Pondo_Allom_Params

Katie Murenbeeld

3/26/2020

Introduction:

Information about allometric curves and functions used in FATES. https://fates-docs.readthedocs.io/en/latest/fates_tech_note.html#allometry-and-growth-along-allometric-curves

Install the BAAD Package:

Install the package from github:

```r
#install.packages("devtools")
#devtools::install_github("richfitz/datastorr")
#devtools::install_github("traitecoevo/baad.data")
```

Review the data:

```r
baad <- baad.data::baad_data()
d_baad <- baad$data
head(d_baad)
```

```
##   studyName      location latitude longitude vegetation map mat grouping
## 1   Abe1981 Japan-Oumu-46       45       143       BorF  NA  NA     <NA>
## 2   Abe1981 Japan-Oumu-46       45       143       BorF  NA  NA     <NA>
## 3   Abe1981 Japan-Oumu-46       45       143       BorF  NA  NA     <NA>
## 4   Abe1981 Japan-Oumu-46       45       143       BorF  NA  NA     <NA>
## 5   Abe1981 Japan-Oumu-46       45       143       BorF  NA  NA     <NA>
## 6   Abe1981 Japan-Oumu-46       45       143       BorF  NA  NA     <NA>
##   lai             species      speciesMatched   family pft
## 1  NA Abies sachalinensis Abies sachalinensis Pinaceae  EA
## 2  NA Abies sachalinensis Abies sachalinensis Pinaceae  EG
## 3  NA Abies sachalinensis Abies sachalinensis Pinaceae  EG
## 4  NA Abies sachalinensis Abies sachalinensis Pinaceae  EG
## 5  NA Abies sachalinensis Abies sachalinensis Pinaceae  EG
## 6  NA Abies sachalinensis Abies sachalinensis Pinaceae  EG
##   growingCondition status light age a.lf a.ssba a.ssbh a.ssbc a.shba
## 1               PM     NA  <NA>  NA   NA     NA     NA     NA     NA
## 2               PM     NA  <NA>  NA   NA     NA     NA     NA     NA
## 3               PM     NA  <NA>  NA   NA     NA     NA     NA     NA
## 4               PM     NA  <NA>  NA   NA     NA     NA     NA     NA
## 5               PM     NA  <NA>  NA   NA     NA     NA     NA     NA
## 6               PM     NA  <NA>  NA   NA     NA     NA     NA     NA
##   a.shbh a.shbc a.sbba a.sbbh a.sbbc     a.stba     a.stbh      a.stbc
## 1     NA     NA     NA     NA     NA 0.18973040 0.13074052 0.066508302
## 2     NA     NA     NA     NA     NA 0.14286569 0.09731397 0.056410438
## 3     NA     NA     NA     NA     NA 0.19658523 0.07068583 0.035298935
## 4     NA     NA     NA     NA     NA 0.07068583 0.04523893 0.020611989
## 5     NA     NA     NA     NA     NA 0.02198274 0.01583677 0.008992024
## 6     NA     NA     NA     NA     NA 0.18957602 0.12692348 0.059828490
```

