# Fusing Times-Series Data: Tree Rings and Forest Inventory

In this exercise we will extend the state-space framework to combine multiple data streams with different observation errors and to separate observation error from process error. We will also demonstrate how to add hierarchical random effects to partition the process error into multiple sources.

Specifically, we will be building up to the model presented by Clark et al. 2007 Ecological Applications that combines tree ring data with forest inventory data. Unlike the original model, which was written all in R, we will rewrite this model into JAGS, which makes it easier to see what is going on and to modify the model. In this exercise we will utilize data from a collection of small plots at the Harvard Forest, Petersham, MA.

We will divide this analysis into a number of steps, which we will encapsulate into functions to make them easier to understand and run. Thus we will begin by defining these functions. Specifically, the steps will be:

1. load forest inventory data
2. load tree ring data
3. match the tree core and inventory data for individual trees and merge these data sets into one data frame
4. format this data into a list for input into JAGS
5. run the JAGS model
6. visualize the output

For steps 2-4 we will leverage functions already written to deal with these steps that are part of the [PEcAn system](https://pecanproject.org). Specifically, they are within PEcAn’s land data R package, which can be downloaded and installed off Github using devtools

if(!require(PEcAn.data.land)){  
 library(devtools)  
 install.packages(c("digest","dplR","PeriodicTable"))  
 devtools::install\_github("PecanProject/pecan/base/logger")  
 devtools::install\_github("PecanProject/pecan/base/remote")  
 devtools::install\_github("PecanProject/pecan/base/utils")  
 devtools::install\_github("PecanProject/pecan/base/db")  
 devtools::install\_github("PecanProject/pecan/modules/data.land")  
 require(PEcAn.data.land)  
}

## Loading required package: PEcAn.data.land

## Loading required package: datapack

## Loading required package: dataone

## Warning: package 'dataone' was built under R version 3.5.2

## Loading required package: PEcAn.DB

## Loading required package: PEcAn.utils

##   
## Attaching package: 'PEcAn.utils'

## The following object is masked from 'package:utils':  
##   
## download.file

## Loading required package: redland

library(rjags)

## Loading required package: coda

## Linked to JAGS 4.3.0

## Loaded modules: basemod,bugs

library(ecoforecastR)

library(PEcAn.data.land)  
## 1. Read tree data  
trees <- read.csv("data/H2012AdultFieldData.csv")  
  
