Example of Bayesian Analysis

Carl Schwarz

2017-05-15

Contents

1 Introduction

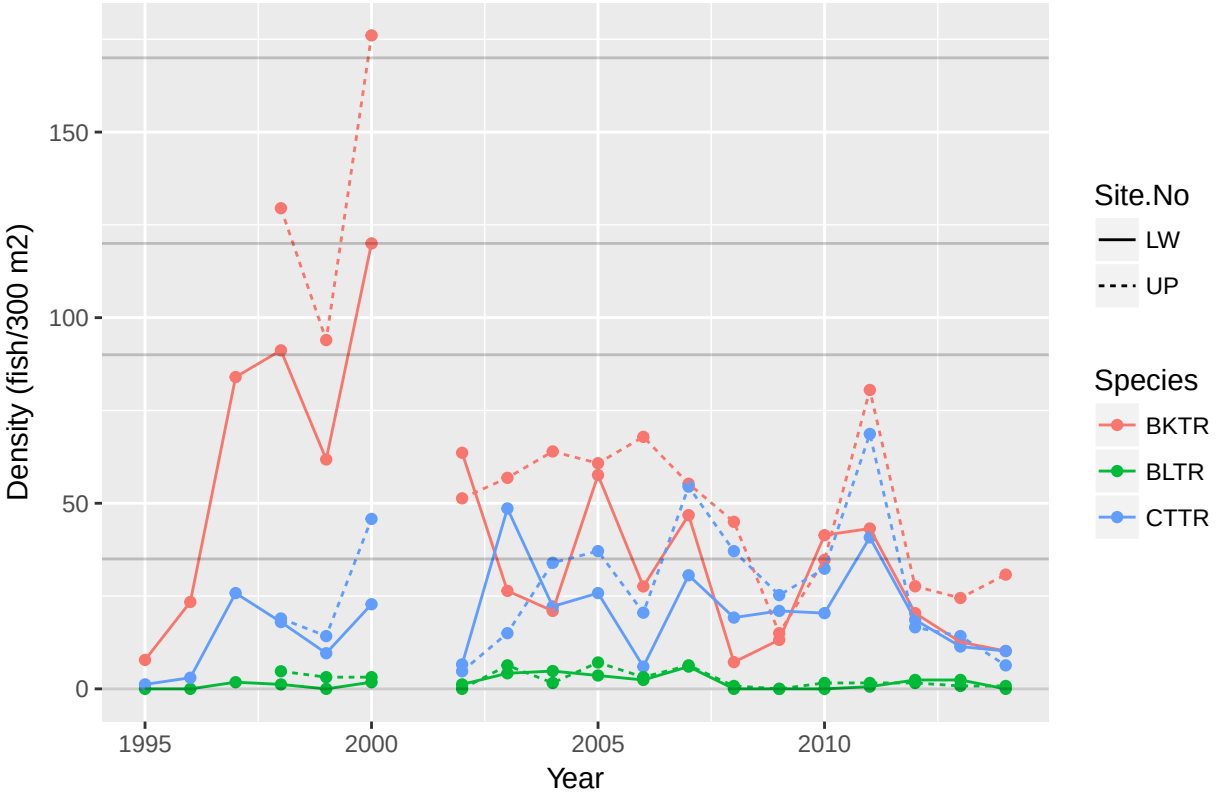
The Fish Sustainability Index (FSI) is Alberta Fish and Wildlife's method of assessing fish stocks on a provincial scale. One of the components of this index is Population Integrity which is partially assessed using population density. Density comparisons are referenced to what the watershed in question could produce if it had no human impacts and consisted of the most ideal habitat in Alberta. The observed density is ranked on six point scale (0 to 5) from Functionally Extirpated to Low Risk. For example, Table 1 presents draft guidelines for BKTR based on observed catch per unit effort from one pass electrofishing.

Table 1: Table 1 . FSI categories for BKTR

Species Code	FSI Category	Lower bound	Upper bound
BKTR	VHR	0	35
BKTR	HR	35	90
BKTR	MR	90	120
BKTR	LR	120	170
BKTR	VLR	170	3000

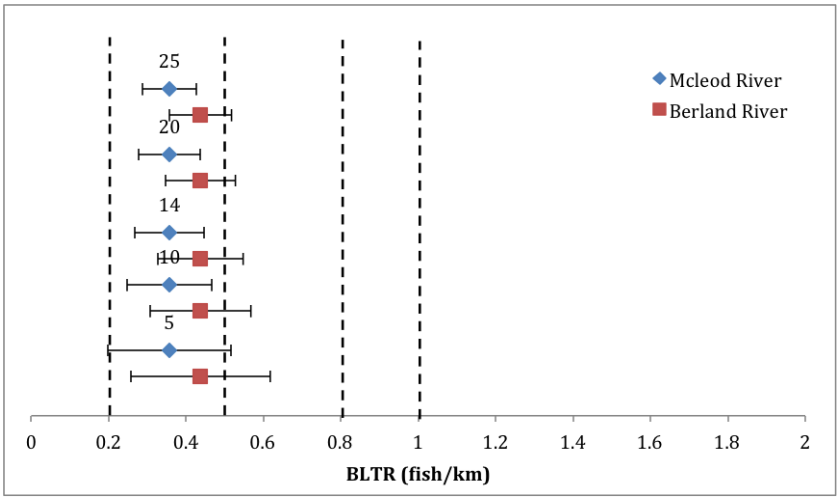
Quantitative classification of a given population occurs by sampling a watershed at numerous sites (reaches) and observing the CPUE. However data can be very noisy. Consider, for example, a plot of CPUE for three species at Quirk Creek in Figure 1.

