Multilevel Urban Tree Allometric equations

Tedward Erker

February 7, 2018

Contents

1 Objective

To create better allometric equations for urban trees

2 Caveat

I am not an expert on this, but I think I have a contribution to make. I'm using a dataset I did not work to collect and it may have idiosyncrasies that I may not understand. I also don't fully understand the modelling approach used to create the equations, namely the weighting.

3 Why needed

The urban tree allometry dataset is an incredibly valuable resource for making better predictions about tree growth in urban environments, and it is essential for accurate ecosystem service evaluation. However, there are a number of limitations with the current set of equations that multilevel modelling can address.

Limitations:

1. Limited number of species in each region
2. Hard boundaries of regions

Solution: Allow for information to be shared across species and across regions when fitting models.

Improvements:

1. Climate based rather than regions based. Continuous rather than discrete, gradient rather than hard boundaries.
2.

1. One equation for 10 species in each region. What about species not on the list? What if I'm interested in the equation for red maple in the southeast, but the only equation comes from the northeast region? How should I adjust the equation?
2. What if my location is on the border of two regions? How should I average the equations from each region, espcially if they are of two different forms (e.g. cubic and log-log)?

1. Currently urban tree allometric equations are built separately for each species and for each region. So there is a separate equation for red maple in the northeast and a separate equation for red maple in the midwest. These are hard differences when such hard divisions don't exist. Better models for red maple could probably be made if the different regions could be pooled to the extent that their climates are similar.
2.

For practicioners who wish to use

what was the metric used to select from models? was it r^2? they used AIC shouldn't we select the model form based on expert knowledge about universal tree growth patterns, rather than a small sample of observations? We'd expect to occasionally find with small samples that a cubic relationship best fits the data. But this relationship suffers from lack of basis in what we know about how trees grow and may give grossly inaccurate predictions if considered outside the range of the data.

4 Exploring dbh and crown diameter

show plots of dbh versus biomass and the published equations on a figure.

look specifically for places where pooling might help (same species in two regions)

4.1 load libraries

4.2 read in data

```
d <- read_csv("data/RDS-2016-0005/Data/TS3_Raw_tree_data.csv")

#str(d)
```

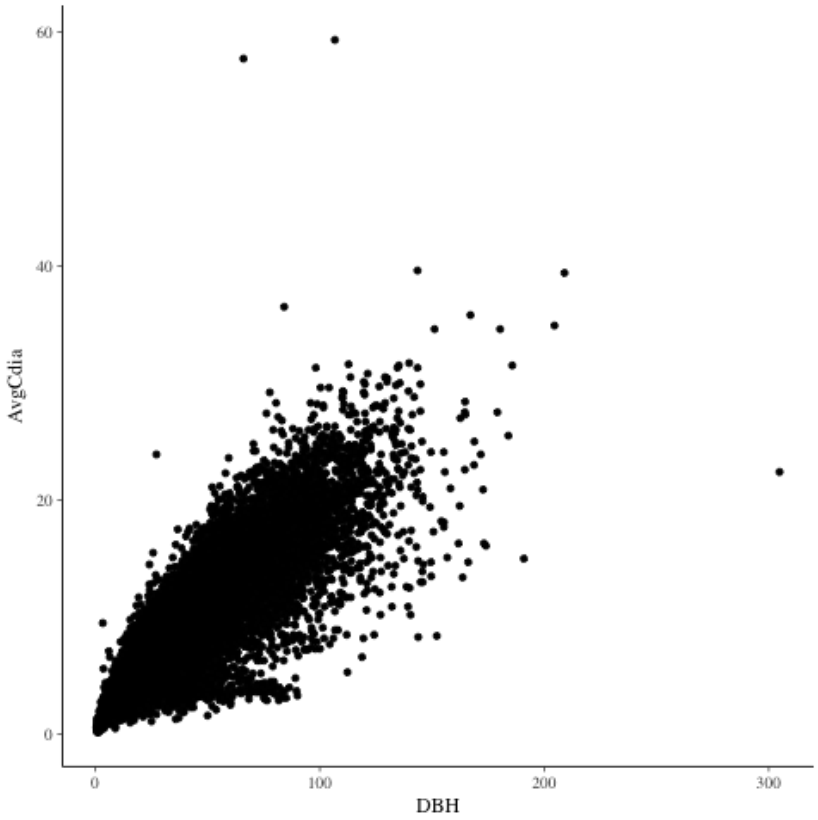
4.3 tidy a few names and select variables of interest here

```
d <- d %>% rename(DBH = `DBH (cm)`, Leaf = `Leaf (m2)`, AvgCdia = `AvgCdia (m)`) %>% select(

summary(d$Leaf)
sum(d$Leaf == -1) / length(d$Leaf)
sum(d$AvgCdia == -1) / length(d$AvgCdia)
sum(d$DBH == -1) / length(d$DBH)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   -1.0    12.9   116.1   302.5   389.2  9516.0
[1] 0.1143094
[1] 0.003244288
[1] 0
```

