

# Pondo\_Allom\_Paramsv02

*Katie Murenbeeld*

*4/6/2020*

## 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
```

```
# Check the data
#head(d_baad)
```

Refine (subset) the data for Pinus Ponderosa:

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

```
# Check the data
#head(pipo)
```

## 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.

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

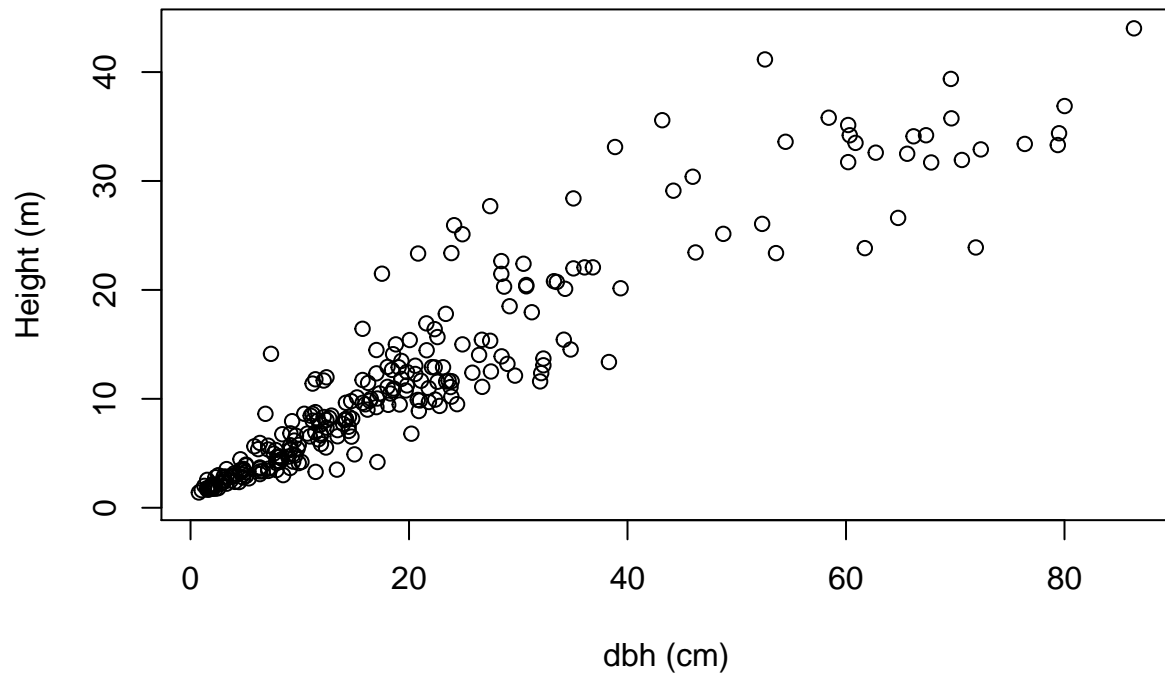
```
# Set variables from the data.

# Diameter at breast height (dbh) will be used for most allometric calculations
dbh_cm <- (pipo$d.bh)*100 # FATES needs dbh in cm