Figure 1. Raw data for Quirk Creek with FSI categories



In some situations, classification is straightforward. For example, if the (very) simplistic assumption is made that the FSI categories are identical for all species, then most readers would clearly place the BLTR (green lines) in the VHR category in all years. However, what should be done with CTTR. In some years, some sites are above the threshold between VHR and HR, while in other years, both values are below the threshold. By making a single classification solely based on the mean or median ignores the variability in the data and the uncertainty in these estimates.

Similarly, consider Figure 2 extracted from results in the Macleod and Berland Watershed. While the mean CPUE all fall within the HR FSI category, the uncertainty in the mean CPUE is large enough that it may actually fall in any of the three categories.



Rather than trying to arbitrarily choose a single FSI category and ignore uncertainty, it would be useful to

view the assignment as a probabilistic task. For example, we would like to make statements such as "We are 80% certain that this watershed is in the HR category".

This gives rise to a number of issues that need to be resolved:

(a) How can such an assessment be done?

(b) How can these assessments be tracked over time?

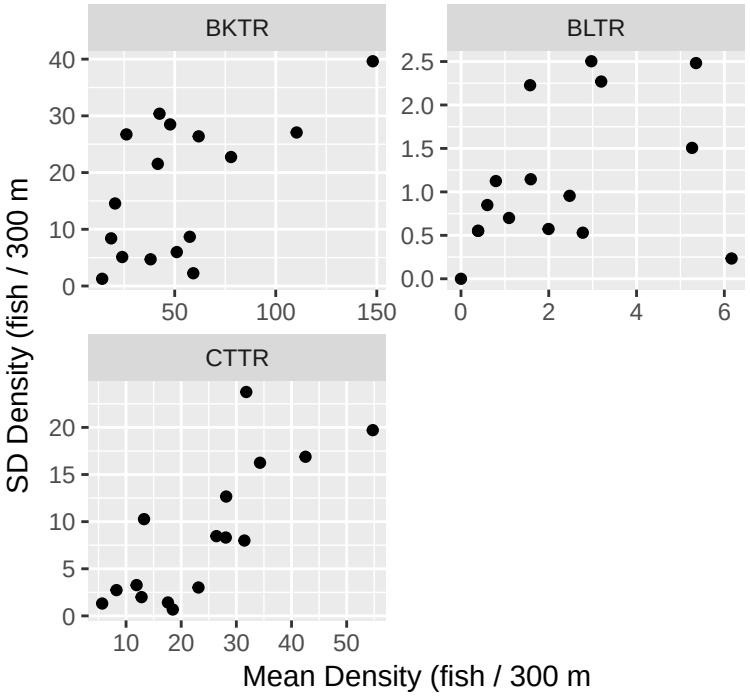
(c) How many sites must be measured so that you are xx% certain that each watershed is in its appropriate risk category?

2 Principles of the analysis

2.1 Using MEDIANS rather than MEANS

Plots of the standard deviation (between the sites in a year) vs. the mean of the sites in a year generally shows a pattern where the standard deviation increases with the mean such as seen in Figure 3 for Quirk Creek:

Figure 3. SD vs. Mean for Density



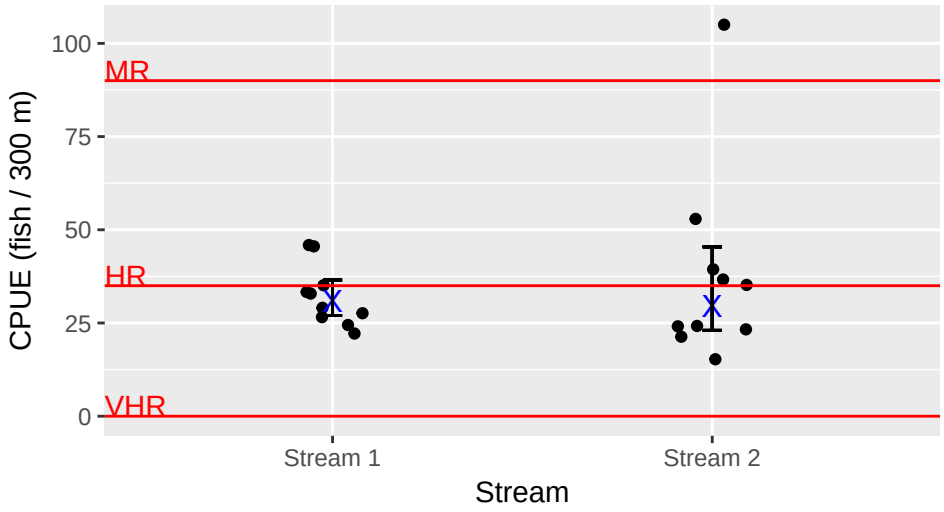
This is quite common when measuring abundance and indicates that a log-normal distribution will be good description of the distribution of measured densities among sites within a year. One of the properties of the log-normal distribution is that large outliers are quite common. To account for the impact of these large outliers on the mean across the sites in a year, the MEDIAN is preferred measure of overall trend and of classification to the FSI category. The median is the point where 50% of the values (measurements at sites) are predicted to be above and 50% of the values are predicted to be below.

If the underlying distribution of readings follows a log-normal distribution, then the median can be estimated by the geometric mean of the raw observations.

2.2 Probabilistic assignment of categories

Consider Figure 4 of two (simulated) CPUE data sets for two different streams (each sampled for a single year) along with the FSI category boundaries. The "X" indicates the median of the dataset and the error bars show 95% confidence intervals for the median of the data.

