

# 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](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:

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

## Review the data:

```
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.ssbh a.ssbh 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.shbh a.ssbh a.ssbh a.ssbh      a.stba      a.stbh      a.stbh
## 1      NA      NA      NA      NA      NA  NA 0.18973040 0.13074052 0.066508302
## 2      NA      NA      NA      NA      NA  NA 0.14286569 0.09731397 0.056410438
## 3      NA      NA      NA      NA      NA  NA 0.19658523 0.07068583 0.035298935
## 4      NA      NA      NA      NA      NA  NA 0.07068583 0.04523893 0.020611989
## 5      NA      NA      NA      NA      NA  NA 0.02198274 0.01583677 0.008992024
## 6      NA      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:

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

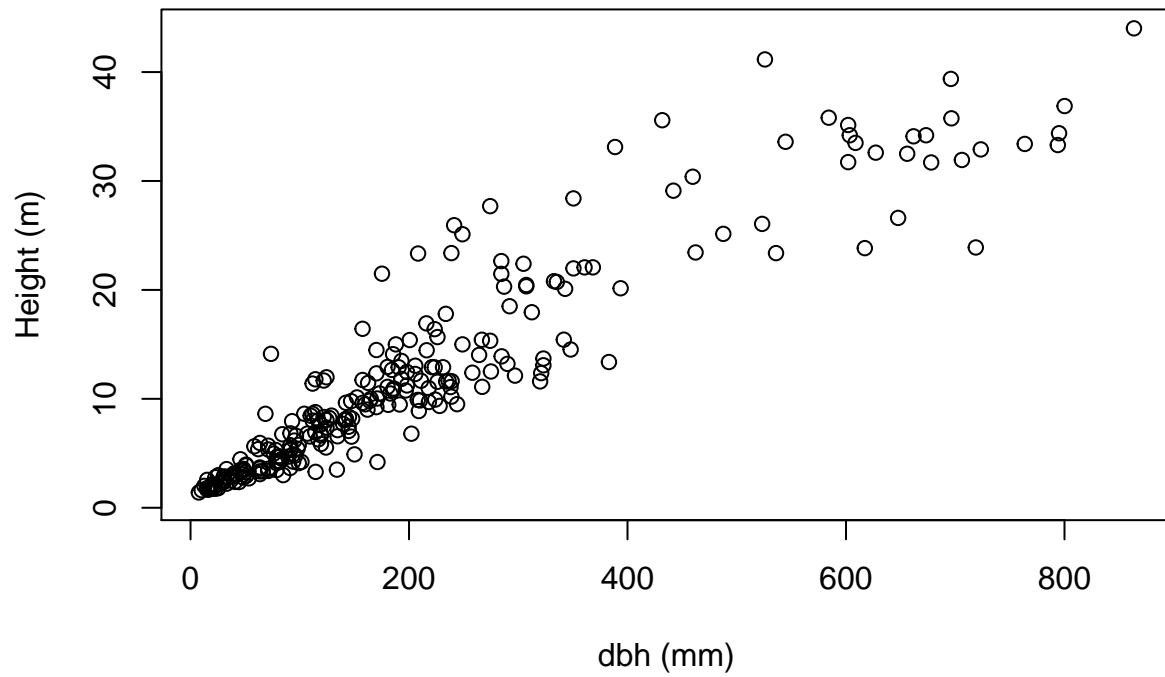
Plot raw data, dbh to height.

```
# Set variables from the data.