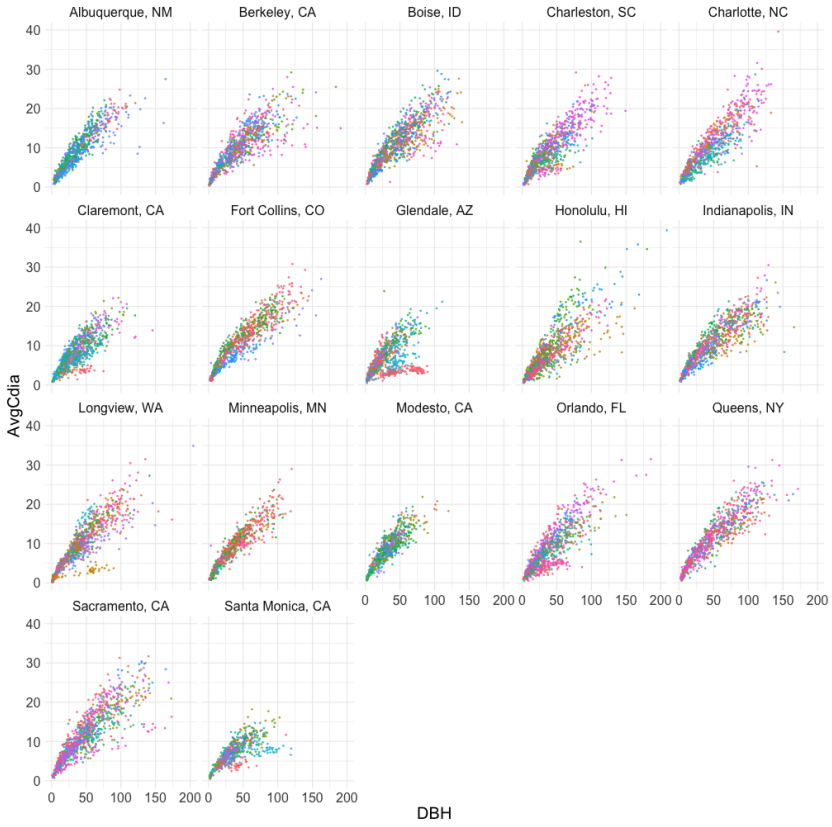
```
d <- filter(d, AvgCdia != -1)
```

4.4 plot dbh versus average crown diamter for all trees



4.5 DBH versus Cdia by cities

Constraining scales to 0-200 dbh and 0-40 AvgCdia. Much of variability within regions is due to species, colored in the plot below.


Albuquerque, NM
Berkeley, CA
Boise, ID
Charleston, SC
Charlotte, NC
Claremont, CA
Fort Collins, CO
Glendale, AZ
Honolulu, HI
Indianapolis, IN
Longview, WA
Minneapolis, MN
Modesto, CA
Orlando, FL
Queens, NY
Sacramento, CA
Santa Monica, CA
AvgCdia
DBH
0
10
20
30
40
0
50
100
150
200

4.6 DBH versus Cdia by species

Woah, there are lots of species. Clearly there is some variability in the relationship between dbh and crown diameter across species.