Figure 4. Initial plot of CPUE for two streams



We would like to make statements about the category (VHR, HR or MR) to which each stream belongs. In both streams, the median CPUE (blue X) are in the VHR category, but it does not seem sensible to definatively rank both streams in the VHR category just because their respective medians fall in this category. Indeed, the data for Stream 2 is more variable which results in the 95% confidence interval for the median for Stream 2 being wider than the corresponding interval for Stream 1. In some sense, we should have a stronger belief that the median CPUE for Stream 1 is in the VHR category that for Stream 2, but there is some belief that each stream could also be in HR category.

It seems natural to ask – what is the "probability that the median density is in each category". This cannot be answered with classical statistics (mean and confidence intervals) because the actual category membership is fixed (non probabilistic) in any one year. A Bayesian analysis comes to the rescue. A Bayesian would change the question slightly to "what is the belief that the median density is in each category". The change is crucial, because belief is naturally expressed in a probabilistic framework.

An intuitive explanation for the process is as follows. For each year, estimate the parameters that describe that year's distribution of density across the sites. If a log-normal distribution is assumed, the parameters desribe the distribution are the log(median density) and the standard deviation on the log-scale across ALL possible sites. These values are estimated based (on this case) data from each stream. The sampling distribution for the log(median) is estimated. A sampling distribution gives the distribution of plausible values for the log(median) for each stream. [A confidence interval can be computed from this sampling distribution but is not used.]. A Bayesian then says that the sampling distribution is interpreted as your belief in the distribution of the log(median) for all sites. Consequently, compute what fraction of the sampling distribution lies between the FSI boundaries and that is your belief (probability) that the median density is in each category.

The actually fitting process cannot be done by hand and requires a method called MCMC to do the Bayesian analysis. As part of the output from an MCMC analysis, quantities such as the probability (belief) that the median is in each category are easily found.

A common language to describe Bayesian models is BUGS. There are several computer programs (WinBugs, OpenBugs, JAGS, etc.) that take the Bayesian model described by BUGS and perform a Monte Carlo Markov Chain (MCMC) analysis to estimate the posterior distributon of the parameters of interest. We use JAGS

which is called from R (an open source statistical software package).

For example, here is the BUGS code to do a probabilistic assignment of each stream to the FSI category above:

```
cat(file="model1.txt", "
    ############################################################
    # Input data is
    #   Ndata   - number of data points measured for the stream
    #   Density   - the Density as measured by the CPUE at each site

    #   NFSI      - number of FSI categories
    #   FSI.lower - the upper and lower bounds of the FSI
    #   FSI.upper   categories
    ############################################################

  model {

    # likelihood - log normal distribution of cpue density values
    for(i in 1:Ndata){
       Density[i] ~ dlnorm(log.median, tau)
    }

    # prior distribution for log.median and tau
    # tau is 1/sd
    tau <- 1/(sd.log*sd.log)
    sd.log ~ dunif(.05, 3)   # on the log-scale sd is proportion of the mean

    # priors for the log.median
    log.median ~dnorm( 0, .00001) # virtually no information in prior

   # derived variables.
   # median.den is antilog of  log.medianl
   median <- exp(log.median)

   # probability of median being in each threshold category
   for(k in 1:NFSI){
      prob.FSI.cat[k] <- ifelse((median >= FSI.lower[k]) && (median < FSI.upper[k]),1,0)
   }
}
") # End of the model
```

All Bayesian model have two components.

First, is the likelihood, i.e. what is the probability distribution of the data? In this case, we are assuming a log-normal distribution (`dlnorm`). Note that in the BUGS langauge, the second parmaeter (denoted as `tau`) represents 1/variance. The two parameters that describe this log-normal distribution are the log(median) (`log.median`) and the standard deviation on the log-scale (`SD.log`). The `for` loop assumes that all of the data from this stream comes from the same log-normal distribution.

Second is the prior distribution for the unknown parameters. Here we chose relatively uninformative priors for both parameters.

During each interation in the MCMC process, the estimates of the parameters (the `log.median` and `sd.log`) are updated using standard Bayesian methods. This updating process generates a sample from the posterior distribution as shown later.

We can also compute derived variables for each interation of the MCMC process. In this case, we take the anti-logarithm of the `log.median` and then use the estimated median to see in which FSI category it lies. For each iteration, if the estimated median lies between the lower and upper bound of the FSI category, it generates a 1 otherwise a 0. This generated string of 0's and 1's is used to estimate the probability of falling in each FSI category as explained later.

We now set up *R* variables to hold the data being passed to JAGS (the data.list), the initial values for the MCMC chains (we leave these blank and let JAGS generate them), and which variable we want posteriors to be estimated (the monitor.list).

```
data.list <- list(Ndata      =length(stream1.cpue$cpue),
                  Density    =stream1.cpue$cpue,
                  NFSI       =nrow(FSI.threshold),
                  FSI.lower  =FSI.threshold$lower,
                  FSI.upper  =FSI.threshold$upper)
data.list
```

```
$Ndata
[1] 10

$Density
 [1] 27.6 45.6 32.9 26.6 24.5 45.9 29.1 35.1 22.2 33.3

$NFSI
[1] 5

$FSI.lower
[1]   0  35  90 120 170

$FSI.upper
[1]   35   90  120  170 3000
```

```
# Next create the initial values.

init.list <- list(
   list(), list(), list()
)

# Next create the list of parameters to monitor to get posterior
#
monitor.list <- c("log.median", "sd.log","median","prob.FSI.cat")
```