# Height
h <- (pipo$h.t)

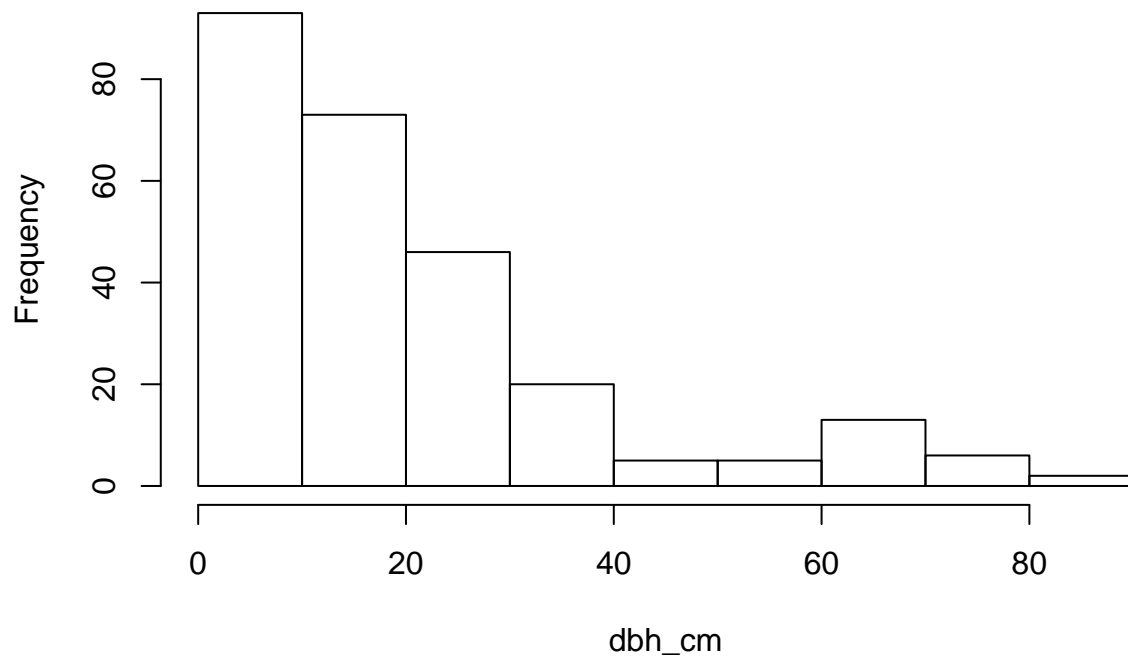
# Plot the raw data
plot(h~dbh_cm,
     main="Diameter at breast height (dbh) to Height",
     xlab = "dbh (cm)",
     ylab = "Height (m)")
```

## Diameter at breast height (dbh) to Height



```
# Plot a histogram  
hist(dbh_cm)
```

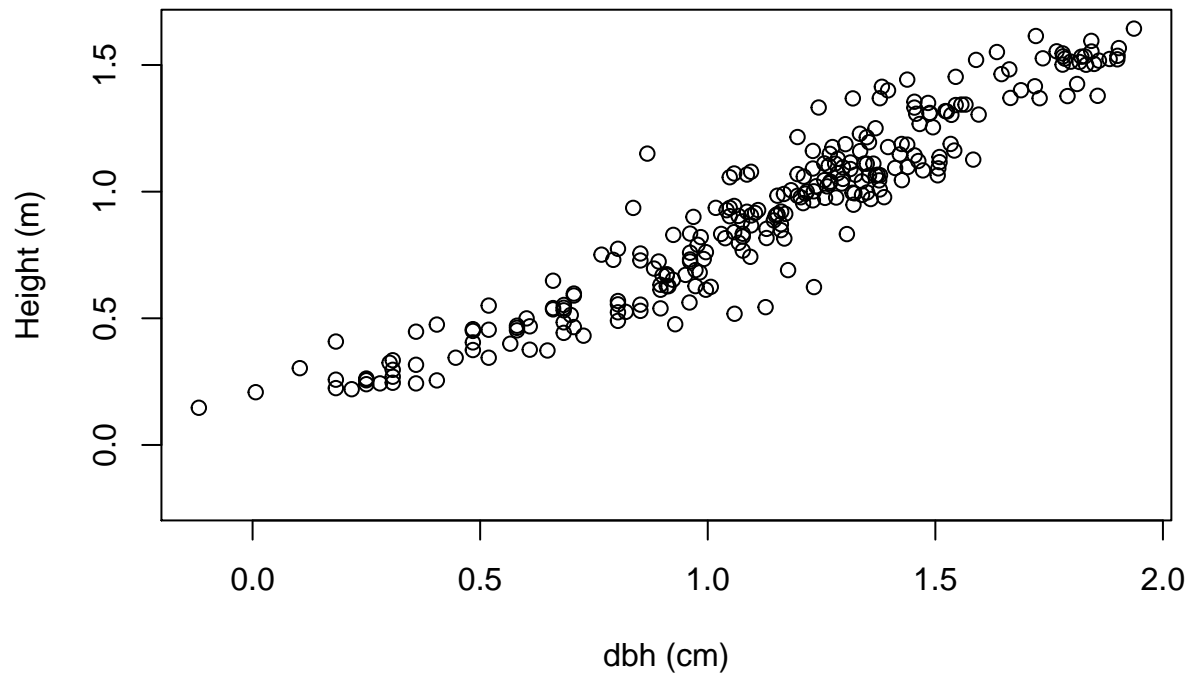
## Histogram of dbh\_cm



