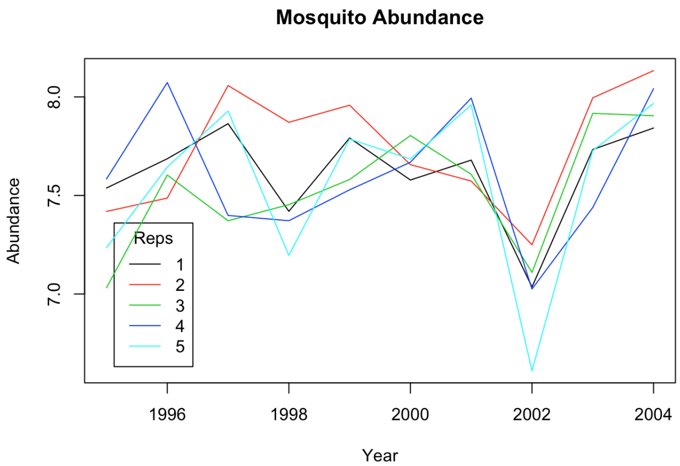
Exercise_10_BC

Betsy Cowdery

November 17, 2014

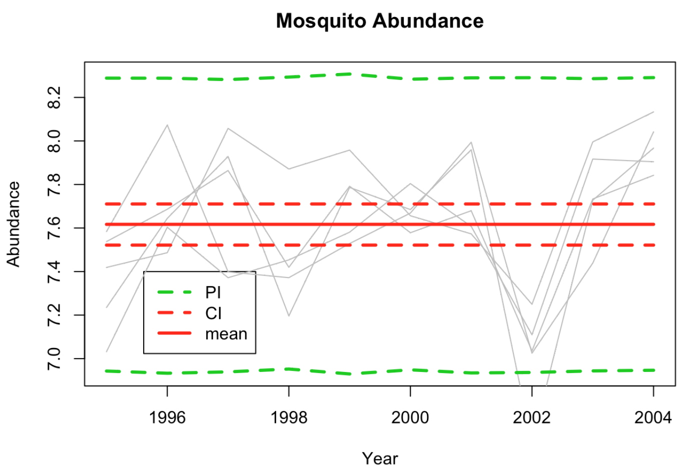
```
## Warning: package 'rjags' was built under R version 3.1.2
```

```
## Loading required package: coda
## Loading required package: lattice
## Linked to JAGS 3.4.0
## Loaded modules: basemod,bugs
```



First Fit

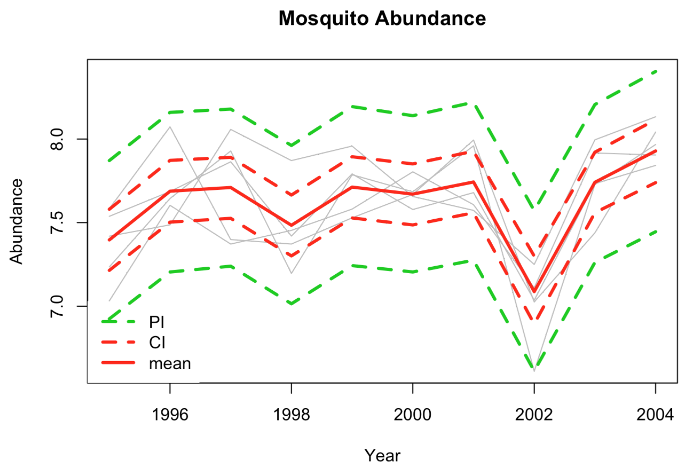
```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
##    Graph Size: 69
##
## Initializing model
```




```
## Warning: NAs introduced by coercion
```

Random time effect

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
##    Graph Size: 90
##
## Initializing model
```



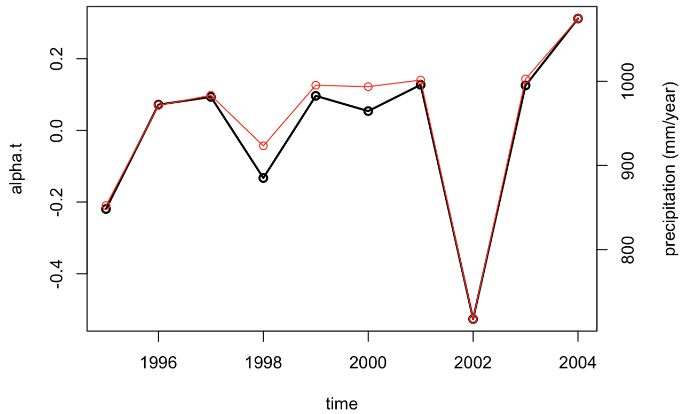