Now we call JAGS using the `jags()` function from the `R2JAGS` package:

```
# Finally, the actual call to JAGS
set.seed(4534534)  # intitalize seed for MCMC

results <- jags(
  data      =data.list,   # list of data variables
  inits     =init.list,   # list/function for initial values
  parameters=monitor.list,# list of parameters to monitor
  model.file="model1.txt",  # file with bugs model
  n.chains=3,
  n.iter  =5000,          # total iterations INCLUDING burn in
  n.burnin=2000,          # number of burning iterations
  n.thin=2,               # how much to thin
  DIC=TRUE,               # is DIC to be computed?
```

```
  working.dir=getwd()     # store results in current working directory
)
```

```
## module glm loaded
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 10
##    Unobserved stochastic nodes: 2
##    Total graph size: 82
##
## Initializing model
```

The output from the call to JAGS (the `results` object) is a complex object that has many components. We will look at some of them:

There are several thousand samples generated for each posterior. Here are some of the samples from the posterior distributions (extracted from `results$BUGSoutput$sim.matrix`) for the `log.median` and the `median`.

```
     log.median median
[1,]       3.61  37.03
[2,]       3.31  27.37
[3,]       3.38  29.23
[4,]       3.45  31.60
[5,]       3.43  30.87
```

For each iteration in the MCMC process, a value for the `log.median` and the `median` is generated as shown above. These values form the posterior sample for their respective parameters.

Then for each of the posterior samples from the median, a 0/1 indicator variable is created for each FSI category:

```
     prob.FSI.cat[1] prob.FSI.cat[2] prob.FSI.cat[3] prob.FSI.cat[4] prob.FSI.cat[5]
[1,]               0               1               0               0               0
[2,]               1               0               0               0               0
[3,]               1               0               0               0               0
[4,]               1               0               0               0               0
[5,]               1               0               0               0               0
```

In this case, 4/5 of the posterior samples for the median are in the first FSI category, and 1/5 of the posterior samples for the median are in the second fsi category. This would correspond to a 80% posterior belief that the median is in the first FSI category and a 20% posterior belief that the median is in the second FSI category.

Of course, we look at the averages of the entire set of posterior samples (a grand total of 4500 samples in each posterior sample). The mean and standard deviation for the `log.median` and `median` are:

```
            mean   sd  2.5% 97.5%
log.median  3.45 0.09  3.26  3.64
median     31.55 2.98 26.00 38.07
```

The estmated median of the CPUE for stream 1 is 31.55 with a 95% credible interval of (26 - 38.07).

Finally, the posterior belief that the median is in each FSI category is

```
Probability of Stream 1 in each FSI category
```

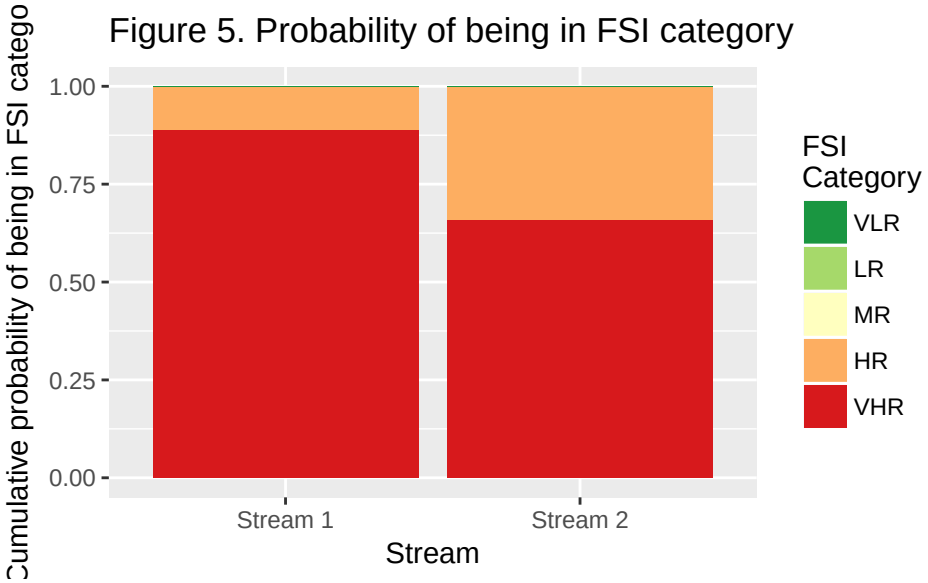
```
                Probability
prob.FSI.cat[1]        0.89
prob.FSI.cat[2]        0.11
prob.FSI.cat[3]        0.00
prob.FSI.cat[4]        0.00
prob.FSI.cat[5]        0.00
```

The same set of computation can be done for Stream 2 and we obtain the following results for Stream 2.

```
Probability of Stream 2 in each FSI category
```

```
                Probability
prob.FSI.cat[1]        0.66
prob.FSI.cat[2]        0.34
prob.FSI.cat[3]        0.00
prob.FSI.cat[4]        0.00
prob.FSI.cat[5]        0.00
```

As expected, we see a shift in the posterior beliefs for Stream 2 vs Stream 1. The probabilistic assignments can be graphed using the cumulative probability of being in each category as shown in FIgure 5.



3 Multiple years of data

The previous section showed how a probabilistic assessment could be made for a single year of data for a watershed. Naively, you could do a similar assessment for each year. However, there are two problem with such a naive analysis.