```
# Plot the log transformed data  
plot(log10(h)~log10(dbh_cm),
```

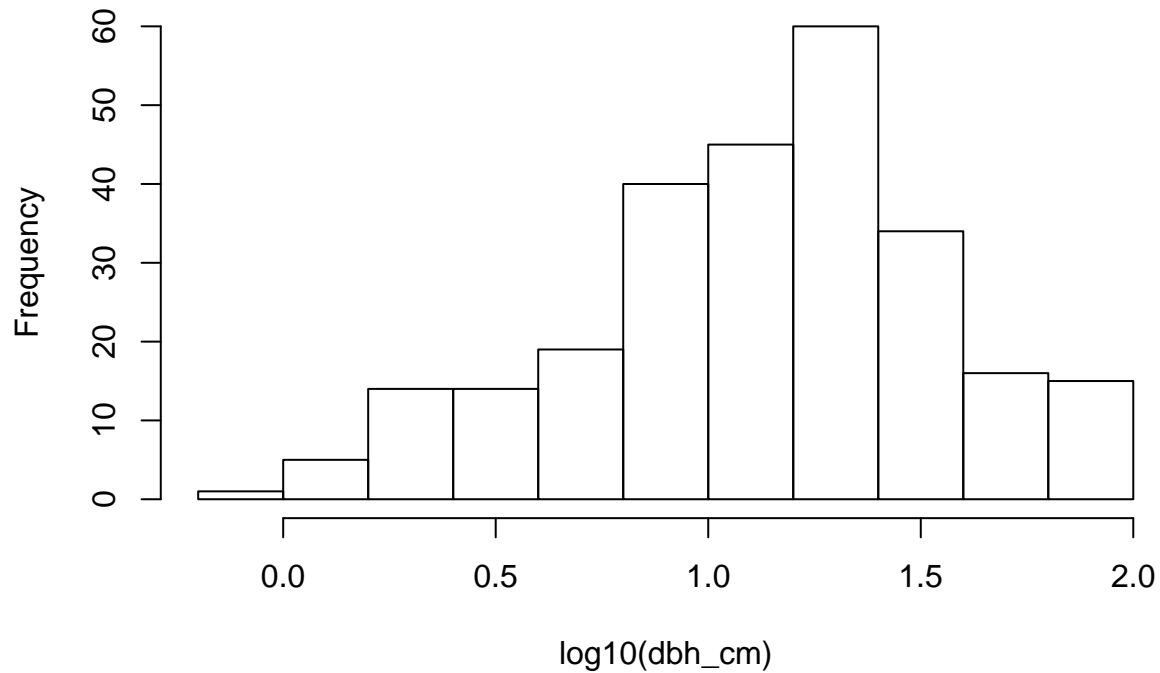
```
main="Diameter at breast height (dbh) to Height",  
xlab = "dbh (cm)",  
ylab = "Height (m)")
```

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