# Diameter at breast 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)

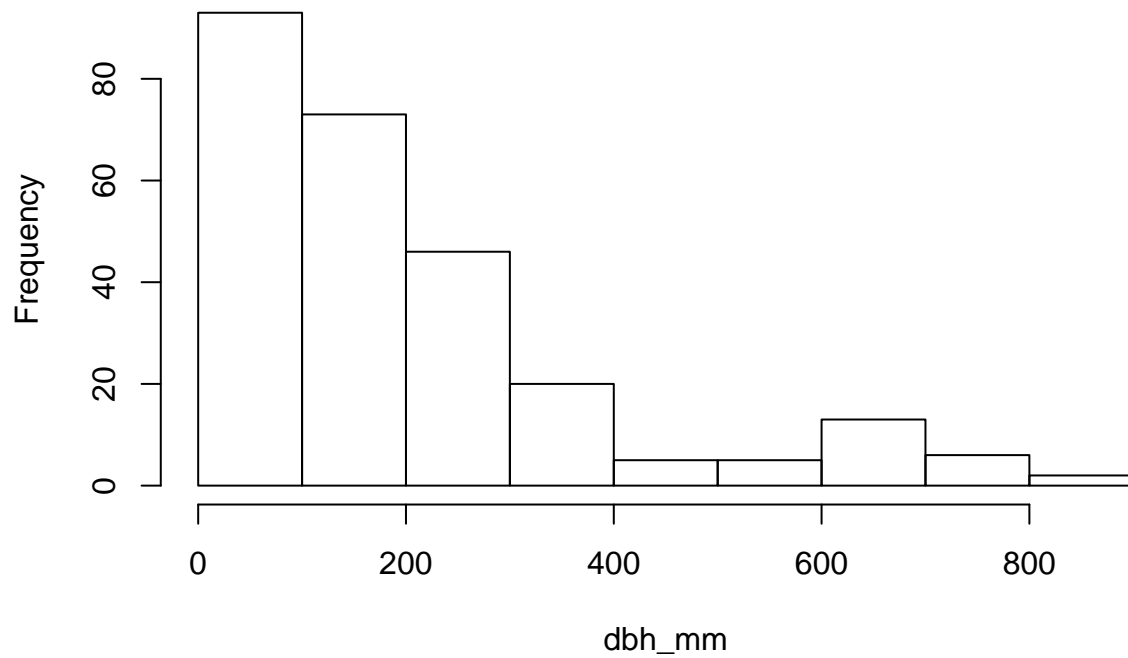
plot(h~dbh_mm,
     main="Diameter at breast height (dbh) to Height",
     xlab = "dbh (mm)",
     ylab = "Height (m)")
```

**Diameter at breast height (dbh) to Height**



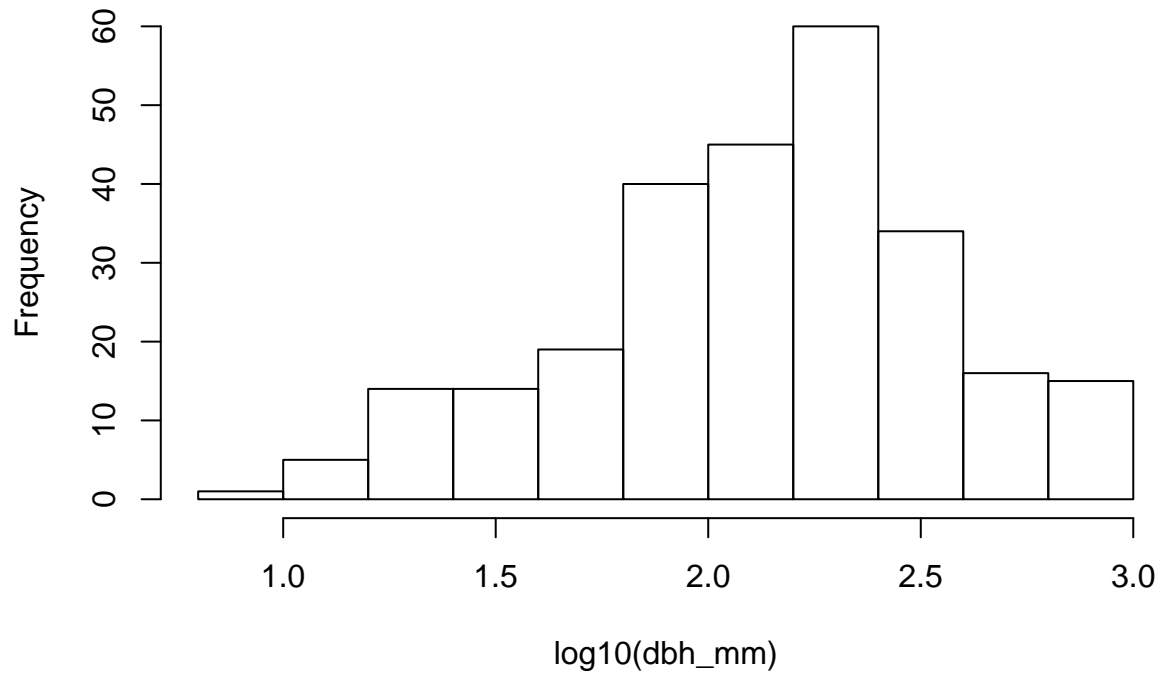
```
hist(dbh_mm)
```

**Histogram of dbh\_mm**



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

**Histogram of  $\log_{10}(\text{dbh\_mm})$**



## 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  $\log_{10}Height$  ( $m$ ) is regressed on  $\log_{10}DBH$  ( $mm$ ). 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.

$$Height = 10^{(\log_{10}(\min(d, dbh_{max})) * p1 + p2)}$$