(a) Some sites are repeatedly measured over time. We saw in the Quirk Creek examples above that the same two sites were repeatedly measured over time. This implies that the data values across years are not independent from each other. For example, one site (due to local site-specific effects) could tend to always have a higher than average density across years. Any analysis must account for these repeated measurements over tme

(b) There may be considerable year-specific effects (process error). Year-specific effects tend to push the densities in all sites up or down in a particular year due to effects typically that cannot be controlled. For example, a specific year could be warmer than usual leading to warmer water temperatures and affecting the efficiency of the electrofishing in all sites in a year. These year-specific effects could cause

the probabilistic assessment to vary considerably from year to year due to random events that are unrelated to the long-term trends in the data. The related document on the number of years and sites needed to detect certain trends has a fuller discussion of year-specific effects (process error).

Statistical models can be developed to fit trend over time and deal with the above two problems. Such models are linear mixed models (LMM) and take the form (in standard *R* syntax): `log(Density) ~ Year + YearC(R) + Site(R)` where the `log(Density)` is the logarithm of the observed density from electrofishing, `Year` represents the linear trend over time in the log(median); `YearC(R)` represents the (random) year-specific effects; and `Site(R)` represents the (random) site-specific effects.

Both the year-specific and site-specific effects are assumed to follow a Normal distribution with mean 0 and year-specific and site-specific standard deviations.

A similar Bayesian model can be developed that accounts for year-specific and site-specific effects.

```
cat(file="model2.txt", "
    ################################################################
    # Input data
    #      Ndata  - number of data points
    #      Density- density for each data point
    #      Site.num- site number for each data point (1....NSites)
    #      Year.num- year number for each data point (1... Nyears)

    ################################################################

    # compute the number of years (1...) and number of sites
data {
       Nyears   <- max(Year.num)
       Nsites   <- max(Site.num)
     }

model {

    # compute the trend line
    for(i in 1:Nyears){
          log.median.trend[i] <- beta0 + beta1*i
    }

    # add the site-effects and year-effects to the trend
    for(i in 1:Ndata){
       log.median.data[i] <- log.median.trend[Year.num[i]] +
                     site.eff[Site.num[i]] +
                     year.eff[Year.num[i]]
       Density[i] ~ dlnorm( log.median.data[i], tau)
    }

    # tau is 1/(sd.log * sd.log)
    tau <- 1/(sd.log*sd.log)
    sd.log ~ dunif(.05, 3)   # on the log-scale sd is proportion of the mean

    # priors for the intercept and slope
    beta0 ~ dnorm(0, .001)
    beta1 ~ dnorm(0, .001)

    # random effect of Year
    for(i in 1:Nyears){
```

```
          year.eff[i] ~ dnorm(0, tau.year.eff)
    }
    tau.year.eff <- 1/(sd.year.eff*sd.year.eff)
    sd.year.eff  ~ dunif(.01,2)

    # random effect of Sites
    for(i in 1:Nsites){
          site.eff[i] ~ dnorm(0, tau.site.eff)
    }
    tau.site.eff <- 1/(sd.site.eff * sd.site.eff)
    sd.site.eff  ~ dunif(.01, 2)

    # what is the probability that the trend is negative
    p.beta1.lt.0 <- ifelse(beta1<0,1,0)

   # derived variables.
   for(i in 1:Nyears){
      med.den.trend [i] <- exp(log.median.trend  [i])
      med.den.data  [i] <- exp(log.median.data   [i])
   }

   # probability of being in a threshold category for trend line
   for(i in 1:Nyears){
      for(k in 1:NFSI){
          prob.FSI.cat.trend[i,k] <- ifelse((med.den.trend[i] >= FSI.lower[k]) &&
                                            (med.den.trend[i] <  FSI.upper[k]),1,0)
          prob.FSI.cat.data [i,k] <- ifelse((med.den.data [i] >= FSI.lower[k]) &&
                                            (med.den.data [i] <  FSI.upper[k]),1,0)
      }
   }
}
") # End of the model
```

We need to recode the calendar years into values from 1 (for the earliest year) to xx (for the maximum year) in the data. Because JAGS cannot process character data, we also need to create a numeric code for the sites. The `data.list` has the coded year and site numbers and the `site.code` and `year.code` data frames have the translation between the site and year codes and the actual site and year values.