## 2. Read tree ring data  
rings <- PEcAn.data.land::Read\_Tucson("data/TUCSON/")

## There is 1 series  
## 1 S3C5369 1983 2012 0.001  
## There is 1 series  
## 1 S3C5375 1986 2012 0.001  
## There is 1 series  
## 1 S5A5271 1960 2012 0.001  
## There are 2 series  
## 1 B6A8534 1938 2012 0.001  
## 2 B6A8523 1936 2012 0.001  
## There are 2 series  
## 1 B6A8539 1888 2012 0.001  
## 2 V6B8568 1895 2012 0.001  
## There are 2 series  
## 1 B6B8567 1900 2012 0.001  
## 2 B6A8528 1892 2012 0.001  
## There are 12 series  
## 1 B3C6158 1901 2012 0.001  
## 2 B3C6065 1907 2012 0.001  
## 3 H1A1640 1963 2012 0.001  
## 4 B3D6164 1890 2012 0.001  
## 5 H6D1623 1958 2012 0.001  
## 6 H6B1545 1957 2012 0.001  
## 7 H1A1656 1996 2012 0.001  
## 8 S5D5297 1971 2012 0.001  
## 9 S5A5286 1976 2012 0.001  
## 10 B8C6001 1815 2012 0.001  
## 11 B3C6067 1925 2012 0.001  
## 12 B3C6086 1902 2012 0.001  
## There are 10 series  
## 1 B6A8526 1894 2012 0.001  
## 2 B6B8570 1878 2012 0.001  
## 3 H6A1507 1953 2012 0.001  
## 4 S3D5366 1964 2012 0.001  
## 5 B6A8537 1891 2012 0.001  
## 6 B6B8578 1883 2012 0.001  
## 7 H6B1542 1950 2012 0.001  
## 8 H1D1715 1959 2012 0.001  
## 9 H1B1693 1944 2012 0.001  
## 10 H6A1523 1965 2012 0.001  
## There are 2 series  
## 1 B6B8581 1936 2012 0.001  
## 2 B6B8580 1882 2012 0.001  
## There are 8 series  
## 1 B6B8584 1895 2012 0.001  
## 2 B6D8604 1910 2012 0.001  
## 3 B6A8530 1890 2012 0.001  
## 4 B6B8572 1914 2012 0.001  
## 5 B6A8524 1873 2012 0.001  
## 6 B6B8583 1902 2012 0.001  
## 7 B6A8536 1930 2012 0.001  
## 8 B6D8589 1885 2012 0.001  
## There are 2 series  
## 1 B6A8519 1893 2012 0.001  
## 2 B6D8590 1867 2012 0.001  
## There are 2 series  
## 1 B6D8593 1882 2012 0.001  
## 2 B6D8609 1883 2012 0.001  
## There are 2 series  
## 1 B6D8594 1892 2012 0.001  
## 2 B6D8606 1888 2012 0.001  
## There are 10 series  
## 1 B6D8595 1896 2012 0.001  
## 2 B6B8573 1948 2012 0.001  
## 3 B6D8603 1887 2012 0.001  
## 4 B6B8560 1894 2012 0.001  
## 5 H1B1701 1931 2012 0.001  
## 6 S3B5389 1978 2012 0.001  
## 7 B3D6171 1917 2012 0.001  
## 8 B3C6066 1912 2012 0.001  
## 9 B3B6194 1924 2012 0.001  
## 10 B3B6182 1967 2012 0.001  
## There are 10 series  
## 1 B6D8596 1923 2012 0.001  
## 2 B6D8605 1938 2012 0.001  
## 3 H1D1737 1957 2012 0.001  
## 4 H6A1527 1949 2012 0.001  
## 5 S3C5368 1953 2012 0.001  
## 6 H6A1514 1950 2012 0.001  
## 7 H1D1743 1961 2012 0.001  
## 8 H6A1501 1948 2012 0.001  
## 9 H1C1709 1960 2012 0.001  
## 10 H1D1754 1961 2012 0.001  
## There are 10 series  
## 1 B7D5736 1958 2012 0.001  
## 2 B7C5770 1961 2012 0.001  
## 3 B7C5762 1945 2012 0.001  
## 4 B7A5697 1930 2012 0.001  
## 5 B7B5831 1958 2012 0.001  
## 6 B7D5728 1890 2012 0.001  
## 7 B7A5699 1910 2012 0.001  
## 8 B7A5701 1934 2012 0.001  
## 9 B7A5709 1888 2012 0.001  
## 10 B7C5774 1899 2012 0.001  
## There are 9 series  
## 1 B6C8554 1875 2012 0.001  
## 2 B6C8552 1951 2012 0.001  
## 3 B8B5945 1930 2012 0.001  
## 4 B8B5894 1957 2012 0.001  
## 5 B6C8548 1884 2012 0.001  
## 6 B6C8544 1877 2012 0.001  
## 7 B3B6177 1910 2012 0.001  
## 8 B7D5739 1960 2012 0.001  
## 9 B7D5732 1964 2012 0.001  
## There are 10 series  
## 1 H1A1648 1959 2012 0.001  
## 2 H1B1703 1950 2012 0.001  
## 3 H1D1734 1961 2012 0.001  
## 4 H1D1761 1977 2012 0.001  
## 5 H6C1565 1985 2012 0.001  
## 6 H8B1559 1906 2012 0.001  
## 7 H6A1536 1953 2012 0.001  
## 8 H6A1532 1963 2012 0.001  
## 9 H6A1513 1931 2012 0.001  
## 10 H6C1577 1951 2012 0.001  
## There are 13 series  
## 1 H6A1525 1950 2012 0.001  
## 2 H1B1700 1917 2012 0.001  
## 3 S3B5386 1946 2012 0.001  
## 4 B6D8602 1891 2012 0.001  
## 5 S5A5289 1988 2012 0.001  
## 6 S5C5324 1955 2012 0.001  
## 7 H6A1528 1947 2012 0.001  
## 8 H1B1697 1927 2012 0.001  
## 9 H1B1689 1952 2012 0.001  
## 10 S3C5370 1968 2012 0.001  
## 11 S3C5376 1981 2012 0.001  
## 12 H6A1531 1962 2012 0.001  
## 13 H1A1636 1957 2012 0.001  
## There are 8 series  
## 1 H1C1721 1961 2012 0.001  
## 2 H6C1567 1947 2012 0.001  
## 3 H1B1704 1934 2012 0.001  
## 4 S3A5355 1921 2012 0.001  
## 5 H6D1620 1951 2012 0.001  
## 6 H6C1576 1952 2012 0.001  
## 7 H6B1557 1909 2012 0.001  
## 8 H6A1534 1952 2012 0.001  
## There are 14 series  
## 1 H6B1550 1969 2012 0.001  
## 2 H6A1535 1959 2012 0.001  
## 3 H8B0557 1968 2012 0.001  
## 4 H6D1622 1972 2012 0.001  
## 5 S6C5325 1987 2012 0.001  
## 6 S6A5273 1986 2012 0.001  
## 7 B6B8582 1921 2012 0.001  
## 8 B6D8597 1946 2012 0.001  
## 9 B6A8518 1943 2012 0.001  
## 10 B6B8569 1917 2012 0.001  
## 11 5516 1947 2012 0.001  
## 12 H1D1748 1964 2012 0.001  
## 13 5530 1968 2012 0.001  
## 14 H6A1526 1943 2012 0.001  
## There are 14 series  
## 1 B6D8610 1964 2012 0.001  
## 2 B6A8533 1927 2012 0.001  
## 3 5537 1941 2012 0.001  
## 4 5543 1947 2012 0.001  
## 5 B8C5998 1970 2012 0.001  
## 6 B3C6072 1956 2012 0.001  
## 7 H6C1570 1972 2012 0.001  
## 8 H6A1508 1948 2012 0.001  
## 9 B3A6029 1963 2012 0.001  
## 10 B3C6078 1958 2012 0.001  
## 11 B3C6C83 1908 2012 0.001  
## 12 B3B6181 1950 2012 0.001  
## 13 H6C1572 1952 2012 0.001  
## 14 H6D1619 1957 2012 0.001  
## There are 14 series  
## 1 H6A1502 1950 2012 0.001  
## 2 H6D1616 1971 2012 0.001  
## 3 S3A5356 1961 2012 0.001  
## 4 H6B1548 1967 2012 0.001  
## 5 H1A1644 1970 2012 0.001  
## 6 H1A1637 1975 2012 0.001  
## 7 H1D1751 1971 2012 0.001  
## 8 H1C1716 1969 2012 0.001  
## 9 B3D6162 1917 2012 0.001  
## 10 B3C6075 1948 2012 0.001  
## 11 5527 1969 2012 0.001  
## 12 5522 1962 2012 0.001  
## 13 H6A1517 1958 2012 0.001  
## 14 H6B1541 1953 2012 0.001  
## There are 14 series  
## 1 B6B8587 1987 2012 0.001  
## 2 B6B8579 1946 2012 0.001  
## 3 B6A8525 1897 2012 0.001  
## 4 B6A8529 1918 2012 0.001  
## 5 5544 1955 2012 0.001  
## 6 S5C5318 1980 2012 0.001  
## 7 S3A5357 1942 2012 0.001  
## 8 H6D1614 1952 2012 0.001  
## 9 S5A5285 1961 2012 0.001  
## 10 5519 1940 2012 0.001  
## 11 B6A8535 1948 2012 0.001  
## 12 B6A8522 1917 2012 0.001  
## 13 B6D8588 1907 2012 0.001  
## 14 B6D8608 1901 2012 0.001  
## There are 10 series  
## 1 5055 1950 2012 0.001  
## 2 5049 1920 2012 0.001  
## 3 B8A5858 1972 2012 0.001  
## 4 B8A5868 1979 2012 0.001  
## 5 5040 1954 2012 0.001  
## 6 5600 1950 2012 0.001  
## 7 5065 1951 2012 0.001  
## 8 5592 1901 2012 0.001  
## 9 B8A5850 1960 2012 0.001  
## 10 B8A5848 1939 2012 0.001  
## There are 2 series  
## 1 S5D5308 1987 2012 0.001  
## 2 S5C5316 1972 2012 0.001  
## There are 2 series  
## 1 5535 1882 2012 0.001  
## 2 S5D5298 1979 2012 0.001  
## There are 2 series  
## 1 S5A5291 1977 2012 0.001  
## 2 S5C5322 1970 2012 0.001  
## There are 2 series  
## 1 5542 1952 2012 0.001  
## 2 S5A5278 1984 2012 0.001  
## There are 2 series  
## 1 S5A5283 1964 2012 0.001  
## 2 S5D5309 1950 2012 0.001  
## There are 2 series  
## 1 H1C1715 1928 2012 0.001  
## 2 H1B1688 1917 2012 0.001  
## There are 2 series  
## 1 H1B1693 1944 2012 0.001  
## 2 H1A1647 1958 2012 0.001  
## There are 2 series  
## 1 H1C1710 1918 2012 0.001  
## 2 H6B1540 1909 2012 0.001  
## There are 2 series  
## 1 H1D1754 1961 2012 0.001  
## 2 H6C1566 1945 2012 0.001  
## There are 2 series  
## 1 H1B1705 1924 2012 0.001  
## 2 H1D1739 1949 2012 0.001  
## There are 2 series  
## 1 H1D1749 1957 2012 0.001  
## 2 H6A1506 1967 2012 0.001  
## There are 2 series  
## 1 H6B1552 1948 2012 0.001  
## 2 H6A1505 1953 2012 0.001  
## There are 2 series  
## 1 H6B1555 1944 2012 0.001  
## 2 H1D1750 1941 2012 0.001  
## There are 2 series  
## 1 H1D1760 1956 2012 0.001  
## 2 H6C1575 1959 2012 0.001  
## There are 2 series  
## 1 1698 1930 2012 0.001  
## 2 1736 1971 2012 0.001  
## There are 10 series  
## 1 5058 1947 2012 0.001  
## 2 5594 1910 2011 0.001  
## 3 B8A5852 1972 2012 0.001  
## 4 B8A5862 1959 2012 0.001  
## 5 5048 1947 2012 0.001  
## 6 5056 1955 2012 0.001  
## 7 5054 1960 2012 0.001  
## 8 5053 1941 2012 0.001  
## 9 B8A5844 1887 2012 0.001  
## 10 B8A5867 1947 2012 0.001  
## There are 8 series  
## 1 H7C5433 1974 2012 0.001  
## 2 H7C5451 1956 2012 0.001  
## 3 H7C5452 1950 2012 0.001  
## 4 H7C5458 1975 2012 0.001  
## 5 H7C5438 1958 2012 0.001  
## 6 H7C5455 1939 2012 0.001  
## 7 B7B5828 1962 2012 0.001  
## 8 B7D5738 1971 2012 0.001  
## There are 4 series  
## 1 B8A5858 1974 2012 0.001  
## 2 B8A5868 1980 2012 0.001  
## 3 B8A5848 1939 2012 0.001  
## 4 B8A5850 1948 2012 0.001  
## There are 6 series  
## 1 5065 1948 2012 0.001  
## 2 5592 1907 2012 0.001  
## 3 5040 1960 2012 0.001  
## 4 5600 1950 2012 0.001  
## 5 5055 1944 2012 0.001  
## 6 5049 1924 2012 0.001  
## There is 1 series  
## 1 B3D6166 1899 2012 0.001