```
# 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)
```

```

# 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

coef(lm(lmfit_mm~dbins_mm))

## (Intercept)      dbins_mm
##  3.45859976  0.04142493

coef(lm(lmfit_cm~dbins_cm))

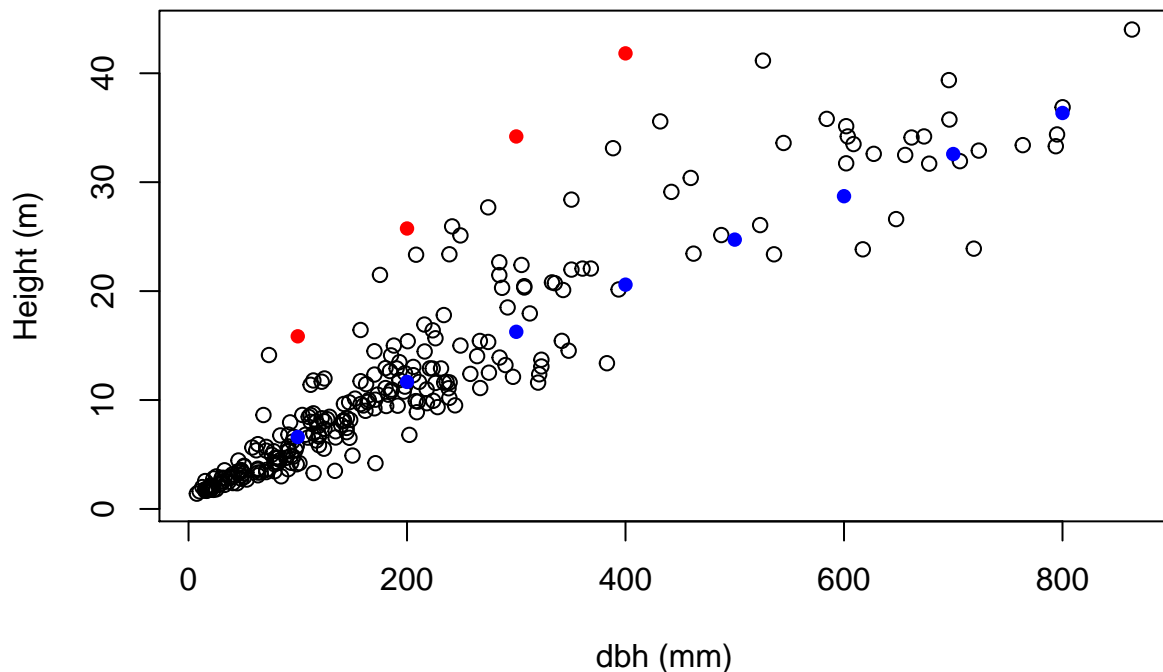
## (Intercept)      dbins_cm
##  3.4825736  0.4171207