```
# Plot a histogram of the log transformed data  
hist(log10(dbh_cm))
```

## Histogram of log10(dbh\_cm)



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)
}

# Use default parameters from O'Brien et al 1995
default_cm <- obrien(dbh_cm, 0.7, -0.2)

# Test a linear model with log10(h) regressed on log10(dbh)
d2hmodlog_cm <- lm(log10(h)~log10(dbh_cm))
coef(d2hmodlog_cm)

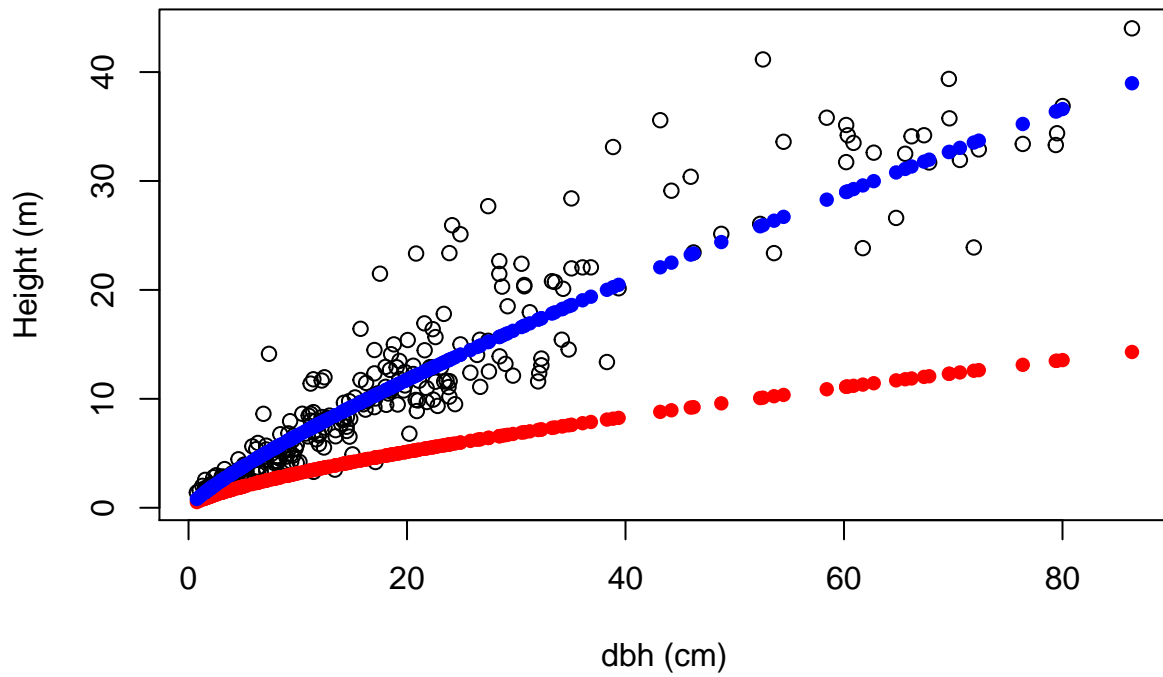
##      (Intercept) log10(dbh_cm)
##      0.003029536      0.818112721

# Use those coefficients as new parameters in the O'Brien function
lmfit_cm <- obrien(dbh_cm, 0.82, 0.003) # dbh in cm

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

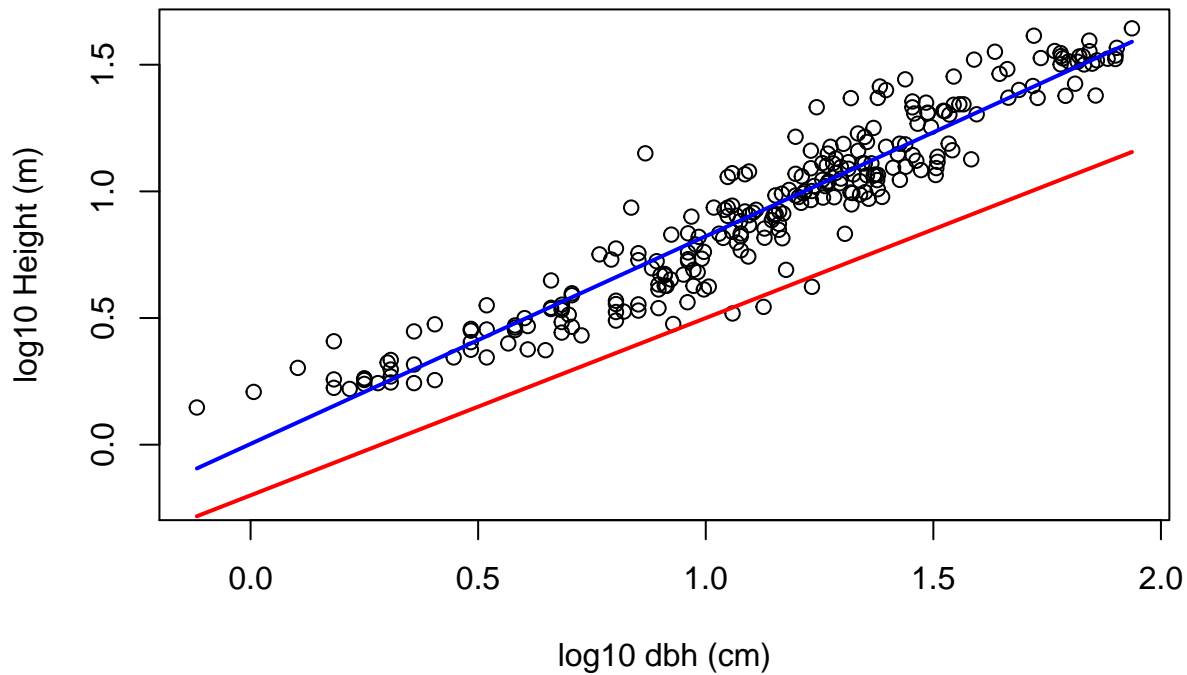
# default O'Brien slope (0.7) and intercept (-0.2)
points(default_cm~dbh_cm, col="red", pch=16)
# BAAD data slope (0.82) and intercept (-0.82) dbh in mm
points(lmfit_cm~dbh_cm, col="blue", pch=16)
```