## 3. merge inventory and tree ring data, extract most recent nyears  
combined <- PEcAn.data.land::matchInventoryRings(trees,rings,nyears=15)  
  
## take a look at the first few rows of data to see the structure  
knitr::kable(combined[1:5,])

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | SITE | PLOT | SUB | TAG | TREEID | NUM | SPP | X | Y | DATE11 | DBH11 | DBH12 | DATE\_CORE\_COLLECTED11 | DATE12 | DATE\_CORE\_COLLECT12 | 1998 | 1999 | 2000 | 2001 | 2002 | 2003 | 2004 | 2005 | 2006 | 2007 | 2008 | 2009 | 2010 | 2011 | 2012 |
| H6A1501 | H | 6 | A | 1501 | H6A1501 | 1 | PRSE2 | 1.2 | 0.6 | 7/10/11 | 24.8 | 24.75 | 7/10/11 | 7/2/12 | 7/2/12 | 2.433 | 1.357 | 1.300 | 1.301 | 1.108 | 1.303 | 0.870 | 0.951 | 1.319 | 1.431 | 1.253 | 1.381 | 1.107 | 0.686 | 0.624 |
| H6A1502 | H | 6 | A | 1502 | H6A1502 | 1 | ACRU | 0.2 | 2.0 | 7/10/11 | 24.2 | 24.5 | 7/10/11 | 7/2/12 | 7/2/12 | 1.227 | 0.581 | 1.066 | 0.950 | 1.387 | 0.917 | 1.084 | 1.824 | 1.513 | 2.099 | 2.066 | 1.603 | 0.744 | 2.104 | 1.940 |
| H6A1505 | H | 6 | A | 1505 | H6A1505 | 1 | ACRU | 3.9 | 3.4 | 7/10/11 | 16.75 | 16.75 | 7/10/11 | 7/2/12 | 7/2/12 | 0.939 | 0.750 | 0.867 | 0.656 | 0.457 | 0.822 | 0.529 | 0.698 | 0.381 | 0.465 | 0.233 | 0.433 | 0.275 | 0.180 | 0.233 |
| H6A1506 | H | 6 | A | 1506 | H6A1506 | 1 | BEAL2 | 5.7 | 3.2 | 7/10/11 | 17.2 | 17.2 | 7/10/11 | 7/2/12 | 7/2/12 | 0.979 | 0.595 | 0.561 | 0.958 | 0.821 | 0.765 | 0.804 | 0.635 | 1.143 | 0.485 | 0.825 | 0.439 | 1.028 | 0.361 | 0.727 |
| H6A1507 | H | 6 | A | 1507 | H6A1507 | 1 | TSCA | 8.2 | 5.5 | 7/10/11 | 30.1 | 30.4 | 7/10/11 | 7/2/12 | 7/2/12 | 0.624 | 0.536 | 0.444 | 0.408 | 0.272 | 0.720 | 0.519 | 1.842 | 1.701 | 1.749 | 1.344 | 1.373 | 1.277 | 1.106 | 0.696 |