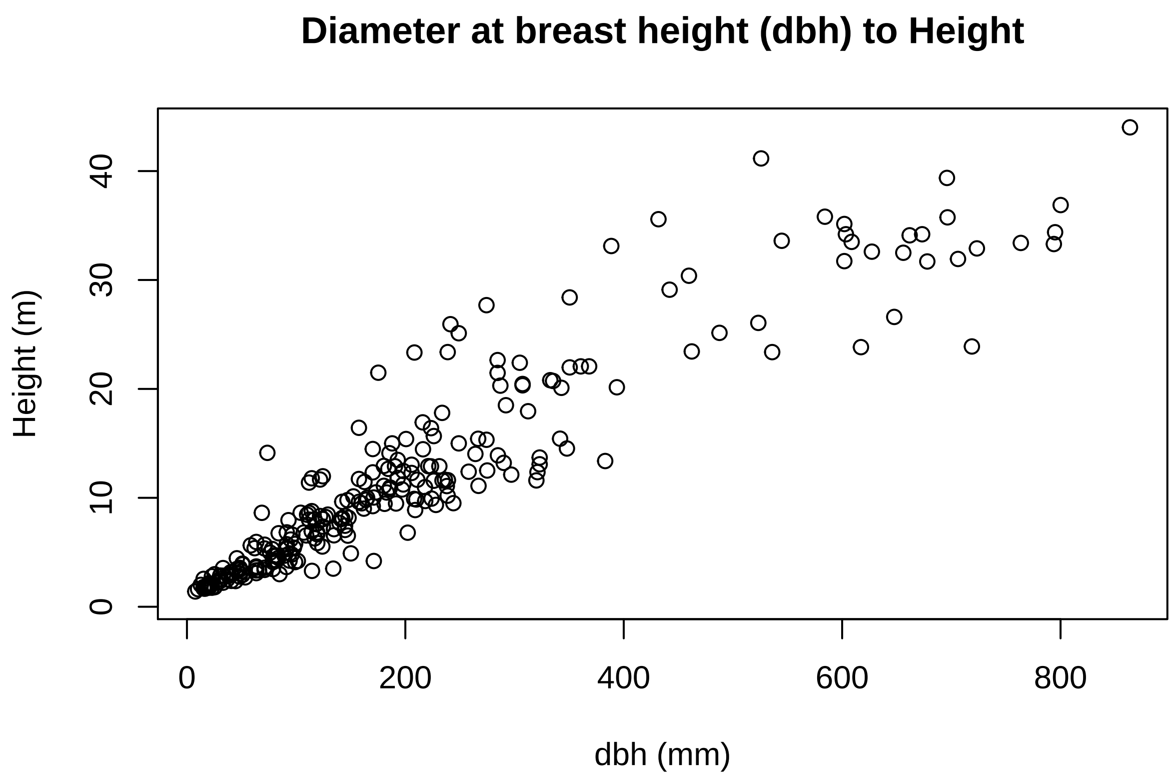
```
##   a.cp a.cs   h.t   h.c   d.ba  d.bh h.bh d.cr   c.d m.lf m.ss m.sh m.sb
## 1   NA   NA 21.96  8.90 0.4915 0.408  1.3   NA 13.06 49.2   NA   NA   NA
## 2   NA   NA 19.70  8.02 0.4265 0.352  1.3   NA 11.68 28.0   NA   NA   NA
## 3   NA   NA 20.10  9.50 0.5003 0.300  1.3   NA 10.60 19.9   NA   NA   NA
## 4   NA   NA 19.23 11.60 0.3000 0.240  1.3   NA  7.63 11.1   NA   NA   NA
## 5   NA   NA 16.14  9.35 0.1673 0.142  1.3   NA  6.79  3.8   NA   NA   NA
## 6   NA   NA 22.26 11.02 0.4913 0.402  1.3   NA 11.24 33.4   NA   NA   NA
##    m.st  m.so  m.br m.rf m.rc m.rt m.to a.ilf ma.ilf r.st r.ss r.sb r.sh
## 1 516.8 566.0 113.8   NA   NA   NA   NA    NA     NA   NA   NA   NA   NA
## 2 427.7 455.7  84.9   NA   NA   NA   NA    NA     NA   NA   NA   NA   NA
## 3 255.0 274.9  30.2   NA   NA   NA   NA    NA     NA   NA   NA   NA   NA
## 4 172.9 184.0  14.4   NA   NA   NA   NA    NA     NA   NA   NA   NA   NA
## 5  58.6  62.4   4.6   NA   NA   NA   NA    NA     NA   NA   NA   NA   NA
## 6 561.9 595.3  61.3   NA   NA   NA   NA    NA     NA   NA   NA   NA   NA
##   n.lf n.ss n.sb n.sh n.rf n.rc
## 1   NA   NA   NA   NA   NA   NA
## 2   NA   NA   NA   NA   NA   NA
## 3   NA   NA   NA   NA   NA   NA
## 4   NA   NA   NA   NA   NA   NA
## 5   NA   NA   NA   NA   NA   NA
## 6   NA   NA   NA   NA   NA   NA
```

Refine (subset) the data for Pinus Ponderosa:

