **ANOVA** model as follows:

$$\text{bwt}_i = \mu_1 \cdot \text{no}_i + \mu_2 \cdot \text{sm}_i + \mu_3 \cdot \text{hy}_i + \mu_4 \cdot \text{smhy}_i + \varepsilon_i$$

In R, this translates to:

```
> mod1 <- blim(bwt~0+no+sm+hy+smhy, data = birthwt, iter = 99999,
              mtsprior = TRUE, method = "rmhs", burnin = 100, dtuning = 100)
```

And a second model with mother's weight (lwt) as a covariate. The parameter for this covariate has a nonconjugate cauchy prior.

```
> mod2 <- blim(bwt~0+no+sm+hy+smhy+scale(lwt), data = birthwt, iter = 99999,
              prior_b = c("dnorm(2950,402403)", "dnorm(2950,402403)",
                          "dnorm(2950,402403)", "dnorm(2950,402403)",
                          "dcauchy(0,100)"),
              method = "rmhs", burnin = 100, dtuning = 100)
```

Posterior Predictive Check

An important assumption of the ANOVA is the **normality of residuals**. blim can constructs a p-value for this using the kolmogorov-smirnov statistic as its discrepancy measure. The difference between expected and observed discrepancies is of interest. In R the observed discrepancy vector is calculated like so:

```
ObsD <- apply(trace,1,function(b){
    resid <- blimfit$y-blimfit$X%*%b
    D <- ks.test(resid,pnorm)$statistic
    return(D)
}
```

The p-value in the example is around 0.5, so no violation

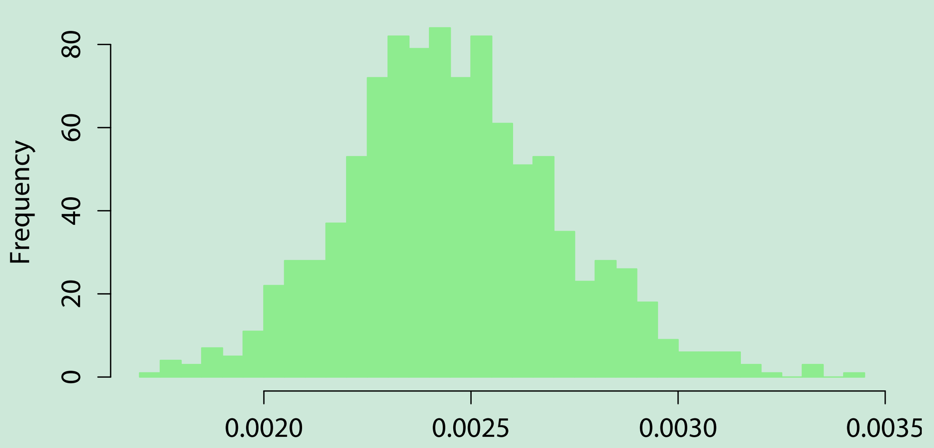
Model Selection

Unlike frequentist statistics, Bayesian analysis allows for model selection with **Bayes Factors**, quantifying the amount of evidence there is for model 1 relative to model 2. blim can calculate a Bayes Factor comparing two models using the **Laplace-Metropolis estimator**:

$$f(\mathrm{D}) \approx (2\pi)^{P/2}\,|\mathbf{H}^*|^{1/2}\,f(\theta^*)f(\mathrm{D}|\theta^*)$$

It can also indicate the precision of this estimate using a **bootstrap**:

```
> BF(mod1, mod2, bootstrap = TRUE)
```

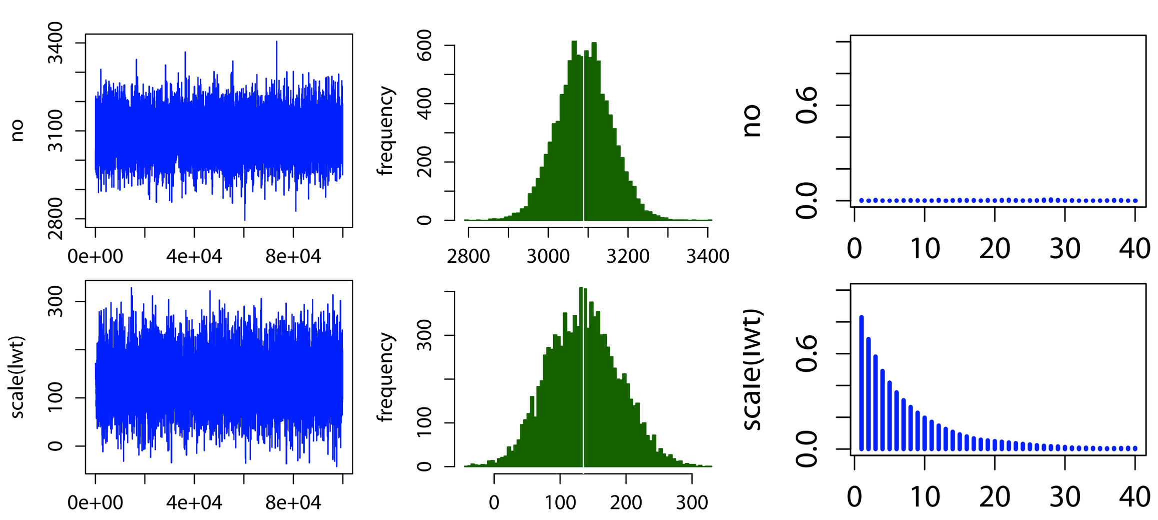


BF12 is around 0.0025. This indicates that model 2 is 400 times as likely as model 1. **DIC** agrees.

Convergence

Convergence and autocorrelation of µ1 and the covariate are shown below.

```
> cplots(fit)        > aplots(fit)
```



Dynamic Tuning

blim performs dynamic tuning of the Metropolis algorithm acceptance rate by optimising the variance of the proposal distribution:

```
# pcur is current acceptrate, popt=0.45
if (pcur > 0.5 || pcur < 0.15){
var[v] <- (var[v]*(1/pnorm(popt/2)))/
          (1/pnorm(pcur/2))
}
```

Conclusion

µ1: 3089, 95% CI [2957; 3221]
µ2: 2811, 95% CI [2645; 2978]
µ3: 2469, 95% CI [1946; 2992]
µ4: 2337, 95% CI [1696; 2971]

BF is the result of evaluating the inequality in the posterior (**fit**) and the prior (**complexity**):

```
# model in the form: "par[1]>par[2]"
f <- mean(apply(trace, 1,
function(par) eval(parse(text=model))))
c <- mean(apply(prior, 1,
function(par) eval(parse(text=model))))
BF <- f/c
```

My hypothesis has a Bayes Factor of around 12.5, which indicates **substantial evidence** in favour of this hypothesis!