We will the trend model to the CTTR species data.

```
# Create the data list

pass.cttr <- pass.melt[ !is.na(pass.melt$Density) & pass.melt$Species=="CTTR",]

year.code <- data.frame(Year    =sort(unique(pass.cttr$Year)),
                        Year.num=1:length(unique(pass.cttr$Year)))
year.code
```

```
##    Year Year.num
## 1  1995        1
## 2  1996        2
## 3  1997        3
## 4  1998        4
## 5  1999        5
## 6  2000        6
```

```
## 7  2002        7
## 8  2003        8
## 9  2004        9
## 10 2005       10
## 11 2006       11
## 12 2007       12
## 13 2008       13
## 14 2009       14
## 15 2010       15
## 16 2011       16
## 17 2012       17
## 18 2013       18
## 19 2014       19
```

```
pass.cttr <- merge(pass.cttr, year.code)
head(pass.cttr)
```

```
##   Year Watershed Site.No Species Density Year.num
## 1 1995     Quirk      LW    CTTR    1.20        1
## 2 1996     Quirk      LW    CTTR    3.00        2
## 3 1997     Quirk      LW    CTTR   25.80        3
## 4 1998     Quirk      LW    CTTR   18.00        4
## 5 1998     Quirk      UP    CTTR   18.96        4
## 6 1999     Quirk      LW    CTTR    9.60        5
```

```
# Convert Site.no to a unique numeric values
site.code <- data.frame(Site.No  =unique(pass.cttr$Site.No),
                        Site.num =1:length(unique(pass.cttr$Site.No)),
                        stringsAsFactors=FALSE)
site.code
```

```
##   Site.No Site.num
## 1      LW        1
## 2      UP        2
```

```
pass.cttr <- merge(pass.cttr, site.code)


pass.cttr <- pass.cttr[ order(pass.cttr$Year.num, pass.cttr$Site.num),]


data.list <- list(Ndata      =nrow(pass.cttr),
                  Year.num   =pass.cttr$Year.num,
                  Year       =pass.cttr$Year,
                  Site.num   =pass.cttr$Site.num,
                  Density =pass.cttr$Density+.1*min(pass.cttr$Density[pass.cttr$Density>0]),
                  NFSI       =nrow(FSI.threshold),
                  FSI.lower  =FSI.threshold$lower,
                  FSI.upper  =FSI.threshold$upper)
data.list
```

```
## $Ndata
## [1] 35
##
## $Year.num
##  [1]  1  2  3  4  4  5  5  6  6  7  7  8  8  9  9 10 10 11 11 12 12 13 13 14 14 15 15 16 16 17 17 18
##
```

```
## $Year
##  [1] 1995 1996 1997 1998 1998 1999 1999 2000 2000 2002 2002 2003 2003 2004 2004 2005 2005 2006 2006 1
##
## $Site.num
##  [1] 1 1 1 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2
##
## $Density
##  [1]  1.32  3.12 25.92 18.12 19.08  9.72 14.34 22.92 45.90  6.72  4.86 48.72 15.12 22.32 34.08 25.92
## [33] 14.34 10.32  6.45
##
## $NFSI
## [1] 5
##
## $FSI.lower
## [1]   0  35  90 120 170
##
## $FSI.upper
## [1]   35   90  120  170 3000
```

```
# Next create the initial values.
# If you are using more than one chain, you need to create a function
# that returns initial values for each chain.

init.list <- list(
   list(), list(), list()
)

# Next create the list of parameters to monitor.
# The deviance is automatically monitored.
#
monitor.list <- c("log.median.data","log.median.trend",
                  "med.den.data","med.den.trend",
                  "site.eff","year.eff",
                  "sd.log","sd.year.eff","sd.site.eff",
                  "prob.FSI.cat.trend","prob.FSI.cat.data",
                  "beta0","beta1","p.beta1.lt.0")
```

We again fit the model in JAGS using `R2JAGS` and save the MCMC output in the `result.cttr` list.

```
set.seed(4532234)  # intitalize seed for MCMC

results.cttr <- jags(
  data      =data.list,   # list of data variables
  inits     =init.list,   # list/function for initial values
  parameters=monitor.list,# list of parameters to monitor
  model.file="model2.txt",  # file with bugs model
  n.chains=3,
  n.iter  =5000,          # total iterations INCLUDING burn in
  n.burnin=2000,          # number of burning iterations
  n.thin=2,               # how much to thin
  DIC=TRUE,               # is DIC to be computed?
  working.dir=getwd()    # store results in current working directory
)
```

```
## Warning in jags.model(model.file, data = data, inits = init.values, n.chains = n.chains, : Unused va
```

```
## Compiling data graph
##    Resolving undeclared variables
##    Allocating nodes
##    Initializing
##    Reading data back into data table
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 35
##    Unobserved stochastic nodes: 26
##    Total graph size: 2413
##
## Initializing model
```

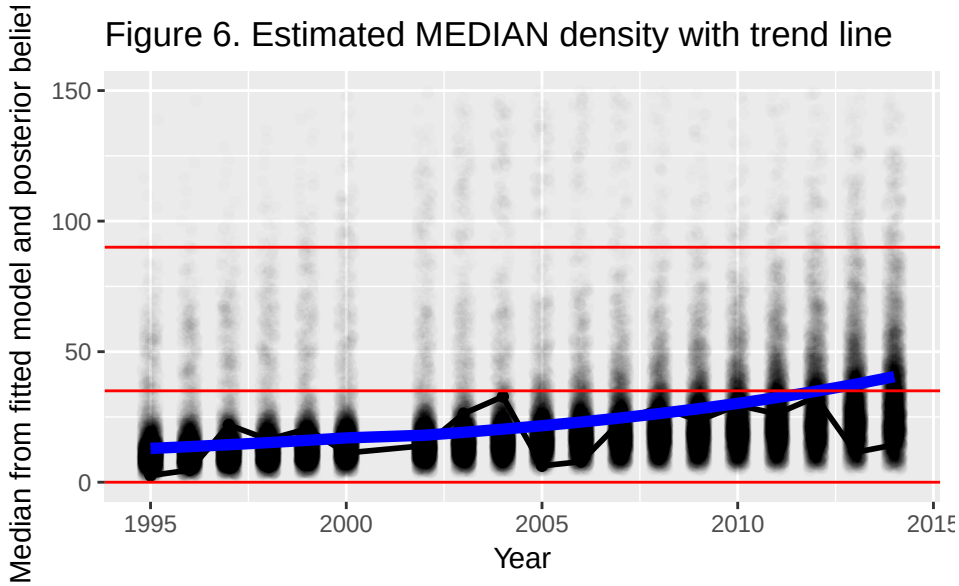
In exactly the same way, a sample from the posterior of the estimated median based on the underlying trend can be found using MCMC methods. Each value from the posterior sample from the median can be compared to the FSI categories and the probabilistic assessment can be computed based on the underlying trend. This should be more stable because the year-specific effect have been "removed" before the probabilistic assessment is made.