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 D'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)

```

### Diameter at breast height (dbh) to Height

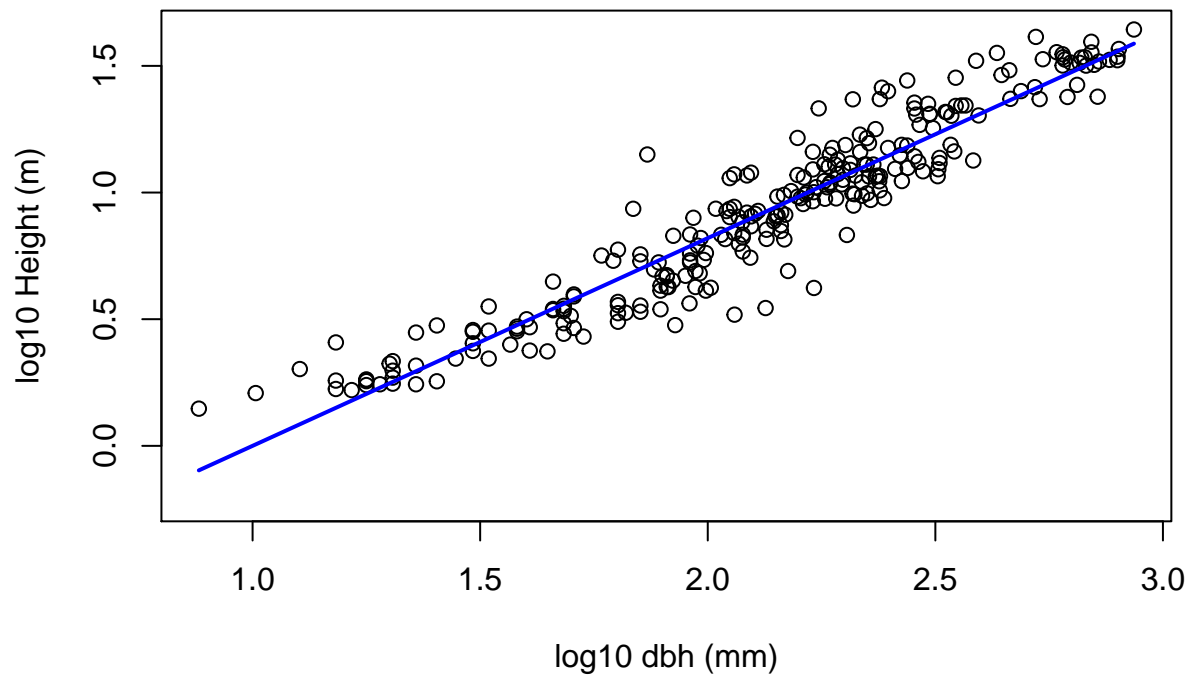