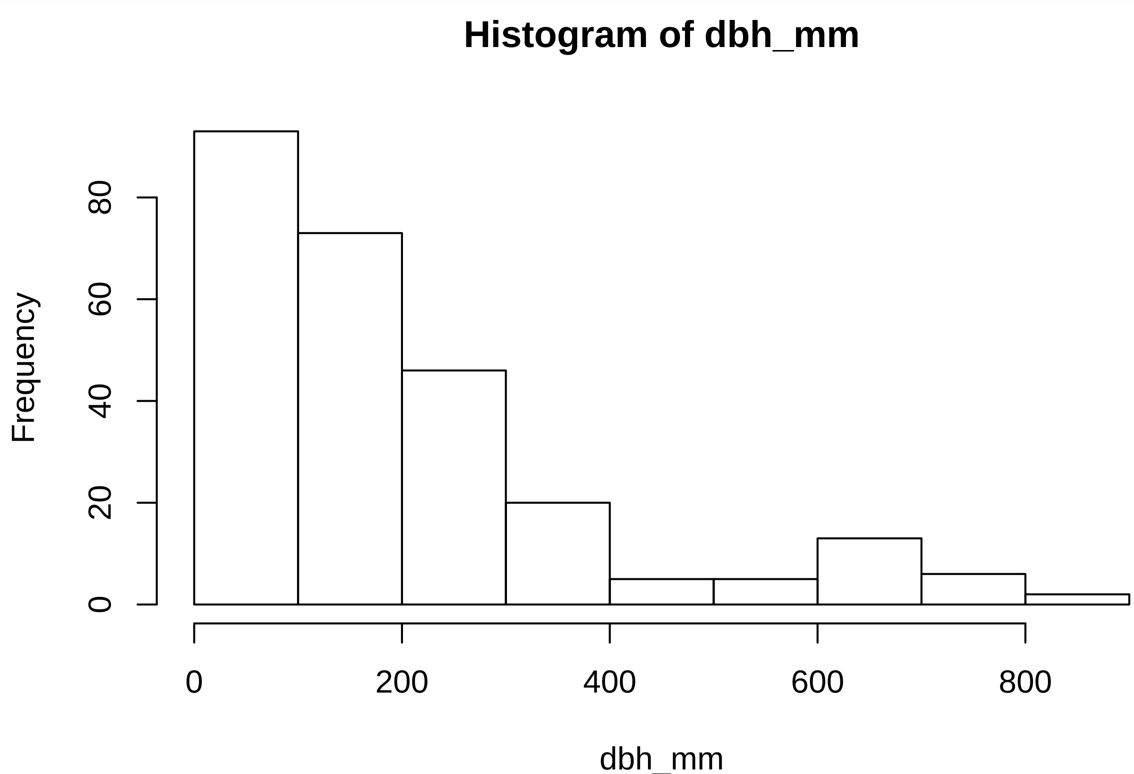
```r
pipo <- d_baad[ which(d_baad$species == 'Pinus ponderosa'), ]
#head(pipo)
```

Plot raw data, dbh to height.

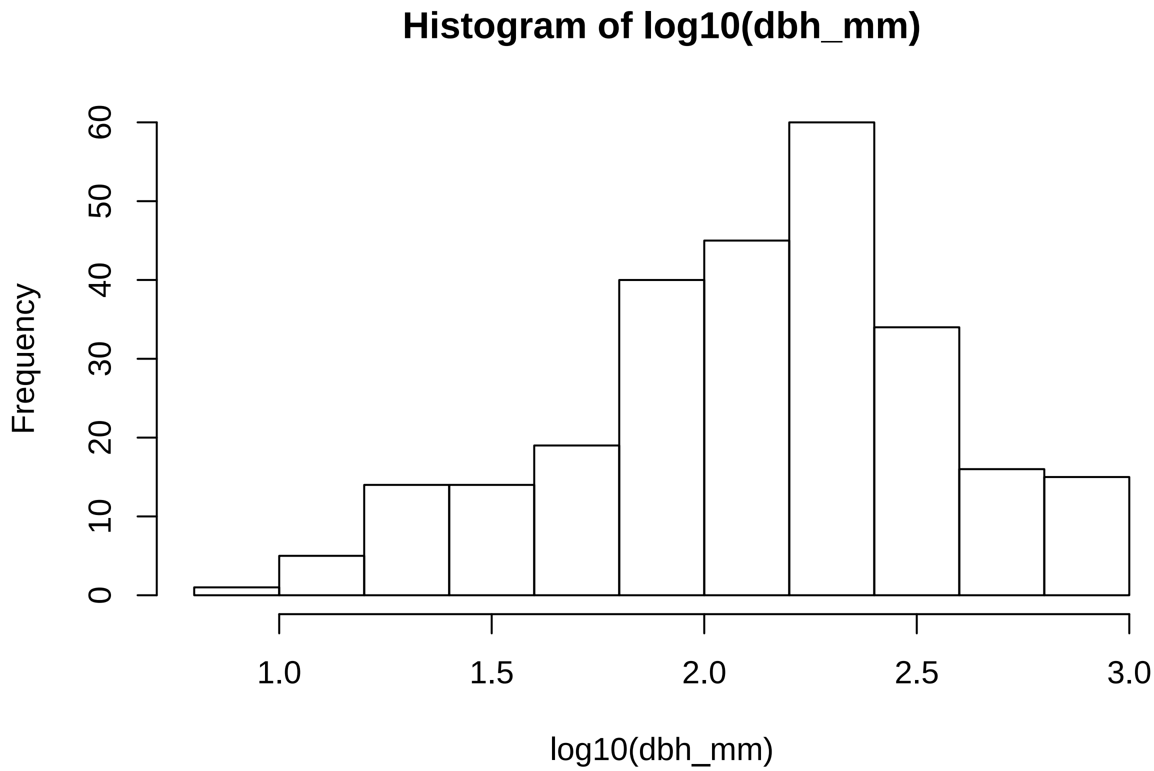
```r
# Set variables from the data.

# Diameter at breat height (dbh) will be used for most allometric calculations
dbh_mm <- (pipo$d.bh)*1000
dbh_cm <- (pipo$d.bh)*100
# Height
h <- (pipo$h.t)

plot(h~dbh_mm,
     main="Diameter at breast height (dbh) to Height",
     xlab = "dbh (mm)",
     ylab = "Height (m)")
```




```r
hist(dbh_mm)
```




```r
hist(log10(dbh_mm))
```