```
bmcmc2.df <- as.data.frame(as.matrix(bmcmc2))
mean(bmcmc2.df$tau.t)/(mean(bmcmc2.df$sigma) + mean(bmcmc2.df$tau.t))
```

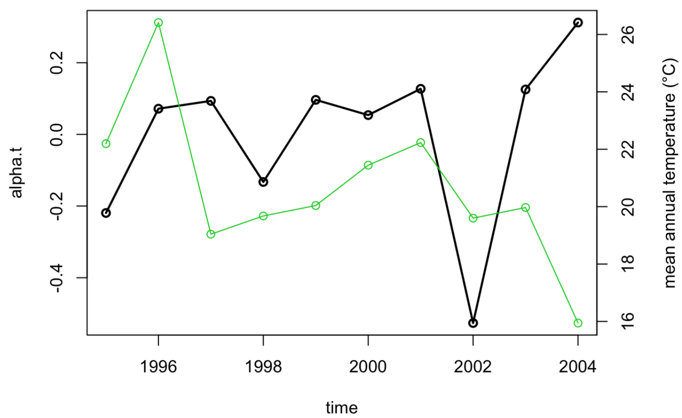
```
## [1] 0.628
```

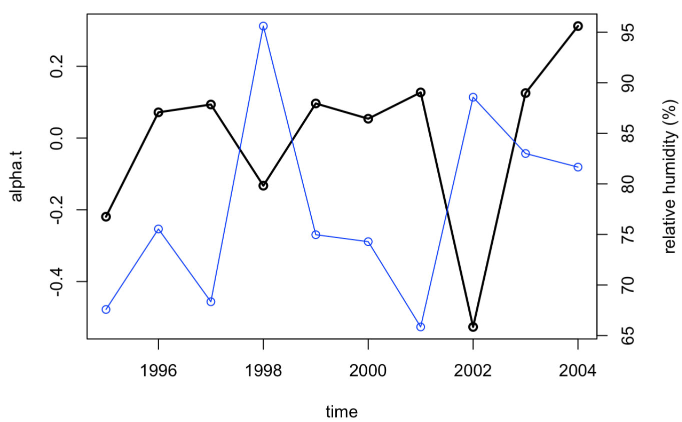
We can calculate the percentage of the variance in the mosquito densities that is explained by the year effects by dividing the mean variance of year effects by the total variance of the year effects and residuals. This gives us .63 so we can say that 63% of the variance in the mosquito densities is explained by the year effects.

Mixed Effects

```
## Warning: longer object length is not a multiple of shorter object length
```







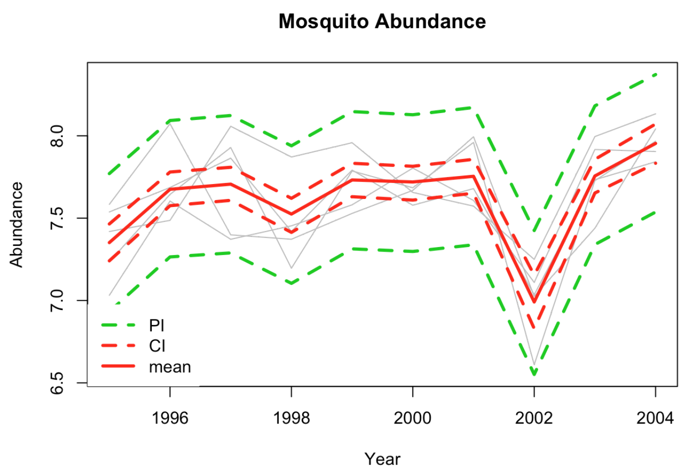
Clearly precipitation is the first variable worth exploring. Precipitation and alpha.t almost perfectly line up over time. With more time I would experiment with looking at mean annual temperature, but I doubt that it would explain much (if any) of the variance in mosquito densities.

```
model{
  for(i in 1:2){beta[i]~dnorm(0,.001)}       ## prior mean
  sigma ~ dgamma(0.001,0.001)   ## prior residual precision
  tau.t ~ dgamma(0.001,0.001)   ## prior year-effect precision

  for(t in 1:nt){           ## loop over years
    alpha.t[t] ~ dnorm(0,tau.t)     ## random year effect
    Ex[t] <- beta[1] + beta[2]*precip[t] + alpha.t[t]        ## process model
    for(i in 1:nrep){           ## loop over reps

      x[t,i] ~ dnorm(Ex[t],sigma) ## data model
    }
    px[t] ~ dnorm(Ex[t],sigma)  ## predictive interval
  }
}
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
##    Graph Size: 116
##
## Initializing model
```



DIC values for all 3 models

```
##         Mean deviance Penalized deviance
## Model 1         32.06            36.7806
## Model 2        -10.57             0.2171
## Model 3        -17.50           -12.5089
```

The model with the lowest DIC score “wins” - in other words, according to this model selection method, model 3 is the best model of the three.

Summary table with parameter means and CI for all 3 models

```
##            mu  2.5% 97.5%   sigma    2.5%   97.5%      tau      2.5%
## Model 1  7.617 7.521  7.71 0.12667 0.07631 0.16870       NA        NA
## Model 2  7.616 7.427  7.81 0.04883 0.03107 0.07655 0.082436 0.0230844
## Model 3     NA    NA    NA 0.04243 0.02814 0.06348 0.005246 0.0003699
##           97.5%
## Model 1      NA
## Model 2  0.23331
## Model 3  0.01223
```

Additional parameters only in Model 3

```
##    beta1     2.5%    97.5%    beta2     2.5%    97.5%
## 5.053549 4.374059 5.731133 0.002692 0.001985 0.003400
```