## 4. organize data into a list  
data <-PEcAn.data.land::buildJAGSdata\_InventoryRings(combined)

## Warning in `[.data.frame`(combined, , !  
## is.na(as.numeric(colnames(combined)))): NAs introduced by coercion

# y = increment (tree x year)  
# z = dbh (tree x year)  
# make sure to take a look at all the priors!  
str(data)

## List of 13  
## $ y : num [1:80, 1:15] 0.487 0.245 0.188 0.196 0.125 ...  
## ..- attr(\*, "dimnames")=List of 2  
## .. ..$ : chr [1:80] "H6A1501" "H6A1502" "H6A1505" "H6A1506" ...  
## .. ..$ : chr [1:15] "1998" "1999" "2000" "2001" ...  
## $ z : num [1:80, 1:15] NA NA NA NA NA NA NA NA NA NA ...  
## $ ni : int 80  
## $ nt : int 15  
## $ x\_ic : num 1  
## $ tau\_ic: num 1e-04  
## $ a\_dbh : num 16  
## $ r\_dbh : num 8  
## $ a\_inc : num 0.001  
## $ r\_inc : num 1  
## $ a\_add : num 1  
## $ r\_add : num 1  
## $ time : num [1:15] 1998 1999 2000 2001 2002 ...

Now that we have the data prepped we need to fit the model itself. The bulk of this code is just the same JAGS syntax we’ve used before, so lets focus on the JAGS code itself. To begin with, lets look back at the JAGS code for the random walk

model{  
   
 #### Data Model  
 for(i in 1:n){  
 y[i] ~ dnorm(x[i],tau\_obs)  
 }  
   
 #### Process Model  
 for(i in 2:n){  
 x[i]~dnorm(x[i-1],tau\_add)  
 }  
   
 #### Priors  
 x[1] ~ dnorm(x\_ic,tau\_ic)  
 tau\_obs ~ dgamma(a\_obs,r\_obs)  
 tau\_add ~ dgamma(a\_add,r\_add)  
}

Since we’re fusing two data sources, we’ll need to add a second data model. We’ll also modify our process model to include a mean growth rate term. Finally, we’ll need to specify priors on both observation errors, the process error, and the mean.

model{  
  
 #### Data Model: DBH  
 for(i in 1:n){  
 z[i] ~ dnorm(x[i],tau\_dbh)  
 }  
  
 #### Data Model: growth  
 for(i in 2:n){  
 inc[i] <- x[i]-x[i-1]  
 y[i] ~ dnorm(inc[i],tau\_inc)  
 }  
  
 #### Process Model  
 #### Dnew is the expected new diameter given the previous diameter, x[i-1], and the mean growth rate, mu  
 for(i in 2:n){  
 Dnew[i] <- x[i-1] + mu   
 x[i]~dnorm(Dnew[i],tau\_add)  
 }  
  
 #### Priors  
 x[1] ~ dnorm(x\_ic,tau\_ic) ## initial DBH  
 tau\_dbh ~ dgamma(a\_dbh,r\_dbh) ## observation error: DBH  
 tau\_inc ~ dgamma(a\_inc,r\_inc) ## observation error: tree rings  
 tau\_add ~ dgamma(a\_add,r\_add) ## process error: growth  
 mu ~ dnorm(0.5,0.5) ## mean growth  
}

This code would work perfectly if we only had only measured a single tree, but we measured a number of trees so next need to modify the code to work with tree-by-year matrices of DBH and growth.

model{  
  
 ### Loop over all individuals  
 for(i in 1:ni){  
   
 #### Data Model: DBH  
 for(t in 1:nt){  
 z[i,t] ~ dnorm(x[i,t],tau\_dbh)  
 }  
   
 #### Data Model: growth  
 for(t in 2:nt){  
 inc[i,t] <- x[i,t]-x[i,t-1]  
 y[i,t] ~ dnorm(inc[i,t],tau\_inc)  
 }  
   
 #### Process Model  
 for(t in 2:nt){  
 Dnew[i,t] <- x[i,t-1] + mu  
 x[i,t]~dnorm(Dnew[i,t],tau\_add)  
 }  
   
 x[i,1] ~ dnorm(x\_ic,tau\_ic)  
 } ## end loop over individuals  
   
 #### Priors  
 tau\_dbh ~ dgamma(a\_dbh,r\_dbh)  
 tau\_inc ~ dgamma(a\_inc,r\_inc)  
 tau\_add ~ dgamma(a\_add,r\_add)  
 mu ~ dnorm(0.5,0.5)  
}

Finally, since growth is indexed by both tree and year, lets add random effects for both individuals and years. In this case our process model now becomes

Dnew[i,t] <- x[i,t-1] + mu + ind[i] + year[t]

where ind and year are the random effects for individual and year respectively. Next, we’ll need to specify the distributions that these random effects are drawn from, as well as the priors on the random effect variances

model{  
   
### Loop over all individuals  
for(i in 1:ni){  
   
 #### Data Model: DBH  
 for(t in 1:nt){  
 z[i,t] ~ dnorm(x[i,t],tau\_dbh)  
 }  
   
 #### Data Model: growth  
 for(t in 2:nt){  
 inc[i,t] <- x[i,t]-x[i,t-1]  
 y[i,t] ~ dnorm(inc[i,t],tau\_inc)  
 }  
   
 #### Process Model  
 for(t in 2:nt){  
 Dnew[i,t] <- x[i,t-1] + mu + ind[i] + year[t]  
 x[i,t]~dnorm(Dnew[i,t],tau\_add)  
 }  
   