Histogram of log10(dbh_mm)
Frequency
0
10
20
30
40
50
60
1.0
1.5
2.0
2.5
3.0
log10(dbh_mm)

Diameter at breast height (dbh) to height - d2h

FATES allows for four different approaches to predicting height from dbh:

- A power function
$$h = p_1 * d^(p_2)$$
- O'Brien et al. (1995)
$$log_{10}Height = log_{10}DBH * slope + intercept$$
- Poorter et al. (2006)
$$h = p_1 * (1 - exp(p_2 * d^(p_3)))$$
- Martinex Cano et al. (2019)
$$h = (p_1 * d^(p_2)/(p_3 + d^(p_2)))$$

For my research I use the dbh to height relationship from O'Brien et al., 1995. The log10Height (m) is regressed on log10DBH (mm). Where slope is p1 or fates_allom_d2h1 and intercept is p2 or fates_allom_d2h2 in the parameter file.

$$log_{10}Height = log_{10}DBH * slope + intercept$$

So modeled height would be. I want to solve for the best fit p1 and p2 given height and dbh data from BAAD.

$$Height = 10^{(log10(min(d,dbhmax))*p1+p2)}$$

```
# But first I will test out just a linear model
d2hmod_mm <- lm(h~dbh_mm)
coef(d2hmod_mm)
```

```
## (Intercept)      dbh_mm
##  1.88991834  0.04858541
```

```
# Then a linear model with log10(h) regressed on log10(dbh)
d2hmodlog_mm <- lm(log10(h)~log10(dbh_mm))
coef(d2hmodlog_mm)
```

```
##   (Intercept) log10(dbh_mm)
##    -0.8150832     0.8181127
```

```
d2hmodlog_cm <- lm(log10(h)~log10(dbh_cm))
coef(d2hmodlog_cm)
```

```
##   (Intercept) log10(dbh_cm)
##   0.003029536   0.818112721
```

Next, I will create a function to represent the O'Brien calculation used for dbh to height relationships.

```
obrien <- function(dbh, p1, p2){
  height <- 10^((log10(dbh))* p1 + p2)
  return(height)
}

dbins_mm <- c(100,200,300,400,500,600,700,800,900) # Remember in O'Brien dbh is in mm.
dbins_cm <- c(10,20,30,40,50,60,70,80,90) # But does FATES do that conversion or use cm?

# Use default parameters from O'Brien et al 1995
default_mm <- obrien(dbins_mm, 0.7, -0.2)
```

```r
# Use the parameters from the log10 linear model d2hmodlog
lmfit_mm <- obrien(dbins_mm, 0.82, -0.82) # dbh in mm
lmfit_cm <- obrien(dbins_cm, 0.82, 0.003) # dbh in cm

coef(lm(default_mm~dbins_mm))
```

```
## (Intercept)    dbins_mm
## 11.67774249  0.07124032
```

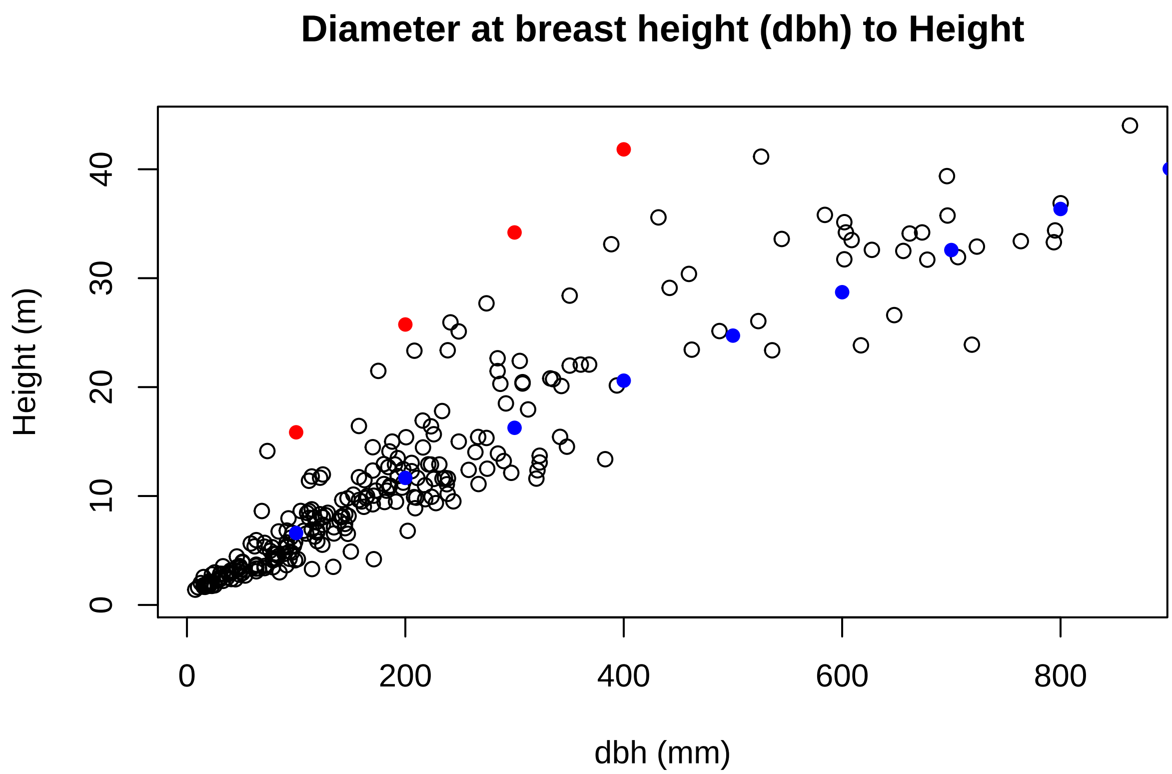
```r
coef(lm(lmfit_mm~dbins_mm))
```

```
## (Intercept)    dbins_mm
##  3.45859976  0.04142493
```

