Supporting Online Material for
Height-Diameter Modeling of *Cinnamomum tamala* Grown in Natural Forest in Mid-hills of Nepal

Mukti Ram Subedi[*1], Bishwa Nath Oli[2], and Surendra Shrestha[3]

[1*] Forest Resources Management Program, Division of Forestry and Natural Resources, West Virginia University PO Box 6125, Morgantown, WV 26506-6125

Contact (cellular):- +13612282801, Email: mrs0070@mix.wvu.edu, suvedimukti@gmail.com

[2] Ministry of Population and Environment, Kathmandu, Nepal

Email: bn_oli@yahoo.com

[3]Graduate School of Geography, Clark University, 950 Main St, Worcester, MA 01610

Email: SurShrestha@clarku.edu, sbs.stha111@gmail.com

Supporting online materials includes

Table SA-SB

Fig. SA-SB

Table SA: Models selected for model fitting and validation, model designation, models selected for further evaluation (in text Model #) and parameters to be estimated.

Model	Designation	In text (model #)	Parameter	Reference
$H_i = 1.3 + b_0 D^{b_1} + \varepsilon_i$	M1	M1	2	(Huxley and Tesissier 1936)
$H_i = 1.3 + b_0 \left[1 - exp^{(-b_1 D^{b2})}\right] + \varepsilon_i$	M2	M2	3	(Zeide 1993)
$H_i = 1.3 + b_0 \left[1 - exp^{(-b_1 D)}\right]^{b_2} + \varepsilon_i$	M3	M3	3	(Richards 1959)
$H_i = 1.3 + \frac{D}{(b_0 + b_1 D)} + \varepsilon_i$	M4	M4	2	(Hosoda and Iehara 2010)
$H_i = 1.3 + \exp(b_0 D^{b_1}) + \varepsilon_i$	M5	M5	2	(Sharma 2011)
$H_i = 1.3 + b_0 \exp\left(\frac{-b_1}{D}\right) + \varepsilon_i$	M6	M6	2	(Schumacher 1939)
$H_i = 1.3 + \left[\frac{D}{(1 + b_0 D)}\right]^{b_1} + \varepsilon_i$	M7	M7	2	(Curtis 1967; Huang et al. 2000)
$H_i = 1.3 + \frac{b_0 D}{(1 + D) + (b_1 D)} + \varepsilon_i$	M8	M8	2	(Larson 1986)
$H_i = 1.3 + \left[\frac{D}{(b_0 + b_1 D)}\right]^3 + \varepsilon_i$	M9	M9	2	(Näslund 1936) cited in (Sharma et al. 2016)
$H_i = 1.3 + \frac{D^2}{(b_0 + b_1 D)^2} + \varepsilon_i$	M10		2	(Loetsch et al. 1973)
$H_i = 1.3 + \frac{b_0}{\left(1 + \frac{1}{(b_1 D^{b_2})}\right)} + \varepsilon_i$	M11	M10	3	(Ratkowsky and Giles 1990)
$H_i = 1.3 + \frac{D^2}{(b_0 + b_1 D)^2} + \varepsilon_i$	M12	M11	2	(Loetsch et al. 1973)
$H_i = 1.3 + \frac{b_0 D}{(b_1 + D)} + \varepsilon_i$	M13	M12	2	(Ratkowsky and Reedy 1986; Larson 1986)
$H_i = 1.3 + b_0 D^{b1 D^{b_2}}$	M14		3	(Huang et al. 2000)
$H_i = 1.3 + \frac{b_0}{1 + b_1\, exp(D^{b_2})} + \varepsilon_i$	M15	M13	3	(Pearl and Reed 1920)
$H_i = 1.3 + b_0 Exp\left(\frac{b_1}{D + b_2}\right) + \varepsilon_i$	M16		3	(Ratkowsky and Reedy 1986)
$H_i = 1.3 + b_0 \exp(-b_1 \times exp^{-b_2 D}) + \varepsilon_i$	M17	M14	3	(Winsor 1932)
$H_i = 1.3 + b_0 \exp(-b_1 \times D^2) + \varepsilon_i$	M18	M15	2	(Huang et al. 2000)

H_i –total height (m) of tree i=1, 2, 3,....,n, D-diameter at breast height (cm), b_0,b_1, and b_2 are parameter to be estimated, 1.3 is added avoid prediction of zero height when D approaches zero, εi is a random error term which is assumed to be normally and identically distributed with mean 0 and variance σ^2 [NID (0, σ^2)]

Table SB: Parameter estimates and fit statistics of model M4 and M12 for validation and combined dataset