4.7 DBH versus Cdia by species for species that are in more than 1 city

Does the relationship between DBH and AvgCdia for a species change depending on the city where it is?

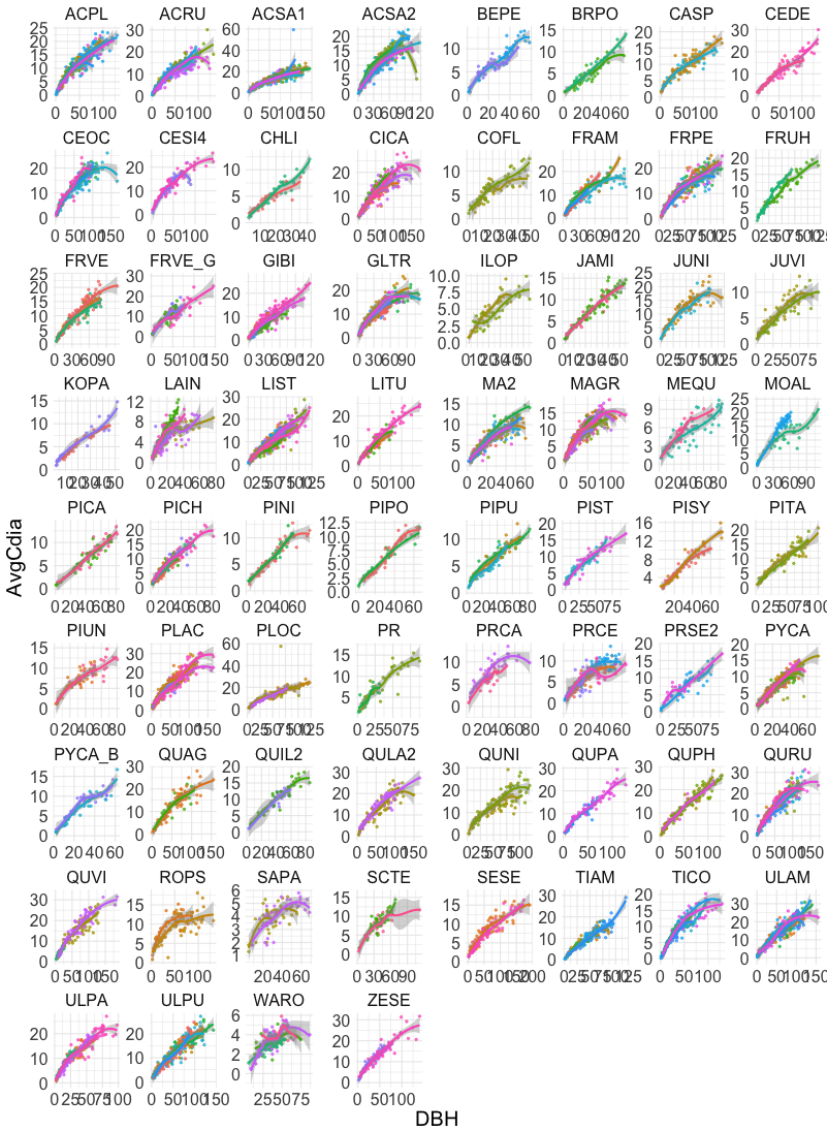
Is there evidence for a different equation for every species city combination? Or can we use one equation for each species, regardless of city?

```
  sp.w.multiple.cities <- d %>% group_by(City, SpCode) %>% summarize(n = n()) %>% ungroup()
    summarize(n_cities_per_species = n()) %>%
    filter(n_cities_per_species > 1) %>%
    pull(SpCode)

d.sp.w.multiple.cities <- filter(d, SpCode %in% sp.w.multiple.cities)
```

4.7.1 Each panel is a species, Each color is a different city

4.7.2 Adding Loess trend lines

It looks like some species have the same relationship (e.g. ACPL), others may have a different relationship (e.g. LAIN, MOAL), and for some it is hard to tell because the data don't fully overlap (e.g. LITU). This is a very informal assessment, but I think there's something here. Similarities are likely due to the cities having similar climates. Differences are likely due to cities having different climates. There could be a genetic component too.