 ## individual effects  
 ind[i] ~ dnorm(0,tau\_ind)  
   
 ## initial condition  
 x[i,1] ~ dnorm(x\_ic,tau\_ic)  
   
} ## end loop over individuals  
   
 ## year effects  
 for(t in 1:nt){  
 year[t] ~ dnorm(0,tau\_yr)  
 }  
   
   
 #### Priors  
 tau\_dbh ~ dgamma(a\_dbh,r\_dbh)  
 tau\_inc ~ dgamma(a\_inc,r\_inc)  
 tau\_add ~ dgamma(a\_add,r\_add)  
 tau\_ind ~ dgamma(1,0.1)  
 tau\_yr ~ dgamma(1,0.1)  
 mu ~ dnorm(0.5,0.5)  
   
 }

Putting this all together gives the following R code for the base case (no random effects)

n.iter = 50000 ## INCREASE THIS NUMBER FOR ACTUAL ANALYSES \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
## this code fuses forest inventory data with tree growth data (tree ring or dendrometer band)  
## for the same plots. Code is a rewrite of Clark et al 2007 Ecol Appl into JAGS  
TreeDataFusionMV = "  
model{  
  
### Loop over all individuals  
for(i in 1:ni){  
   
 #### Data Model: DBH  
 for(t in 1:nt){  
 z[i,t] ~ dnorm(x[i,t],tau\_dbh)  
 }  
   
 #### Data Model: growth  
 for(t in 2:nt){  
 inc[i,t] <- x[i,t]-x[i,t-1]  
 y[i,t] ~ dnorm(inc[i,t],tau\_inc)  
 }  
   
 #### Process Model  
 for(t in 2:nt){  
 Dnew[i,t] <- x[i,t-1] + mu  
 x[i,t]~dnorm(Dnew[i,t],tau\_add)  
 }  
   
 x[i,1] ~ dnorm(x\_ic,tau\_ic)  
  
} ## end loop over individuals  
   
 #### Priors  
 tau\_dbh ~ dgamma(a\_dbh,r\_dbh)  
 tau\_inc ~ dgamma(a\_inc,r\_inc)  
 tau\_add ~ dgamma(a\_add,r\_add)  
 mu ~ dnorm(0.5,0.5)  
}"  
  
 ## state variable initial condition (subtract observed diameter increments off from the observed diameter)  
 z0 = t(apply(data$y,1,function(y){-rev(cumsum(rev(y)))})) + data$z[,ncol(data$z)]   
   
 ## JAGS initial conditions  
 nchain = 3  
 init <- list()  
 for(i in 1:nchain){  
 y.samp = sample(data$y,length(data$y),replace=TRUE)  
 init[[i]] <- list(x = z0,tau\_add=runif(1,1,5)/var(diff(y.samp),na.rm=TRUE),  
 tau\_dbh=1,tau\_inc=500,tau\_ind=50,tau\_yr=100,ind=rep(0,data$ni),year=rep(0,data$nt))  
 }  
   
 ## compile JAGS model  
 j.model <- jags.model (file = textConnection(TreeDataFusionMV),  
 data = data,  
 inits = init,  
 n.chains = 3)

## Warning in jags.model(file = textConnection(TreeDataFusionMV), data =  
## data, : Unused variable "time" in data

## Compiling model graph  
## Resolving undeclared variables  
## Allocating nodes  
## Graph information:  
## Observed stochastic nodes: 1274  
## Unobserved stochastic nodes: 2250  
## Total graph size: 5855

## Warning in jags.model(file = textConnection(TreeDataFusionMV), data =  
## data, : Unused initial value for "tau\_ind" in chain 1

## Warning in jags.model(file = textConnection(TreeDataFusionMV), data =  
## data, : Unused initial value for "tau\_yr" in chain 1

## Warning in jags.model(file = textConnection(TreeDataFusionMV), data =  
## data, : Unused initial value for "ind" in chain 1

## Warning in jags.model(file = textConnection(TreeDataFusionMV), data =  
## data, : Unused initial value for "year" in chain 1

## Warning in jags.model(file = textConnection(TreeDataFusionMV), data =  
## data, : Unused initial value for "tau\_ind" in chain 2

## Warning in jags.model(file = textConnection(TreeDataFusionMV), data =  
## data, : Unused initial value for "tau\_yr" in chain 2

## Warning in jags.model(file = textConnection(TreeDataFusionMV), data =  
## data, : Unused initial value for "ind" in chain 2

## Warning in jags.model(file = textConnection(TreeDataFusionMV), data =  
## data, : Unused initial value for "year" in chain 2

## Warning in jags.model(file = textConnection(TreeDataFusionMV), data =  
## data, : Unused initial value for "tau\_ind" in chain 3

## Warning in jags.model(file = textConnection(TreeDataFusionMV), data =  
## data, : Unused initial value for "tau\_yr" in chain 3

## Warning in jags.model(file = textConnection(TreeDataFusionMV), data =  
## data, : Unused initial value for "ind" in chain 3

## Warning in jags.model(file = textConnection(TreeDataFusionMV), data =  
## data, : Unused initial value for "year" in chain 3

## Initializing model

## burn-in  
 jags.out <- coda.samples (model = j.model,  
 variable.names = c("tau\_add","tau\_dbh","tau\_inc","mu","tau\_ind","tau\_yr"),  
 n.iter = min(n.iter,20000))

## Warning in FUN(X[[i]], ...): Failed to set trace monitor for tau\_ind  
## Variable tau\_ind not found

## Warning in FUN(X[[i]], ...): Failed to set trace monitor for tau\_yr  
## Variable tau\_yr not found

#plot(jags.out)  
   
 ## run MCMC  
 #jags.out <- coda.samples (model = j.model,  
 # variable.names = c("x","tau\_add","tau\_dbh","tau\_inc","mu",  
 # "tau\_ind","tau\_yr","ind","year"),  
 # n.iter = n.iter)

Next, lets generate some diagnostic plots to look at the model. First, lets plot the posterior CI for growth and DBH and compare these to observations. Since we have scores of cores and trees, we’ll pick a random subset of trees to check. One thing that’s critical to note is that for the confidence intervals on growth that these are calculated pathwise – we’re looking at the growth from a whole MCMC iteration – rather than pairwise (i.e. subtracting the posterior distribution for DBH at one point from the posterior distribution of DBH at the next). Because there’s high correlations between successive time points, the pathwise uncertainty estimates are considerably lower in uncertainty – essentially saying that we know can know the growth rate of the tree better than we can know the actual size of the tree