## 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
curve(-0.2 + (0.7*x), add=T, col="red", lwd=2) # default int and slope from O'Brien
```

## Diameter at breast height (dbh) to Height



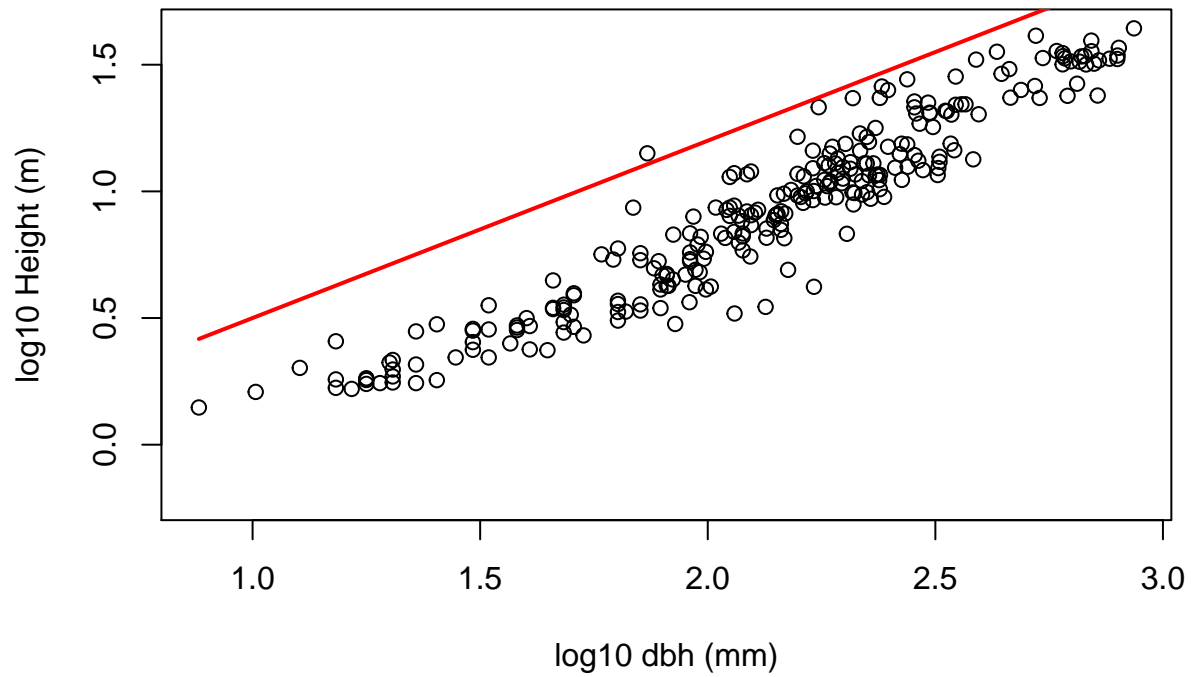
From a visual inspection, one can see that the default intercept and slope parameters from O'Brien et al. 1995 underestimate height in ponderosa pine compared to the observational data. This is likely due to the fact that O'Brien uses dbh in mm. However, even with dbh in mm, the fit of the O'Brien parameters to the BAAD ponderosa pine dataset is not very good.