ACPL's cities

```
filter(d.sp.w.multiple.cities, SpCode == "ACPL") %>% pull(City) %>% unique
```

```
[1] "Fort Collins, CO" "Minneapolis, MN"  "Indianapolis, IN" "Queens, NY"
[5] "Boise, ID"        "Longview, WA"
```

MOAL's cities

```
filter(d.sp.w.multiple.cities, SpCode == "MOAL") %>% pull(City) %>% unique
```

```
[1] "Glendale, AZ" "Longview, WA"
```

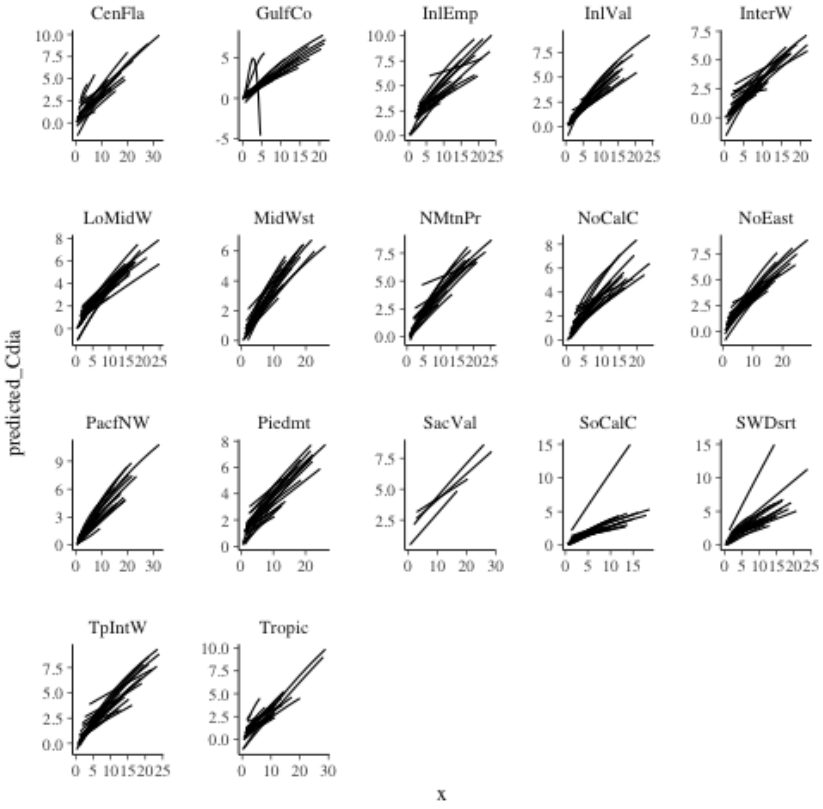
4.8 Concluding thoughts

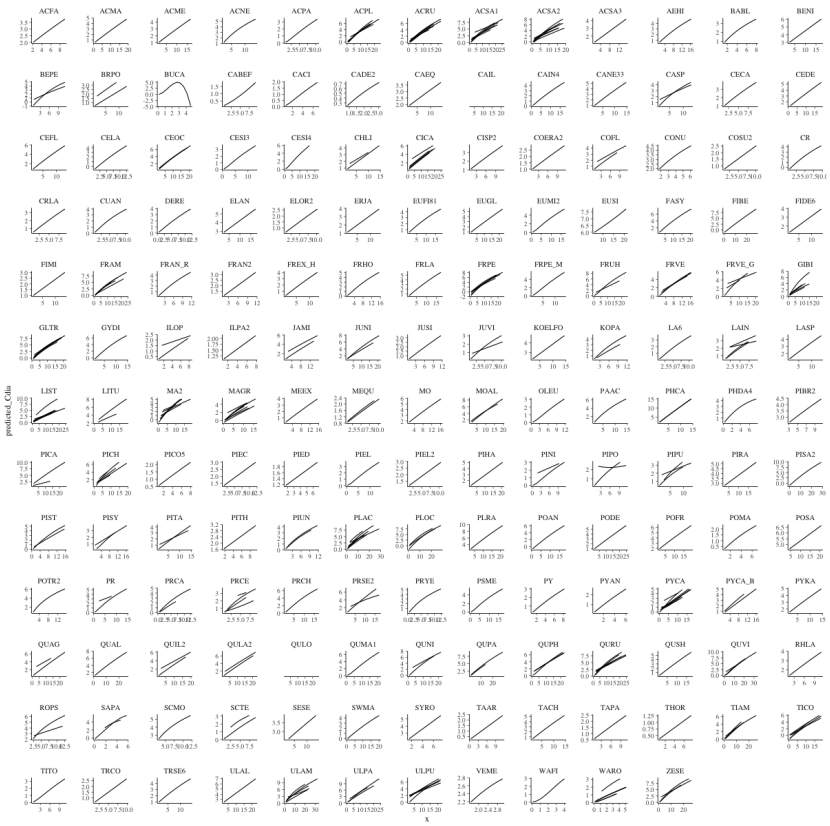
Clearly the relationship between DBH and AvgCdia varies with species and with location. However, for many species there is little

4.9 Plot Urban Tree Allometric equations on top of data

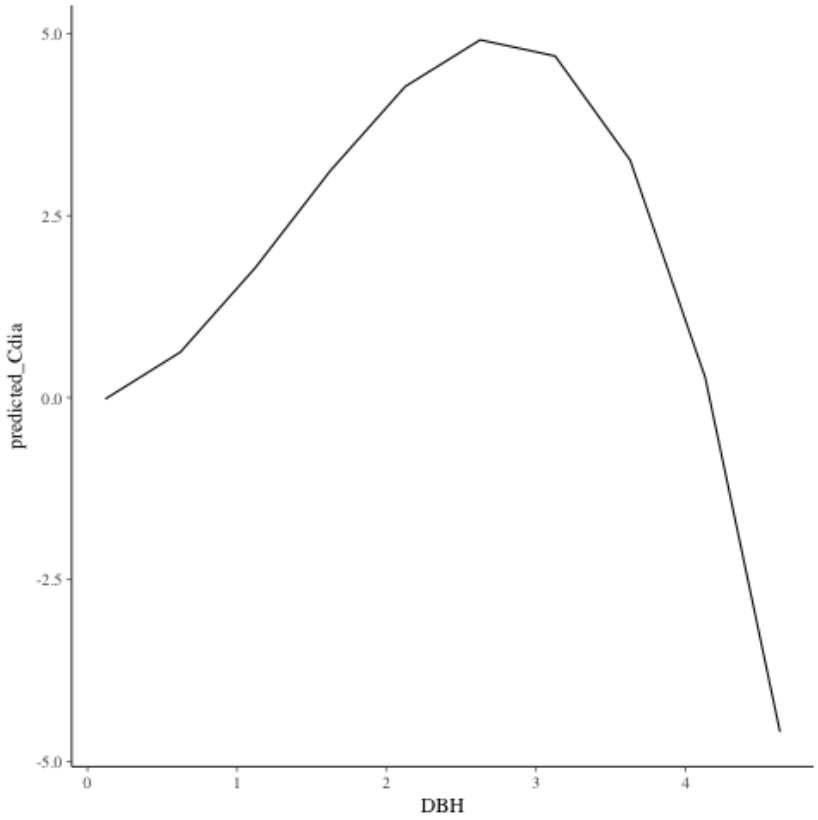
```
predict.allo <- function(x, EqName, a, b, c, d, e) {
    if(EqName == "loglogw1") {
        y = exp(a + b*log(log(x + 1) + c/2))
        }
    else if(EqName == "loglogw2") {
      y = exp(a + b*log(log(x + 1) + (sqrt(x) * (c/2))))
    }
    else if (EqName == "loglogw3") {
      y = exp(a + b*log(log(x + 1) + x * c/2))
    }
    else if (EqName == "loglogw4") {
      y = exp(a + b*log(log(x + 1) + x^2 * c/2))
    }
    else if (EqName == "expow1") {
        y = exp(a+ b * (x) + (c/2))
    }
    else if (EqName == "lin") {
            y = a + b * x
        }
    else if (EqName == "quad") {
            y = a + b * x + c* x^2
        }
    else if (EqName == "cub") {
            y = a+b * x+c *x^2 + d * x^3
        }
    else if (EqName == "quart") {
            y = a+b * x+c *x^2 + d * x^3 + e * x^4
        }
    return(y)
}


  eqn <- read.csv("data/RDS-2016-0005/Data/TS6_Growth_coefficients_fromNatalie.csv", strings
        mutate(a = as.numeric(a))

    eqn <- eqn %>%
        filter(Predicts.component %in% c("crown dia"), Independent.variable == "dbh")

  newdata <- lapply(1:nrow(eqn), function(i) {
      x <- seq(eqn$Apps.min[i], eqn$Apps.max[i],  (eqn$Apps.max[i] - eqn$Apps.min[i]) / 9)
```