The estimated slope, a measure of uncertainty of the estimate, and the posterior belief that the slope is negative can be extracted from the MCMC output:

```
##        slope         sd p.slope.lt.0
## 1 0.05581604 0.03565525        0.052
```

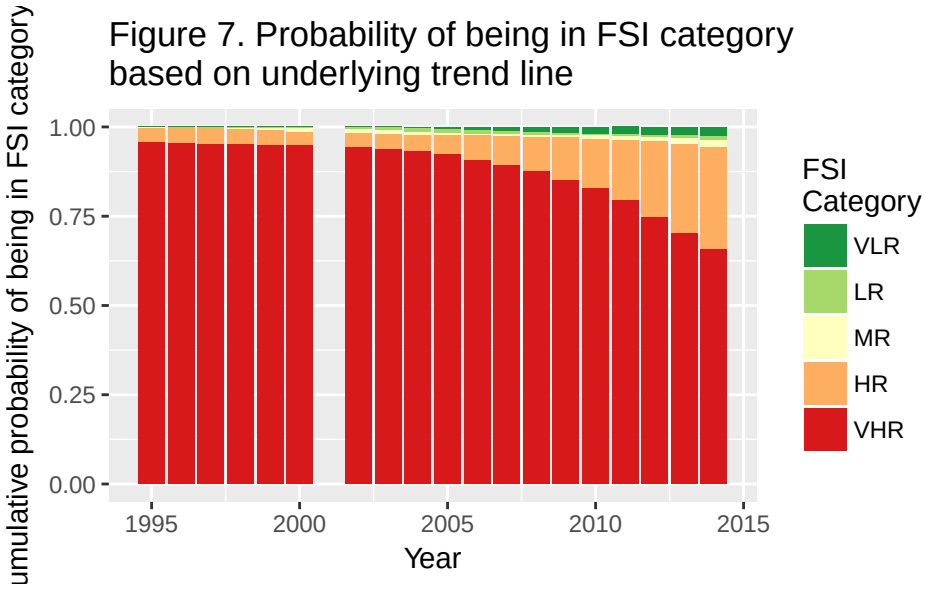
The estimated slope of the trend line for the log(median) is 0.056 (SE 0.036) and the posterior belief that the slope is POSITIVE is 0.948.

We can create plots of the posterior density of the median density for each year in the study as shown in Figure 6.



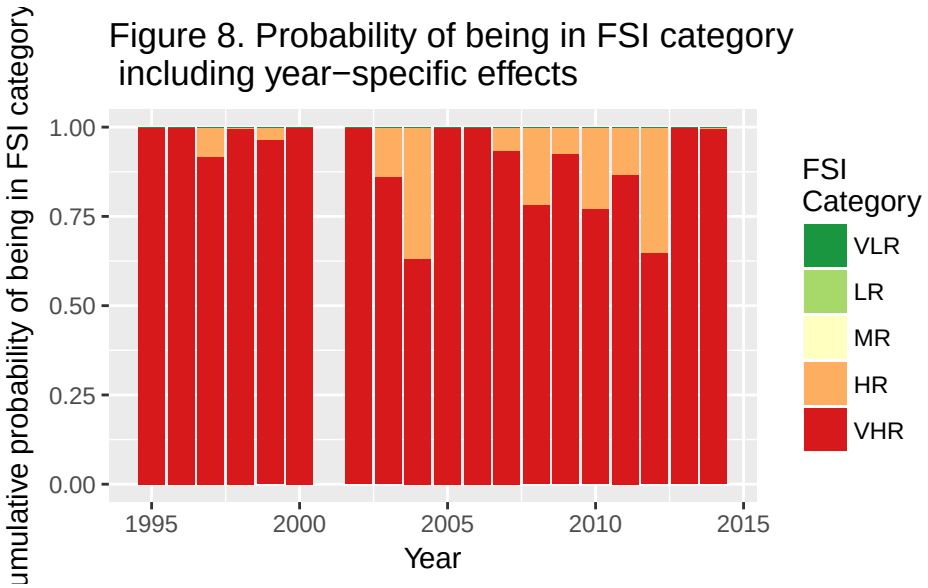
The blue line is the estimated trend in the median while the black line is the estimated median in each year adjusting for the sites that were sampled and potential process error. In this case, there were 2 years around 2013 where the yearly median declined, but this also happened aroung 2005 so may not be unusual.

Similarly, we can estimate the probability of belonging to each FSI category over time as shown in Figure 7.



We see with the estimated increasing trend in the median, that the probability has declined in the VHR fish category.

THe impact of process error (year-specific effects) can be seen by looking at the FSI categorization without the smoothing trend applied as shown in Figure 8.



We can see from this figure, that year-specific effects (process error) lead to much variability in the FSI categorization due to factors not under the control of the program.

In this study, the trend line is fit using the entire data series. However, you may feel that this is too rigid and want some flexibility. A cubic spline could be fit instead where the fit of the spline for a particular year depends more on the years surrounding the data point that on years that are very far away (in time) from the particular year. This has not yet been implemented.

4 How much sampling is needed?

The question of sample size can be broken into two (separate) questions

1. How many years of sampling and how many sites must be sampled to detect a trend? This is covered in a companion document.
2. How many sites must be sampled in a particular year to be reasonably sure that the FSI category is known with a high probability.

This sections will consider the latter question.