```
# O'Brien uses dbh in mm

dbh_mm <- (pipo$d.bh)*1000

plot(log10(h)~log10(dbh_mm),
     main="Diameter at breast height (dbh) to Height, O'Brien Parameters",
     xlab = "log10 dbh (mm)",
     ylab = "log10 Height (m)")
curve(-0.2 + (0.7*x), add=T, col="red", lwd=2) # log regression coefficients from d2hmodlog
```

## Diameter at breast height (dbh) to Height, O'Brien Parameters



```
# Set up a function to calculate  $r^2$ 
r2 <- function(y_hat,y){
  RSS<-sum((((y_hat))-(y))^2)
  TSS<-sum(((y)-(mean(y)))^2)
  return(1-RSS/TSS)}

d2h_defaulttR2 <- r2(default_cm, h)
d2h_lmfitR2 <- r2(lmfit_cm, h)
```



## 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 test Saldarriaga et al. (1998) as well as the 2 parameter power function.

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

$$(C_{agb} = p_1 / c2b * d^{p_2})$$

There is no AGB data for ponderosa pines within BAAD. Therefore, AGB data calculated from Chojnacky et al. (year) will be used as proxy for observed data.

*# In this case dbh and agb calculated from Chojnacky serves as  
# the "observed data" to fit the other parameters against.*

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

```
# Define the parameters needed for the Saldarriaga function  
rho = 0.367 # wood density  
f_agb = 0.6 # fraction of abg  
d.agb <- c(10, 20, 30, 40, 50, 60, 70, 80, 90, 100) # set of dbh in cm  
h.agb <- c(2.82, 7.7, 13.9, 21.0, 29.1, 37.9, 37.9, 37.9, 37.9, 57.5)  
c2b = 2 # carbon to biomass  
# Use heights from d2h  
lmfit.h <- obrien(d.agb, 0.82, 0.003)
```

```
# Calculate the "observed" data from Chojnacky et al. (yea)  
choj <- choj_eq(b0, b1, d.agb)
```

```
# 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 Chojnacky if those param values are used)  
par2_pwr <- function(p1,d,p2,c2b){  
  bagw <- (p1*(d^p2))/c2b  
  return(bagw)
```

```

}

test_2par <- par2_pwr(0.146, d.agb, 2.464, c2b)