#### Diagnostic plots  
   
 ### DBH  
 #layout(matrix(1:8,4,2))  
 #out <- as.matrix(jags.out)  
   
# x.cols = which(substr(colnames(out),1,1)=="x") ## which columns are the state variable, x  
 # ci <- apply(out[,x.cols],2,quantile,c(0.025,0.5,0.975))  
 # ci.names = PEcAn.data.land::parse.MatrixNames(colnames(ci),numeric=TRUE)  
   
 # smp = c(sample.int(data$ni,3),49) ## I've rigged the sampling to make sure you see tree 49!  
# for(i in smp){  
# sel = which(ci.names$row == i)  
# plot(data$time,ci[2,sel],type='n',ylim=range(ci[,sel],na.rm=TRUE),ylab="DBH (cm)",main=i)  
# ciEnvelope(data$time,ci[1,sel],ci[3,sel],col="lightBlue")  
# points(data$time,data$z[i,],pch="+",cex=1.5)  
# }  
   
 ## growth  
 #for(i in smp){  
 # sel = which(ci.names$row == i)  
 # inc.mcmc = apply(out[,x.cols[sel]],1,diff)  
 # inc.ci = apply(inc.mcmc,1,quantile,c(0.025,0.5,0.975))\*5  
 #   
 # plot(data$time[-1],inc.ci[2,],type='n',ylim=range(inc.ci,na.rm=TRUE),ylab="Ring Increment (mm)")  
 # ciEnvelope(data$time[-1],inc.ci[1,],inc.ci[3,],col="lightBlue")  
 # points(data$time,data$y[i,]\*5,pch="+",cex=1.5,type='b',lty=2)  
 #}

Second, let’s look at the histogram of our fixed effect, mu, and the precisions. Let’s also convert the precisions to standard deviations to make them easier to interpret

Note: I had to comment these sections for knitting. I have yet to debug why.

## process model  
 # vars = (1:ncol(out))[-c(which(substr(colnames(out),1,1)=="x"),grep("tau",colnames(out)),  
 # grep("year",colnames(out)),grep("ind",colnames(out)))]  
 # par(mfrow=c(1,1))  
 # for(i in vars){  
 # hist(out[,i],main=colnames(out)[i])  
 # }  
 # if(length(vars)>1) pairs(out[,vars])  
 #   
 # ## Standard Deviations  
 # par(mfrow=c(2,3))  
 # prec = out[,grep("tau",colnames(out))]  
 # for(i in 1:ncol(prec)){  
 # hist(1/sqrt(prec[,i]),main=colnames(prec)[i])  
 # }  
 # cor(prec)  
 # pairs(prec)

Third, let’s look at the random effects. It is easy enough to plot the year effects by year. For the individual effects we’ll plot these twice, first ordering the effects by plot and the second ordering them by species.

out <- as.matrix(jags.out)  
 par(mfrow=c(1,1))  
 ### YEAR  
 year.cols = grep("year",colnames(out))  
 if(length(year.cols>0)){  
 ci.yr <- apply(out[,year.cols],2,quantile,c(0.025,0.5,0.975))  
 plot(data$time,ci.yr[2,],type='n',ylim=range(ci.yr,na.rm=TRUE),main="Year Effect",ylab="cm")  
 ciEnvelope(data$time,ci.yr[1,],ci.yr[3,],col="lightBlue")  
 lines(data$time,ci.yr[2,],lty=1,lwd=2)  
 abline(h=0,lty=2)  
 }  
   
 ### INDIV  
 ind.cols= which(substr(colnames(out),1,3)=="ind")  
 if(length(ind.cols)>0){  
 boxplot(out[,ind.cols],horizontal=TRUE,outline=FALSE,col=combined$PLOT,main="Individual Effects By Plot",xlab="cm")  
 abline(v=0,lty=2)  
 ## calculate plot-level means for random effects  
 tapply(apply(out[,ind.cols],2,mean),combined$PLOT,mean)  
 table(combined$PLOT)  
   
 spp = combined$SPP  
 boxplot(out[order(spp),ind.cols],horizontal=TRUE,outline=FALSE,col=spp[order(spp)],main="Individual Effects By Species",xlab="cm")  
 abline(v=0,lty=2)  
 spp.code = levels(spp)[table(spp)>0]  
 legend("bottomright",legend=rev(spp.code),col=rev(which(table(spp)>0)),lwd=4)  
 ## calculate species-level means for random effects  
 tapply(apply(out[,ind.cols],2,mean),combined$SPP,mean)  
 }

By default this code is set to run with a small number of years (15), and a much too low number of MCMC iterations (500), just so that the code with “knit” quickly initially. For your analyses you should obviously increase these – I found that convergence was adequate with around 20,000 samples, though I probably would run 10x longer than that for a publishable analysis. However, such an analysis would take hours to run.

Assignment:

1. Run the model initially with random effects off
2. Rerun the model with random effects on. Compare this to the previous run. What is the relative partitioning of uncertainties in the different versions of the model among observation error, process error, and the different random effects? What does the size of these effects suggest about the drivers of uncertainty in tree growth?

## Random Effects

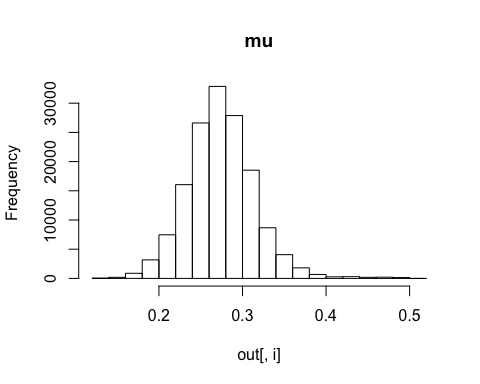
n.iter = 50000 ## INCREASE THIS NUMBER FOR ACTUAL ANALYSES \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
## this code fuses forest inventory data with tree growth data (tree ring or dendrometer band)  
## for the same plots. Code is a rewrite of Clark et al 2007 Ecol Appl into JAGS  
TreeDataFusionMV\_random = "  
model{  
   