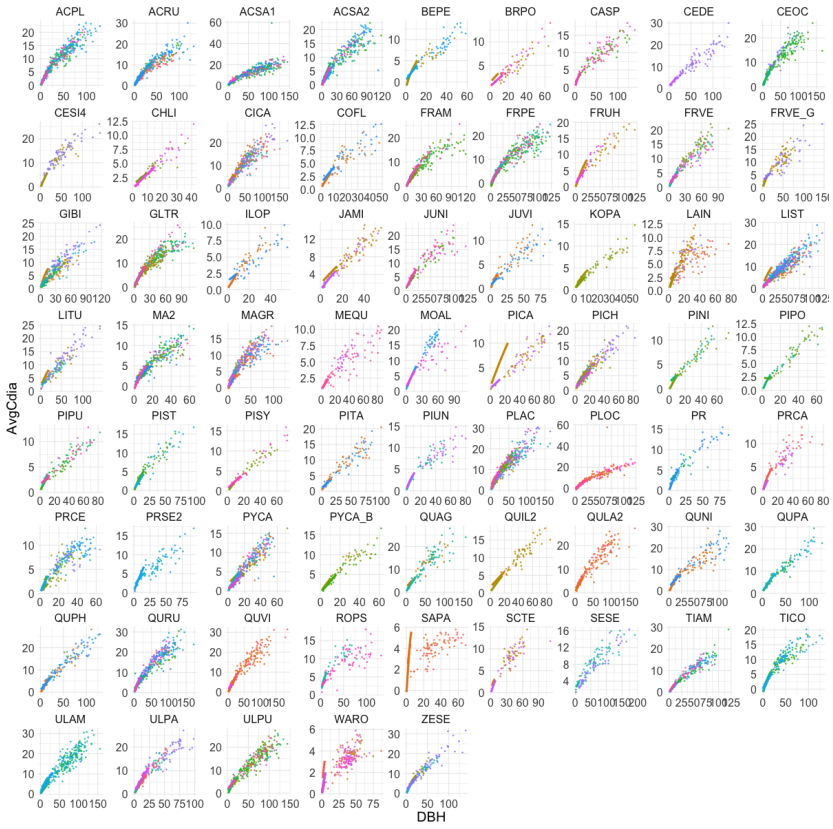
```
      cbind(x, eqn[i,])
      })

newdata <- bind_rows(newdata)

  predictions <- newdata %>% rowwise %>% mutate(predicted_Cdia = predict.allo(x = x, EqName

There were 50 or more warnings (use warnings() to see the first 50)
```





What's up with BUCA and PIPO? Those equations seem wrong.


5.0
2.5
0.0
-2.5
-5.0
predicted_Cdia
0
1
2
3
4
DBH



4.10 Model

the slope between dbh and the other tree characteristic would be a function of the species characteristics and the city's climate.