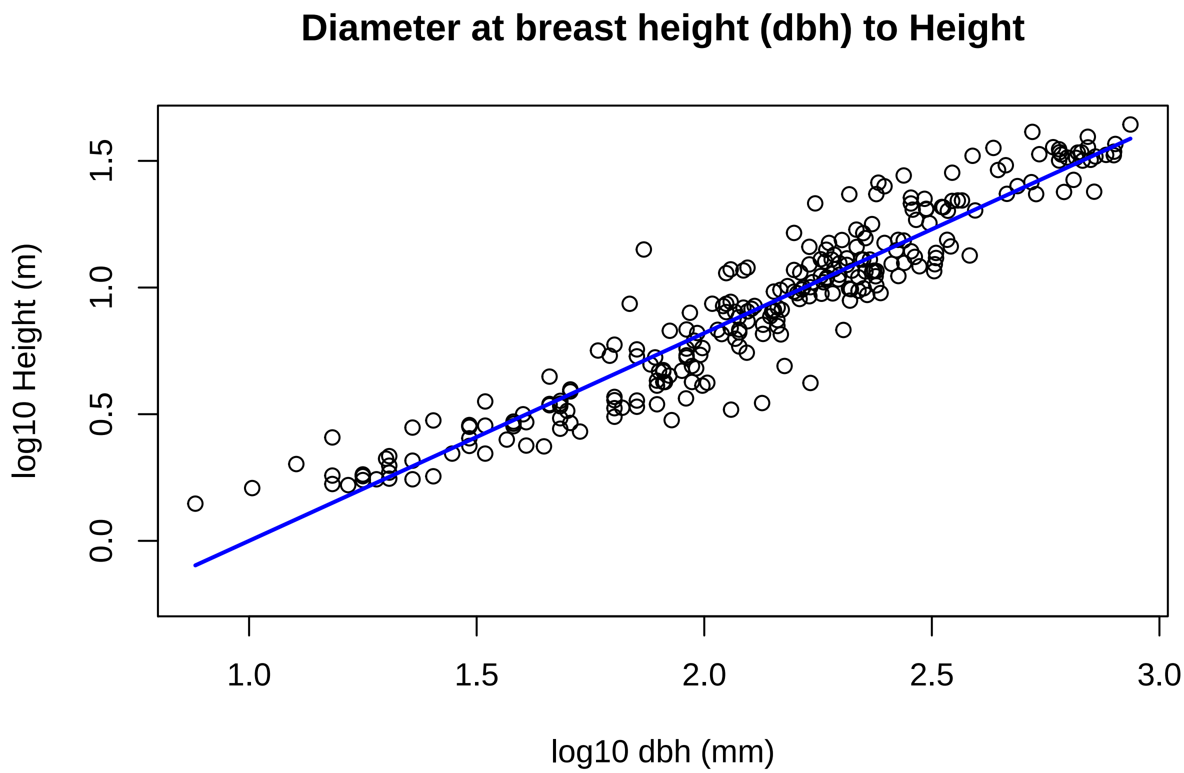
```r
coef(lm(lmfit_cm~dbins_cm))
```

```
## (Intercept)    dbins_cm
##   3.4825736   0.4171207
```