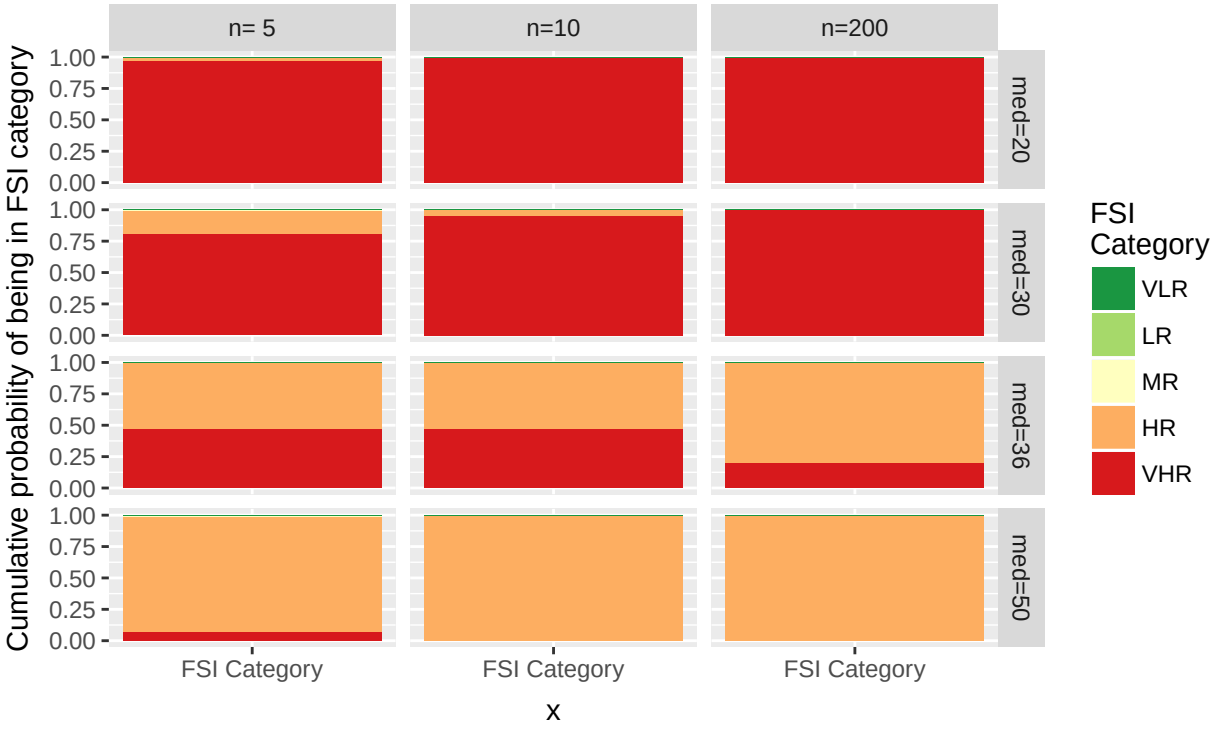
The posterior probability that a site is classified into an FSI category depends on the value of the median and the standard deviation of the values for a particular year. More technically, it depends on the $log(median)$ and the standard deviation of the $log(CPUE)$. For example, if the median is close to the middle of a FSI category, then fewer samples will be needed to determine the FSI category with high probability than if the median is close to a boundary. Similary, if the standard deviation of the CPUE is high, then more samples are needed to estimate the median with enough precision so that most of the confidence interval lies within the FSI category (i.e. the log(median) has a small standard error) than if the standard deviation is small.

There is no easy way to determine the required sample size in a year other than using the Bayesian model with simulated data based on the median and standard deviation of the CPUE at various sample sizes. The Bayesian model used is the same one as used for the analysis of the two streams presented earlier.

For example, using the Quirk Creek data shown earlier, the average standard deviation of the log(CPUE) for CTTR is 0.35 which estimates the coefficient of variation of individual values around the median.

This average standard deviation was used to estimate the probability of belonging to the two worst FSI categories with different values of the median. The cutoffs for the two worst FSI categories are 0, 35, and 90. We used all combinations of values for the median of 20, 30, 36 and 50 and sample sizes of 5, 10 and 200 to compute the posterior probability of belonging to the FSI categories as shown in Figure 9.

Figure 9. Probability of being in FSI category in one particular year from simulated data with cv around median of 0.35



In this case, the median values are in FSI categories 1, 1, 2 and 2 respectively. The plot shows that for median values that are far from the FSI category boundaries, relatively small numbers of samples are needed to ensure that the correct categorization is assigned with high probability, while for values of the median close to a boundard, larger samples sizes (number of sites) will be needed to ensure that the correct categorization is assigned.

This is not an unexpected result. In practise, the value of the median will not be known in advance even if information about the coefficient of variation (the standard deviation of the log(CPUE)) is known from past studies or from similar sites. It would seem prudent to plan sample sizes based on the median being more than 5 units away from a category boundary for planning purposes.

CAUTION: The above analysis is the sample size within a year to categorize A PARTICULAR year into the FSI category. However, the results for a year are HEAVILY INFLUENCED by year-specific factors (the process error) and so the assignment to an FSI category based on each year's results may result in high veriability in the assignments across years (as shown earlier) which makes it difficult to assess long-term trends. For this reason, assessment of the TREND in FSI categorization should be based on the trend line (which 'removes' year-specific effects) rather than inidividual yearly assessments. As noted in a companion document, the standard deviation of the year-specific effects (the process error) is often the limiting factor to determining trend. In these cases, assessment of the FSI category for trend is relatively insensitive to the number of sites sampled per year, and depends heavily on the total number of years of sampling rather than the amount of sampling in each year.