

## Online Resource 1. R code to develop age-to-DBH models from raw input data

```
# set working directory - update file path in the next line
setwd("C:/")

# load necessary packages - install with install.packages() command if needed
library(dvMisc)
library(MuMIn)
library(DescTools)

# import data table as .csv
# update CSV file name in the line of code beginning 'rawdata <-'
# must have the following columns:
# spp - species code
# age - tree age in years
# dbh - tree diameter (cm or inches)
# sources - data source for tree age (e.g., records, core, stump)
rawdata <- read.csv("input_data.csv", header=TRUE, sep = ",")

# generate a species list
spp_list <- unique(rawdata$spp)

# create a data frame to house the output
outputs <- data.frame(matrix(ncol=10, nrow=0))
colheads <- c("spp", "model", "a", "b", "c", "d", "e", "sigma", "AdjR2", "AICc")
colnames(outputs) <- colheads

# set j to 0. j will be used to assign row numbers sequentially in the output
j <- 0

# LINEAR: loops through each species, computes the linear model, and records the output

for (i in spp_list) {
  j <- j+1
  records <- rawdata[ which(rawdata$spp==i),]
  # linear model
  lin_mod <- lm(records$dbh ~ records$age)
  a <- as.numeric(lin_mod$coef[1])
  b <- as.numeric(lin_mod$coef[2])
  c <- c("-")
  d <- c("-")
  e <- c("-")
  sigma_mod <- sigma(lin_mod)
  AICc_mod <- AICc(lin_mod)
  adjr2_mod <- summary(lin_mod)$adj.r.squared
  model <- "linear"
  model_outputs <- c(i, model, a, b, c, d, e, sigma_mod, adjr2_mod, AICc_mod)
  outputs[j,] <- model_outputs
}
```

```
# QUADRATIC: loops through each species, computes the quadratic model, and records the output after the linear model
```

```
for (i in spp_list) {  
  j <- j+1  
  records <- rawdata[ which(rawdata$spp==i),]  
  # quadratic model  
  age2 <- records$age^2  
  quad_mod <- lm(records$dbh ~ records$age + age2)  
  a <- as.numeric(quad_mod$coef[1])  
  b <- as.numeric(quad_mod$coef[2])  
  c <- as.numeric(quad_mod$coef[3])  
  d <- c("-")  
  e <- c("-")  
  sigma_mod <- sigma(quad_mod)  
  AICc_mod <- AICc(quad_mod)  
  adjr2_mod <- summary(quad_mod)$adj.r.squared  
  model <- "quadratic"  
  model_outputs <- c(i, model, a, b, c, d, e, sigma_mod, adjr2_mod, AICc_mod)  
  outputs[j,] <- model_outputs  }
```

```
# CUBIC: loops through each species, computes the cubic model, and records the output after the quadratic model
```

```
for (i in spp_list) {  
  j <- j+1  
  records <- rawdata[ which(rawdata$spp==i),]  
  # cubic model  
  age2 <- records$age^2  
  age3 <- records$age^3  
  cub_mod <- lm(records$dbh ~ records$age + age2 + age3)  
  a <- as.numeric(cub_mod$coef[1])  
  b <- as.numeric(cub_mod$coef[2])  
  c <- as.numeric(cub_mod$coef[3])  
  d <- as.numeric(cub_mod$coef[4])  
  e <- c("-")  
  sigma_mod <- sigma(cub_mod)  
  AICc_mod <- AICc(cub_mod)  
  adjr2_mod <- summary(cub_mod)$adj.r.squared  
  model <- "cubic"  
  model_outputs <- c(i, model, a, b, c, d, e, sigma_mod, adjr2_mod, AICc_mod)  
  outputs[j,] <- model_outputs  }
```

```
# QUARTIC: loops through each species, computes the quartic model, and records the output after the cubic model
```

```
for (i in spp_list) {  
  j <- j+1  
  records <- rawdata[ which(rawdata$spp==i),]  
  # quartic model  
  age2 <- records$age^2
```

```

age3 <- records$age^3
age4 <- records$age^4
quar_mod <- lm(records$dbh ~ records$age + age2 + age3 + age4)
a <- as.numeric(quar_mod$coef[1])
b <- as.numeric(quar_mod$coef[2])
c <- as.numeric(quar_mod$coef[3])
d <- as.numeric(quar_mod$coef[4])
e <- as.numeric(quar_mod$coef[5])
sigma_mod <- sigma(quar_mod)
AICc_mod <- AICc(quar_mod)
adjr2_mod <- summary(quar_mod)$adj.r.squared
model <- "quartic"
model_outputs <- c(i, model, a, b, c, d, e, sigma_mod, adjr2_mod, AICc_mod)
outputs[j,] <- model_outputs  }

```

# EXPONENTIAL: loops through each species, computes the exponential model, and records the output after the quartic model

```

for (i in spp_list) {
  j <- j+1
  records <- rawdata[ which(rawdata$spp==i),]
  # exponential model
  exp_mod <- lm(log(records$dbh) ~ records$age + 1)
  a <- as.numeric(exp_mod$coef[1])
  b <- as.numeric(exp_mod$coef[2])
  c <- get_mse(exp_mod)
  d <- c("-")
  e <- c("-")
  adjr2_mod <- summary(exp_mod)$adj.r.squared
  # need to modify exp model to compare AICc values to the other models; see pages 23-24 in McPherson
  et al. 2016
  geom_mean <- Gmean(records$dbh, c("classic"))
  dbh_log <- log(records$dbh)
  dbh_gmean_log <- dbh_log * geom_mean
  exp_mod_aic <- lm(dbh_gmean_log ~ records$age)
  AICc_mod <- AICc(exp_mod_aic)
  sigma_mod <- sigma(exp_mod_aic)
  model <- "exponential"
  model_outputs <- c(i, model, a, b, c, d, e, sigma_mod, adjr2_mod, AICc_mod)
  outputs[j,] <- model_outputs  }

```

# LOG-LOG: loops through each species, computes the log-log model, and records the output after the exponential model

```

for (i in spp_list) {
  j <- j+1
  records <- rawdata[ which(rawdata$spp==i),]
  # log-log model
  log_mod <- lm(log(records$dbh) ~ log(records$age))
  a <- as.numeric(log_mod$coef[1])
  b <- as.numeric(log_mod$coef[2])

```

```

c <- get_mse(log_mod)
d <- c("-")
e <- c("-")
adjr2_mod <- summary(log_mod)$adj.r.squared
# need to modify log-log model to compare AICc values to the other models; see pages 23-24 in
McPherson et al. 2016
geom_mean <- Gmean(records$dbh, c("classic"))
dbh_log <- log(records$dbh)
age_log <- log(records$age)
dbh_gmean_log <- dbh_log * geom_mean
log_mod_aic <- lm(dbh_gmean_log ~ age_log)
AICc_mod <- AICc(log_mod_aic)
sigma_mod <- sigma(log_mod_aic)
model <- "log-log"
model_outputs <- c(i, model, a, b, c, d, e, sigma_mod, adjr2_mod, AICc_mod)
outputs[j,] <- model_outputs  }

# sort by species
outputs2 <- outputs[order(outputs$spp),]

# writes output to .csv table
write.csv(outputs2, file = "age_to_DBH_output.csv")

```