# Parameters from earlier Jupyter Notebook.
#p1 = 0.131
#p2 = 0.626
#p3 = 2.46
#p4 = 2.18

# New parameters from best fit below. (sal1 and sal2)

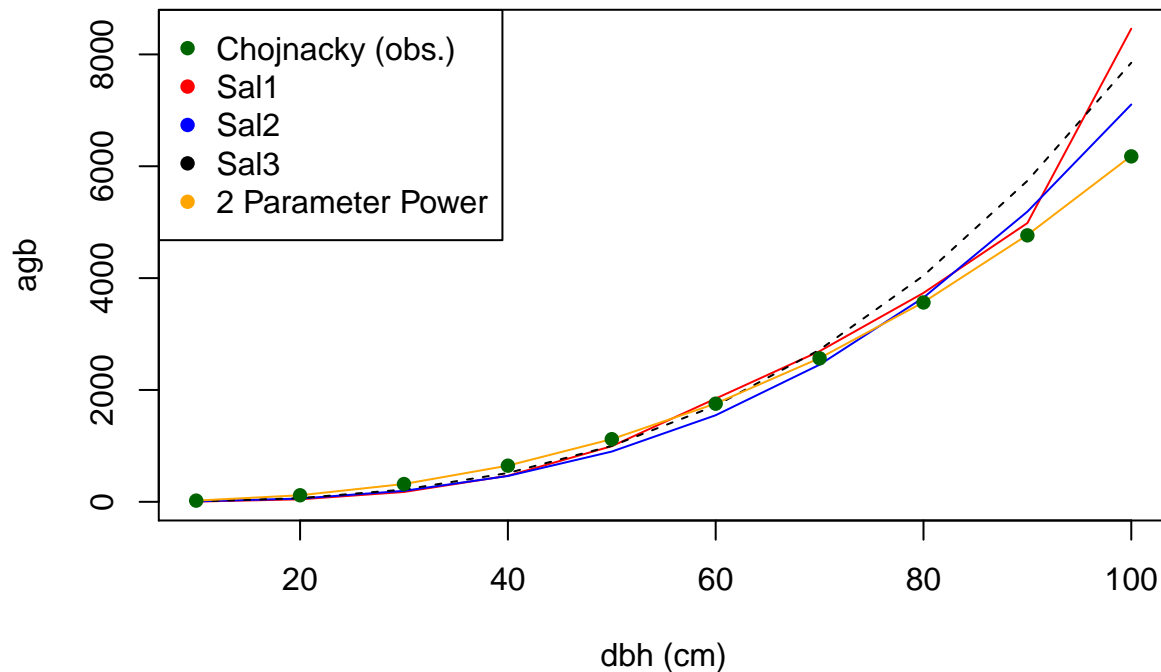
p1 = 0.11
p2 = 0.65
p3 = 2.45
p4 = 2.15

sal1 <- sal(f_agb, p1, h.agb, p2, d.agb, p3, rho, p4)
# sal2 uses the heights from the d2h calculation
# which is the work flow in FATES allometry calculations.
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 the Chojnacky AGB ("observed") against the different parameter tests

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)
legend("topleft",
      legend = c("Chojnacky (obs.)", "Sal1", "Sal2", "Sal3", "2 Parameter Power"),
      col = c("darkgreen", "red", "blue", "black", "orange"),
      pch = 16)

```



*# Here I want to get the r2 to compare the different allom functions and params to the "observed" chojn*

```
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_R2 <- r2(sal1, exp(choj))
```

```
sal2_R2 <- r2(sal2,exp(choj))
```

```
sal3_R2 <- r2(sal3, exp(choj))
```

```
test_2par_R2 <- r2(test_2par, exp(choj))
```

*# Create a table for easy comparison*

```
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_r2, row.names = agb_names)
```

```
agb_mod_fits
```

```
##
## Sal test 1      agb_r2
## Sal test 1      0.8675944
## Sal test 2      0.9703919
## Sal test 3      0.9004400
```

## Two Param Power 0.9999965