4.11 modelling

Subset down to 5 species

```
sp.sub <- c("ACPL", "QURU", "PIST", "CEOC", "FRPE")
ds <- filter(d, SpCode %in% sp.sub)
```

4.11.1 lme4

```
library(lme4)
```

```
Loading required package: Matrix
```

```
d <- mutate(ds, logAvgCdia = log(AvgCdia))
mod <- lmer(logAvgCdia ~ log(DBH) | SpCode, data = ds)
```

```
mod

Linear mixed model fit by REML ['lmerMod']
Formula: logAvgCdia ~ log(DBH) | SpCode
   Data: ds
REML criterion at convergence: -439.0436
Random effects:
 Groups   Name        Std.Dev. Corr
 SpCode   (Intercept) 4.5990
          log(DBH)    0.7474   -1.00
 Residual             0.1977
Number of obs: 1224, groups:  SpCode, 5
Fixed Effects:
(Intercept)
      4.168
```

4.11.2 rstanarm

```
library(rstanarm)
options(mc.cores = parallel::detectCores())

smod <- stan_glm(log(AvgCdia) ~ DBH, data = d, family = gaussian(link = "identity"))


SAMPLING FOR MODEL 'continuous' NOW (CHAIN 1).

Gradient evaluation took 0.003692 seconds
1000 transitions using 10 leapfrog steps per transition would take 36.92 seconds.
Adjust your expectations accordingly!


Iteration:    1 / 2000 [  0%]  (Warmup)

SAMPLING FOR MODEL 'continuous' NOW (CHAIN 2).

Gradient evaluation took 0.003201 seconds
1000 transitions using 10 leapfrog steps per transition would take 32.01 seconds.
Adjust your expectations accordingly!


Iteration:    1 / 2000 [  0%]  (Warmup)

SAMPLING FOR MODEL 'continuous' NOW (CHAIN 3).

Gradient evaluation took 0.005202 seconds
1000 transitions using 10 leapfrog steps per transition would take 52.02 seconds.
Adjust your expectations accordingly!
```

```
Iteration:    1 / 2000 [  0%]  (Warmup)

SAMPLING FOR MODEL 'continuous' NOW (CHAIN 4).

Gradient evaluation took 0.006159 seconds
1000 transitions using 10 leapfrog steps per transition would take 61.59 seconds.
Adjust your expectations accordingly!


Iteration:    1 / 2000 [  0%]  (Warmup)
Iteration:  200 / 2000 [ 10%]  (Warmup)
Iteration:  200 / 2000 [ 10%]  (Warmup)
Iteration:  200 / 2000 [ 10%]  (Warmup)
Iteration:  200 / 2000 [ 10%]  (Warmup)
Iteration:  400 / 2000 [ 20%]  (Warmup)
Iteration:  400 / 2000 [ 20%]  (Warmup)
Iteration:  400 / 2000 [ 20%]  (Warmup)
Iteration:  400 / 2000 [ 20%]  (Warmup)
Iteration:  600 / 2000 [ 30%]  (Warmup)
Iteration:  600 / 2000 [ 30%]  (Warmup)
Iteration:  600 / 2000 [ 30%]  (Warmup)
Iteration:  600 / 2000 [ 30%]  (Warmup)
Iteration:  800 / 2000 [ 40%]  (Warmup)
Iteration:  800 / 2000 [ 40%]  (Warmup)
Iteration:  800 / 2000 [ 40%]  (Warmup)
Iteration:  800 / 2000 [ 40%]  (Warmup)
Iteration: 1000 / 2000 [ 50%]  (Warmup)
Iteration: 1001 / 2000 [ 50%]  (Sampling)
Iteration: 1000 / 2000 [ 50%]  (Warmup)
Iteration: 1001 / 2000 [ 50%]  (Sampling)
Iteration: 1000 / 2000 [ 50%]  (Warmup)
Iteration: 1001 / 2000 [ 50%]  (Sampling)
Iteration: 1000 / 2000 [ 50%]  (Warmup)
Iteration: 1001 / 2000 [ 50%]  (Sampling)