```

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

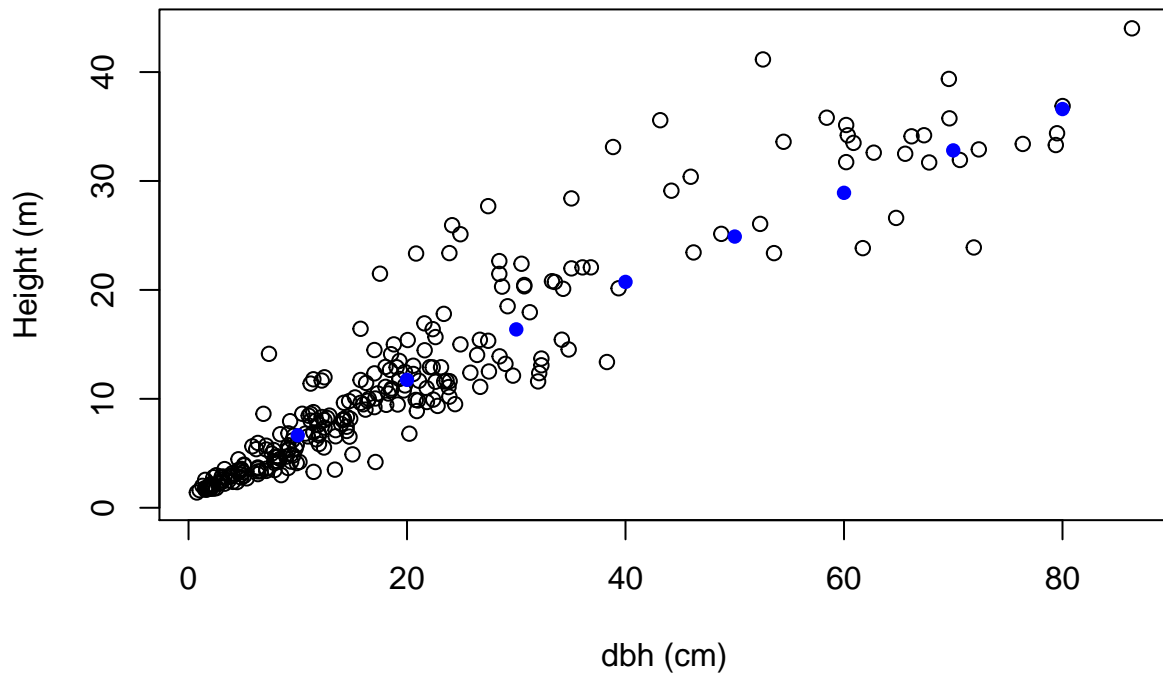
```

## Diameter at breast height (dbh) to Height



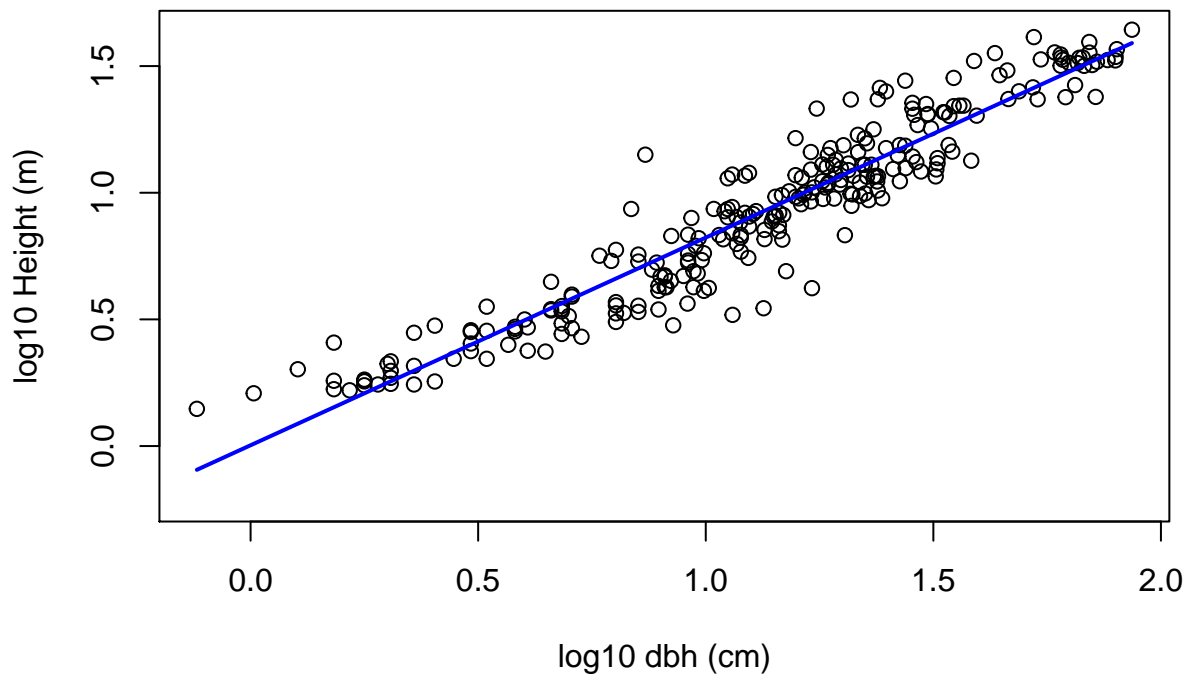
```
plot(h~dbh_cm,  
     main="Diameter at breast height (dbh) to Height",  
     xlab = "dbh (cm)",  
     ylab = "Height (m)")  
  
points(default_mm~dbins_mm, col="red", pch=16) # default O'Brien slope (0.7) and intercept (-0.2)  
points(lmfit_cm~dbins_cm, col="blue", pch=16) # BAAD data slope (0.82) and intercept (-0.82)
```

## Diameter at breast height (dbh) to Height



```
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 Height





## 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)

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

```
# 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, -0.82)
choj.h <- c(21.2, 117.1, 318, 646.1, 1119.6, 1754.5, 2565.1, 3564.4, 4764.4, 6176.6)

# 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(choj_p1, d.agb, choj_p2, c2b)
test_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

test <- sal(f_agb, p1, h.agb, p2, d.agb, p3, rho, p4)
test2 <- sal(f_agb, p1, lmfit.h, p2, d.agb, p3, rho, p4)
```

```

#plot(choj.h~d.agb, type="l", col="blue")
plot(test~d.agb, col="red", type="l", ylab="agb", xlab="dbh (cm)")
points(test2~d.agb, col="yellow", pch=16)
points(test_2par~d.agb, col="pink", pch=16)
points(test_choj~d.agb, col="green", pch=16)

```