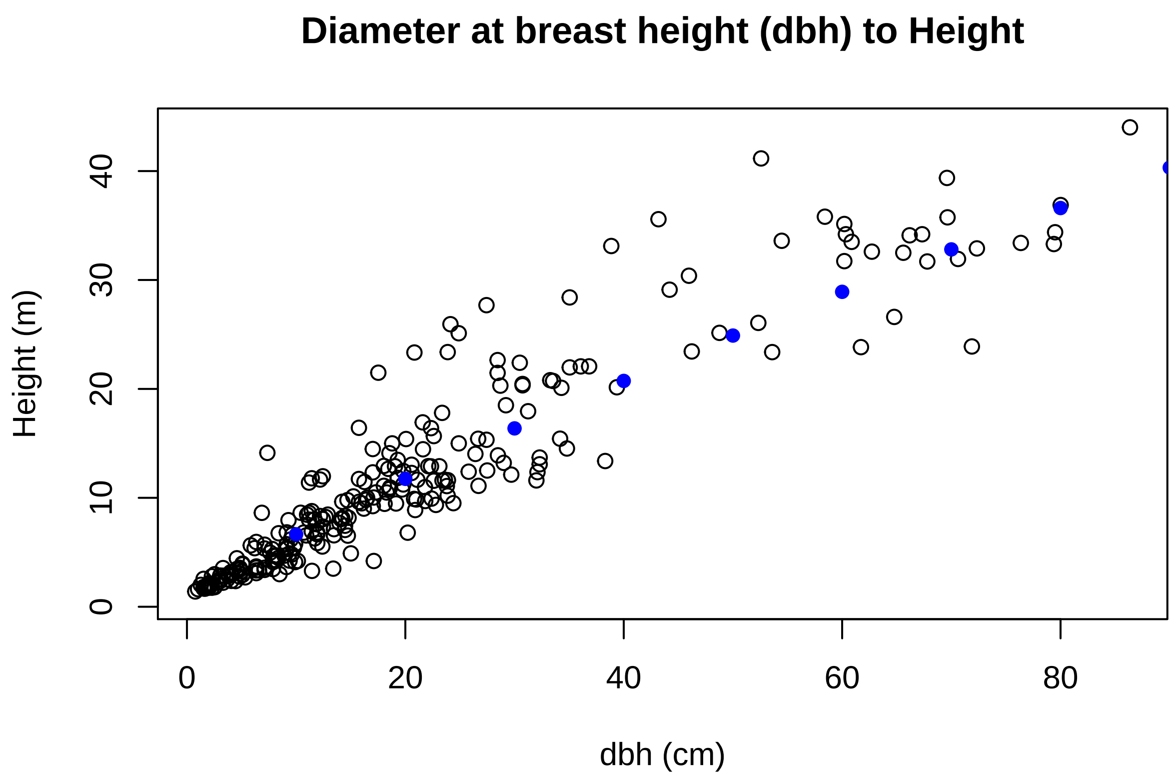
```r
plot(h~dbh_mm,
     main="Diameter at breast height (dbh) to Height",
     xlab = "dbh (mm)",
     ylab = "Height (m)")

points(default_mm~dbins_mm, col="red", pch=16) # default O'Brien slope (0.7) and intercept (-0.2)
points(lmfit_mm~dbins_mm, col="blue", pch=16) # BAAD data slope (0.82) and intercept (-0.82)
```