Iteration: 1200 / 2000 [ 60%]  (Sampling)
Iteration: 1200 / 2000 [ 60%]  (Sampling)
Iteration: 1200 / 2000 [ 60%]  (Sampling)
Iteration: 1200 / 2000 [ 60%]  (Sampling)
Iteration: 1400 / 2000 [ 70%]  (Sampling)
Iteration: 1400 / 2000 [ 70%]  (Sampling)
Iteration: 1400 / 2000 [ 70%]  (Sampling)
Iteration: 1400 / 2000 [ 70%]  (Sampling)
Iteration: 1600 / 2000 [ 80%]  (Sampling)
Iteration: 1600 / 2000 [ 80%]  (Sampling)
```

```
Iteration: 1600 / 2000 [ 80%]  (Sampling)
Iteration: 1600 / 2000 [ 80%]  (Sampling)
Iteration: 1800 / 2000 [ 90%]  (Sampling)
Iteration: 1800 / 2000 [ 90%]  (Sampling)
Iteration: 1800 / 2000 [ 90%]  (Sampling)
Iteration: 1800 / 2000 [ 90%]  (Sampling)
Iteration: 2000 / 2000 [100%]  (Sampling)

 Elapsed Time: 19.5348 seconds (Warm-up)
               17.1469 seconds (Sampling)
               36.6817 seconds (Total)

Iteration: 2000 / 2000 [100%]  (Sampling)

 Elapsed Time: 19.8768 seconds (Warm-up)
               16.8117 seconds (Sampling)
               36.6885 seconds (Total)

Iteration: 2000 / 2000 [100%]  (Sampling)

 Elapsed Time: 21.2892 seconds (Warm-up)
               16.3886 seconds (Sampling)
               37.6777 seconds (Total)

Iteration: 2000 / 2000 [100%]  (Sampling)

 Elapsed Time: 19.5799 seconds (Warm-up)
               17.5401 seconds (Sampling)
               37.12 seconds (Total)

summary(smod)


Model Info:

 function:     stan_glm
 family:       gaussian [identity]
 formula:      log(AvgCdia) ~ DBH
 algorithm:    sampling
 priors:       see help('prior_summary')
 sample:       4000 (posterior sample size)
 observations: 14440
 predictors:   2

Estimates:
                mean    sd      2.5%    25%     50%     75%     97.5%
```

```
(Intercept)       1.2     0.0     1.2     1.2     1.2     1.2     1.2
DBH               0.0     0.0     0.0     0.0     0.0     0.0     0.0
sigma             0.4     0.0     0.4     0.4     0.4     0.5     0.5
mean_PPD          2.0     0.0     2.0     2.0     2.0     2.0     2.1
log-posterior -8910.8     1.2 -8914.0 -8911.4 -8910.5 -8909.9 -8909.4

Diagnostics:
              mcse Rhat n_eff
(Intercept)   0.0  1.0  3986
DBH           0.0  1.0  4000
sigma         0.0  1.0  2642
mean_PPD      0.0  1.0  3693
log-posterior 0.0  1.0  1848

For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effectiv
print(smod, digits = 5)
smod_Sp <- stan_glmer(log(AvgCdia) ~ DBH | SpCode, data = d, family = gaussian(link = "ident
Error: Using '|' in model formula not allowed. Maybe you meant to use 'stan_(g)lmer'?
```

posterior predict with exp transform

5 scratch

Region	GulfCo
Scientific.Name	Butia capitata
SpCode	BUCA
Independent.variable	dbh
Predicts.component	crown dia
Units.of.predicted.components	meters
Model.weight	$1/ht^2$
EqName	cub
a	-0.0575
b	0.10884
c	1.87801
d	-0.45639
e	
Apps.min	0.12
Apps.max	4.63
Sigma	
n	
adj.R2	
Data.min..from.raw.data.	0.7
Data.max..from.raw.data.	7.4
DF	