Universiteit Utrecht

Bayesian Linear Models
The **blim** Package

Erik-Jan van Kesteren
Methodology & Statistics

```
> 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 Algorithm

blim's **rmhs** method uses **Gibbs sampling** for conjugate priors and a **random-walk Metropolis** algorithm for nonconjugate priors:

```
# bhat = current beta sample in trace
# fx = target (posterior) distribution
# var[v] is the tuning param for b[v]

# first sample candidate value:
bstar <- bhat[v]+rnorm(1, 0, var[v])

# then compute acceptance ratio
r <- min(1,fx(bstar)/fx(beta[i,v]))

# Assign to beta when accepted
if (runif(1) <= r){ bhat[v] <- bstar
} else { bhat[v] <- bhat[v] }
```

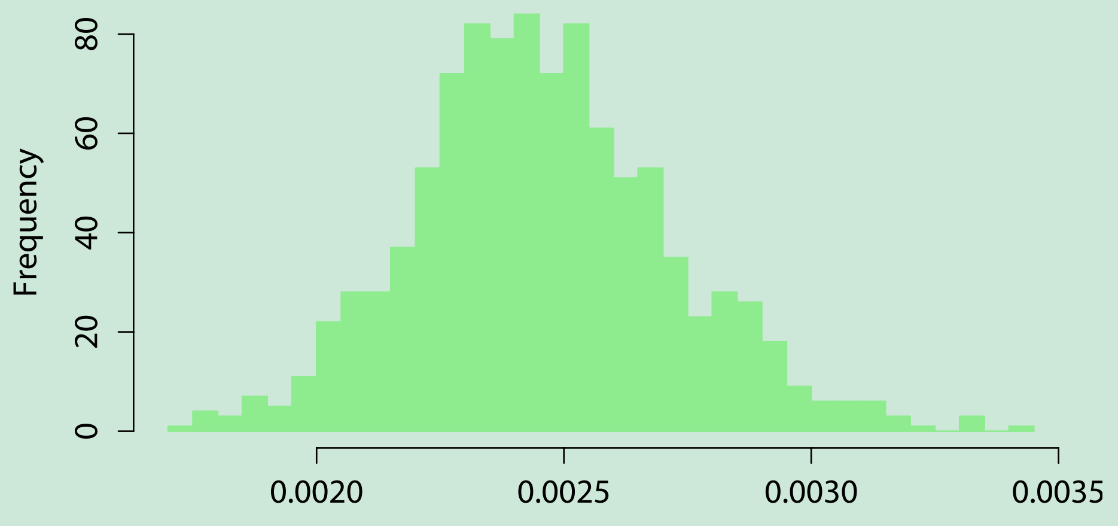
Model Selection

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.

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 Model

I specify an **ANOVA** model:

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

In R, this translates to:

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

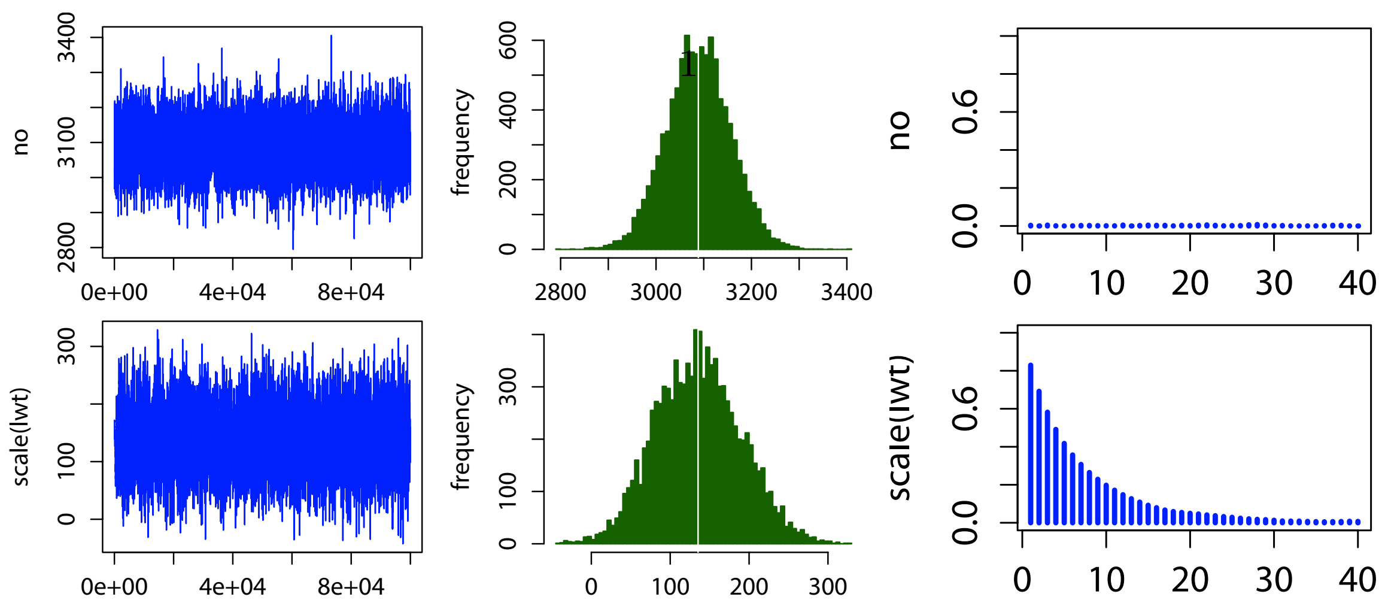
Model 2 with mother's weight (lwt) with nonconjugate cauchy prior:

```
mod2 <- blim(bwt~ ... + scale(lwt), ...
        prior_b = c( ...
                    ,"dcauchy(0,100)"),
        method = "rmhs", dtuning = 100)
```

Convergence

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

```
> cplots(mod2)      > aplots(mod2)
```



Dynamic Tuning

blim performs tuning of the Metropolis algorithm acceptance rate by dynamically 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))
}
```

The example

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

	No Smoking	Smoking
No HY	µ1	µ2
HY	µ3	µ4

Informative Hypothesis

µ1 > µ2 > µ3 > µ4

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

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 also the result of evaluating the hypothesis in the posterior (**fit**) and the prior (**complexity**):

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

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