### Loop over all individuals  
for(i in 1:ni){  
   
 #### Data Model: DBH  
 for(t in 1:nt){  
 z[i,t] ~ dnorm(x[i,t],tau\_dbh)  
 }  
   
 #### Data Model: growth  
 for(t in 2:nt){  
 inc[i,t] <- x[i,t]-x[i,t-1]  
 y[i,t] ~ dnorm(inc[i,t],tau\_inc)  
 }  
   
 #### Process Model  
 for(t in 2:nt){  
 Dnew[i,t] <- x[i,t-1] + mu + ind[i] + year[t]  
 x[i,t]~dnorm(Dnew[i,t],tau\_add)  
 }  
   
 ## individual effects  
 ind[i] ~ dnorm(0,tau\_ind)  
   
 ## initial condition  
 x[i,1] ~ dnorm(x\_ic,tau\_ic)  
   
} ## end loop over individuals  
   
 ## year effects  
 for(t in 1:nt){  
 year[t] ~ dnorm(0,tau\_yr)  
 }  
   
   
 #### Priors  
 tau\_dbh ~ dgamma(a\_dbh,r\_dbh)  
 tau\_inc ~ dgamma(a\_inc,r\_inc)  
 tau\_add ~ dgamma(a\_add,r\_add)  
 tau\_ind ~ dgamma(1,0.1)  
 tau\_yr ~ dgamma(1,0.1)  
 mu ~ dnorm(0.5,0.5)  
   
 }  
"  
  
  
 ## state variable initial condition (subtract observed diameter increments off from the observed diameter)  
 z0 = t(apply(data$y,1,function(y){-rev(cumsum(rev(y)))})) + data$z[,ncol(data$z)]   
   
 ## JAGS initial conditions  
 nchain = 3  
 init <- list()  
 for(i in 1:nchain){  
 y.samp = sample(data$y,length(data$y),replace=TRUE)  
 init[[i]] <- list(x = z0,tau\_add=runif(1,1,5)/var(diff(y.samp),na.rm=TRUE),  
 tau\_dbh=1,tau\_inc=500,tau\_ind=50,tau\_yr=100,ind=rep(0,data$ni),year=rep(0,data$nt))  
 }  
   
 ## compile JAGS model  
 j.model\_random <- jags.model (file = textConnection(TreeDataFusionMV\_random),  
 data = data,  
 inits = init,  
 n.chains = 3)

## Warning in jags.model(file = textConnection(TreeDataFusionMV\_random), data  
## = data, : Unused variable "time" in data

## Compiling model graph  
## Resolving undeclared variables  
## Allocating nodes  
## Graph information:  
## Observed stochastic nodes: 1274  
## Unobserved stochastic nodes: 2347  
## Total graph size: 5955  
##   
## Initializing model

jags.out <- coda.samples (model = j.model\_random,  
 variable.names = c("x","tau\_add","tau\_dbh","tau\_inc","mu",  
 "tau\_ind","tau\_yr","ind","year"),  
 n.iter = n.iter)  
  
out <- as.matrix(jags.out)

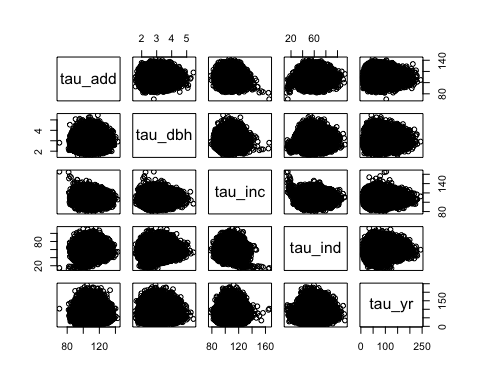
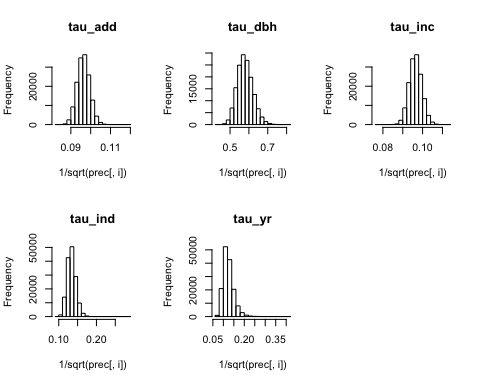
## process model  
 vars = (1:ncol(out))[-c(which(substr(colnames(out),1,1)=="x"),grep("tau",colnames(out)),  
 grep("year",colnames(out)),grep("ind",colnames(out)))]  
 par(mfrow=c(1,1))  
 for(i in vars){  
 hist(out[,i],main=colnames(out)[i])  
 }



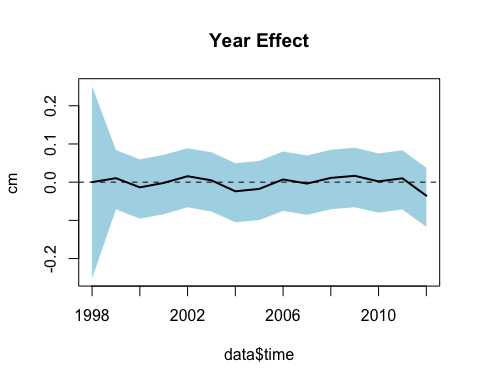
if(length(vars)>1) pairs(out[,vars])  
  
 ## Standard Deviations  
 par(mfrow=c(2,3))  
 prec = out[,grep("tau",colnames(out))]  
 for(i in 1:ncol(prec)){  
 hist(1/sqrt(prec[,i]),main=colnames(prec)[i])  
 }  
 cor(prec)

## tau\_add tau\_dbh tau\_inc tau\_ind tau\_yr  
## tau\_add 1.000000000 0.002271056 -0.090229419 -0.006298776 0.002564358  
## tau\_dbh 0.002271056 1.000000000 0.004209454 -0.004826593 0.013645218  
## tau\_inc -0.090229419 0.004209454 1.000000000 -0.015691974 0.003648713  
## tau\_ind -0.006298776 -0.004826593 -0.015691974 1.000000000 -0.003547307  
## tau\_yr 0.002564358 0.013645218 0.003648713 -0.003547307 1.000000000