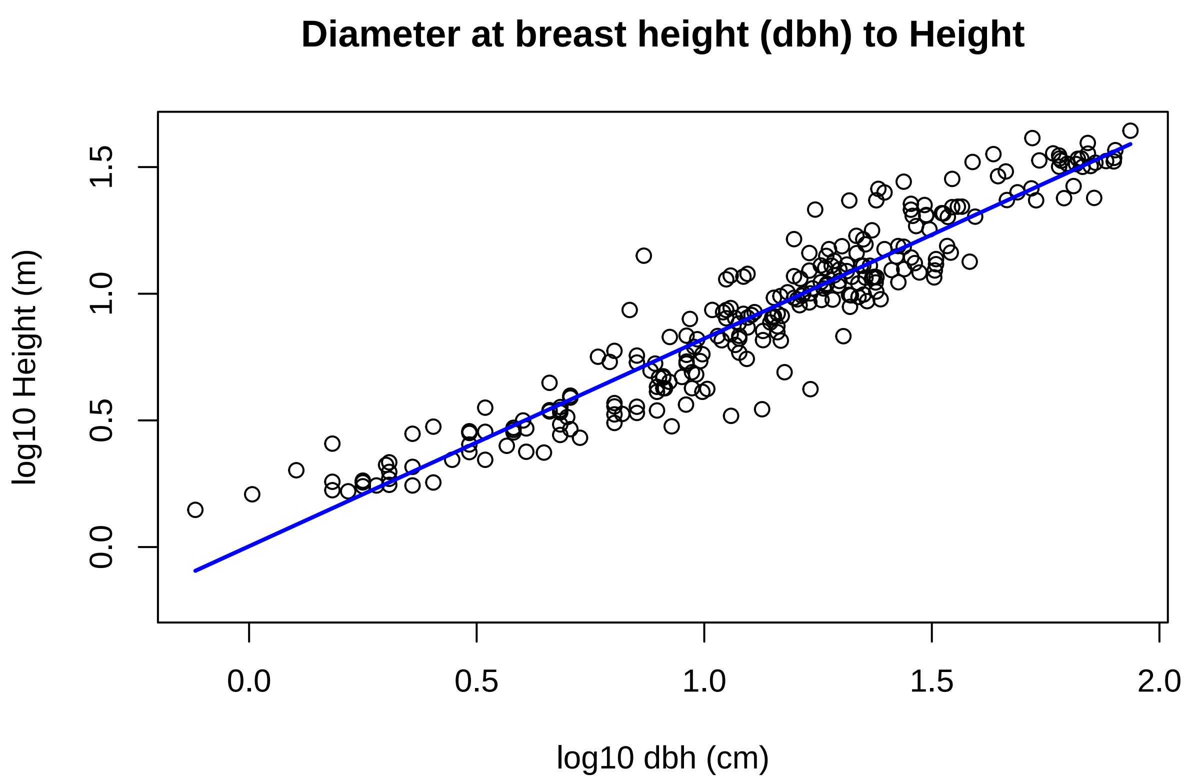

```r
plot(log10(h)~log10(dbh_mm),
     main="Diameter at breast height (dbh) to Height",
     xlab = "log10 dbh (mm)",
     ylab = "log10 Height (m)")
curve(-0.82 + (0.82*x), add=T, col="blue", lwd=2) # log regression coefficients from d2hmodlog
```




```r
plot(h~dbh_cm,
     main="Diameter at breast height (dbh) to Height",
     xlab = "dbh (cm)",
     ylab = "Height (m)")

points(default_mm~dbins_mm, col="red", pch=16) # default O'Brien slope (0.7) and intercept (-0.2)
points(lmfit_cm~dbins_cm, col="blue", pch=16) # BAAD data slope (0.82) and intercept (-0.82)
```




```r
plot(log10(h)~log10(dbh_cm),
     main="Diameter at breast height (dbh) to Height",
     xlab = "log10 dbh (cm)",
     ylab = "log10 Height (m)")
curve(0.003 + (0.82*x), add=T, col="blue", lwd=2) # log regression coefficients from d2hmodlog
```



Diameter at breast height (dbh) to above ground biomass (AGB) - d2bagw

FATES provides three different options for calculating AGB from dbh:

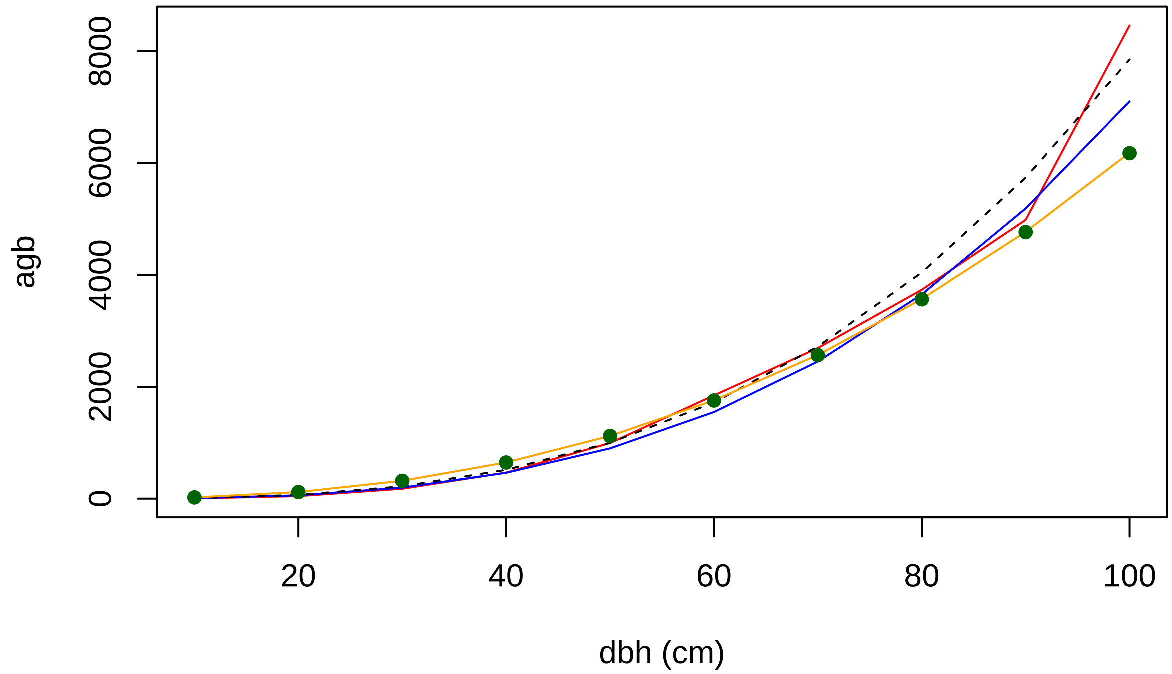
- Saldarriaga et al. (1998)
$$(C_{agb} = f_{agb} * p_1 * h^{p_2} * d^{p_3} * rho^{p_4})$$
- 2 parameter power function
$$(C_{agb} = p_1/c2b * d^{p_2})$$
- Chave et al. (2014)
$$(C_{agb} = p_1/c2b * (rho * d^2 * h)^{p_2})$$