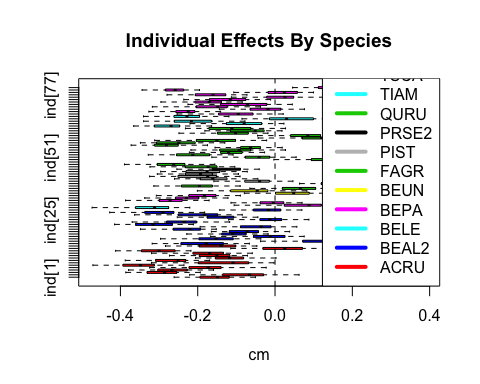
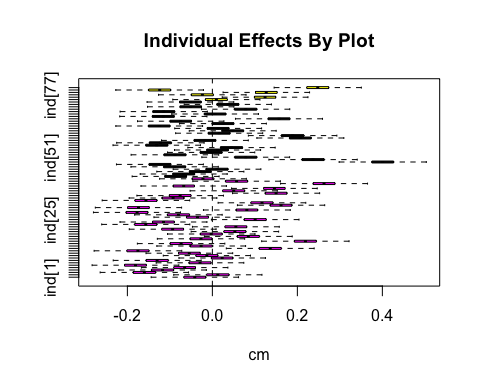
pairs(prec)



par(mfrow=c(1,1))  
 ### YEAR  
 year.cols = grep("year",colnames(out))  
 if(length(year.cols>0)){  
 ci.yr <- apply(out[,year.cols],2,quantile,c(0.025,0.5,0.975))  
 plot(data$time,ci.yr[2,],type='n',ylim=range(ci.yr,na.rm=TRUE),main="Year Effect",ylab="cm")  
 ciEnvelope(data$time,ci.yr[1,],ci.yr[3,],col="lightBlue")  
 lines(data$time,ci.yr[2,],lty=1,lwd=2)  
 abline(h=0,lty=2)  
 }



### INDIV  
 ind.cols= which(substr(colnames(out),1,3)=="ind")  
 if(length(ind.cols)>0){  
 boxplot(out[,ind.cols],horizontal=TRUE,outline=FALSE,col=combined$PLOT,main="Individual Effects By Plot",xlab="cm")  
 abline(v=0,lty=2)  
 ## calculate plot-level means for random effects  
 tapply(apply(out[,ind.cols],2,mean),combined$PLOT,mean)  
 table(combined$PLOT)  
   
 spp = combined$SPP  
 boxplot(out[order(spp),ind.cols],horizontal=TRUE,outline=FALSE,col=spp[order(spp)],main="Individual Effects By Species",xlab="cm")  
 abline(v=0,lty=2)  
 spp.code = levels(spp)[table(spp)>0]  
 legend("bottomright",legend=rev(spp.code),col=rev(which(table(spp)>0)),lwd=4)  
 ## calculate species-level means for random effects  
 tapply(apply(out[,ind.cols],2,mean),combined$SPP,mean)  
 }



## ACRU ACSA3 BEAL2 BELE BEPA   
## NA -0.10265789 NA -0.02205336 -0.08654511 0.01847687   
## BEUN CADE CATO CAUN FAGR FRAM   
## 0.14242372 NA NA NA 0.01802882 NA   
## HAVI PIRE PIRU PIST PRSE2 QUAL   
## NA NA NA 0.13259737 -0.04136590 NA   
## QURU QUVE TIAM TSCA   
## 0.02356415 NA -0.02527159 0.04365854

1. Based on the diagnostics, propose an additional effect (fixed or random) to add to the model. Such an effect should plausibly chip away at a sizable fraction of the unexplained variability – you wouldn’t want to propose an effect that isn’t associated with systematic variability.

Looking at the effects of year, species, and plot, it seemed to mee that individual trees from both plot and species were growing a lot or “pulling away from the pack”. I woud want to know what was causing those individuals to do so well (Maybe soil, maybe position within the comminity (Competition?, a lack of competition? )). Because there seems to be an indivudual level effect, I would make it a fixed effect.

1. Explain any additional exploratory analyses you would perform (e.g. plotting your proposed covariate against one of the random effects).

I would try to look at distance form edge, soil type, and some metric of competition or lack there of. I would probably plot it againts both plot and species, becuase I think that the individual-level effects show up in both. I would ask myself ’Do these covariates explain the varibility I see better than plot did or species did?".

1. Write the JAGS code that would fit the proposed model (note: you don’t have to run this model, just propose the code)

TreeDataFusionMV\_tess = " model{

### Loop over all individuals

for(i in 1:ni){

#### Data Model: DBH for(t in 1:nt){ z[i,t] ~ dnorm(x[i,t],tau\_dbh) }

#### Data Model: growth for(t in 2:nt){ inc[i,t] <- x[i,t]-x[i,t-1] y[i,t] ~ dnorm(inc[i,t],tau\_inc) }

#### Process Model for(t in 2:nt){ Dnew[i,t] <- x[i,t-1] + mu + distance\_from\_edge[i] + year[t] # Keeping year effect. Have yet to explain it. x[i,t]~dnorm(Dnew[i,t],tau\_add) }

## individual effects by distance from edge distance\_from\_edge[i] ~ dnorm(dist\_data[i],tau\_dist) # Still an effect with varibility, but informed by data

## initial condition x[i,1] ~ dnorm(x\_ic,tau\_ic)

} ## end loop over individuals

## year effects for(t in 1:nt){ year[t] ~ dnorm(0,tau\_yr) }

#### Priors tau\_dbh ~ dgamma(a\_dbh,r\_dbh) tau\_inc ~ dgamma(a\_inc,r\_inc) tau\_add ~ dgamma(a\_add,r\_add) tau\_dist ~ dgamma(1,0.1) ## Would make better priors probably tau\_yr ~ dgamma(1,0.1) mu ~ dnorm(0.5,0.5)

} "

\*\* BECAUSE THE PRODUCTION VERSION OF THIS CODE TAKES A LONG TIME TO RUN, PLEASE SUBMIT THE KNIT HTML NOT THE Rmd \*\*