For this project I will use Saldarriaga et al. (1998)

$$C_{agb} = f_{agb} * p_1 * h^{p_2} * d^{p_3} * rho^{p_4}$$

```
# In this case dbh and agb calculated from Chojnacky serves as the "observed data" to fit the other par

# Define the variables needed for the Saldarriaga function
rho = 0.367
f_agb = 0.6
d.agb <- c(10, 20, 30, 40, 50, 60, 70, 80, 90, 100)
h.agb <- c(2.82, 7.7, 13.9, 21.0, 29.1, 37.9, 37.9, 37.9, 37.9, 57.5)
#use heights from d2h
lmfit.h <- obrien(d.agb, 0.82, 0.003)

# Define the Saldarriaga function
sal <- function(f_agb, p1, h, p2, d, p3, rho, p4){
  agb <- f_agb * p1 * h^p2 * d^p3 * rho^p4
  return(agb)
}

# Define a 2 parameter power function (as a stand-in for Chojnakcy if those param values are used)
par2_pwr <- function(p1,d,p2,c2b){
  bagw <- (p1*(d^p2))/c2b
  return(bagw)
}

choj_eq <- function(b0,b1,d){
  biom <- b0  + (b1*log(d))
  return(biom)
}
choj_p1 = -2.6177
choj_p2 = 2.4638
c2b = 2

test_2par <- par2_pwr(0.146, d.agb, 2.464, c2b)
choj <- choj_eq(choj_p1, choj_p2, d.agb)

# Parameters from earlier Jupyter Notebook. Not sure how to determine the best fit for the Sal function
#p1 = 0.131
#p2 = 0.626
#p3 = 2.46
#p4 = 2.18

p1 = 0.11
```

```r
p2 = 0.65
p3 = 2.45
p4 = 2.15

sal1 <- sal(f_agb, p1, h.agb, p2, d.agb, p3, rho, p4)
sal2 <- sal(f_agb, p1, lmfit.h, p2, d.agb, p3, rho, p4)
sal3 <- sal(f_agb, 0.131, lmfit.h, 0.626, d.agb, 2.46, rho, 2.18)
#plot(choj.h~d.agb, type="l", col="blue")
plot(sal1~d.agb, col="red", type="l", ylab="agb", xlab="dbh (cm)")
points(sal2~d.agb, col="blue", type="l")
points(sal3~d.agb, col="black", lty=2, type="l")
points(test_2par~d.agb, col="orange", type="l")
points(exp(choj)~d.agb, col="darkgreen", pch=16)
```




```r
# Here I want to get the r2 or rmse to compare the different allom functions and params to the "observe

rmse <- function(y_hat,y){
  return(sqrt(mean((y-y_hat)^2)))
}

r2 <- function(y_hat,y){
  RSS<-sum((((y_hat))-(y))^2)
  TSS<-sum(((y)-(mean(y)))^2)
  return(1-RSS/TSS)}


# remember y_hat is the model predicted agb
# y observed = test_choj
# y_hat(s) = test (sal), test2 (sal), test_2par, py_test (sal)

sal1_rmse <- rmse(sal1,choj)
sal1_R2 <- r2(sal1, exp(choj))
```

```r
sal2_rmse <- rmse(sal2,choj)
sal2_R2 <- r2(sal2,exp(choj))

sal3_rmse <- rmse(sal3, choj)
sal3_R2 <- r2(sal3, exp(choj))

test_2par_rmse <- rmse(test_2par, choj)
test_2par_R2 <- r2(test_2par, exp(choj))

agb_rmse <- c(sal1_rmse, sal2_rmse, sal3_rmse, test_2par_rmse)
agb_r2 <- c(sal1_R2, sal2_R2, sal3_R2, test_2par_R2)
agb_names <- c("Sal test 1", "Sal test 2", "Sal test 3", "Two Param Power")

agb_mod_fits <- data.frame(agb_rmse, agb_r2, row.names = agb_names)

agb_mod_fits
```

```
##                 agb_rmse    agb_r2
## Sal test 1      3490.966 0.8675944
## Sal test 2      3159.715 0.9703919
## Sal test 3      3495.647 0.9004400
## Two Param Power 2913.377 0.9999965
```