Dataset	Model	Parameter	Coefficient	Std. error	t	p-value	Bias	RMSE	MAE
Validation data	M4	b_0	1.441	0.186	7.739	<0.001			
		b_1	0.034	0.010	3.306	0.002	0.115	1.129	1.494
	M12	b_0	29.035	8.782	3.306	0.002			
		b_1	41.845	17.723	2.361	0.024	0.115	1.129	1.494
Combined data	M4	b_0	1.242	0.088	14.086	<0.001			
		b_1	0.045	0.005	8.996	<0.001	0.033	1.336	1.716
	M12	b_0	22.362	2.486	8.996	<0.001			
		b_1	27.766	4.953	5.606	<0.001	0.033	1.336	1.716

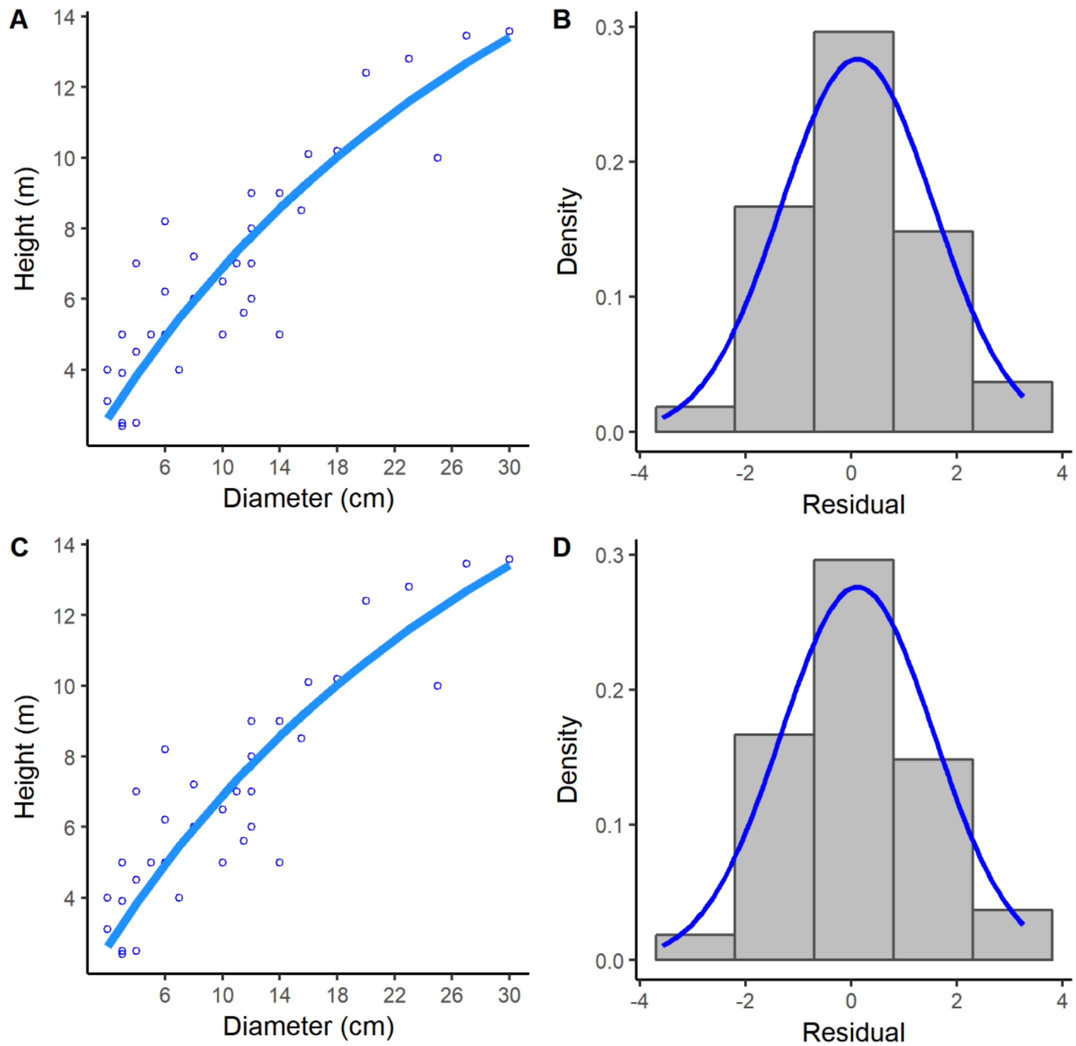

Figure SA.: Scatter plot of total heights of validation dataset against D superimposed with Model M4 (A) and model M12 (C), a histogram of individual residuals with the normal curve for model M4 (B) and M12 (D)

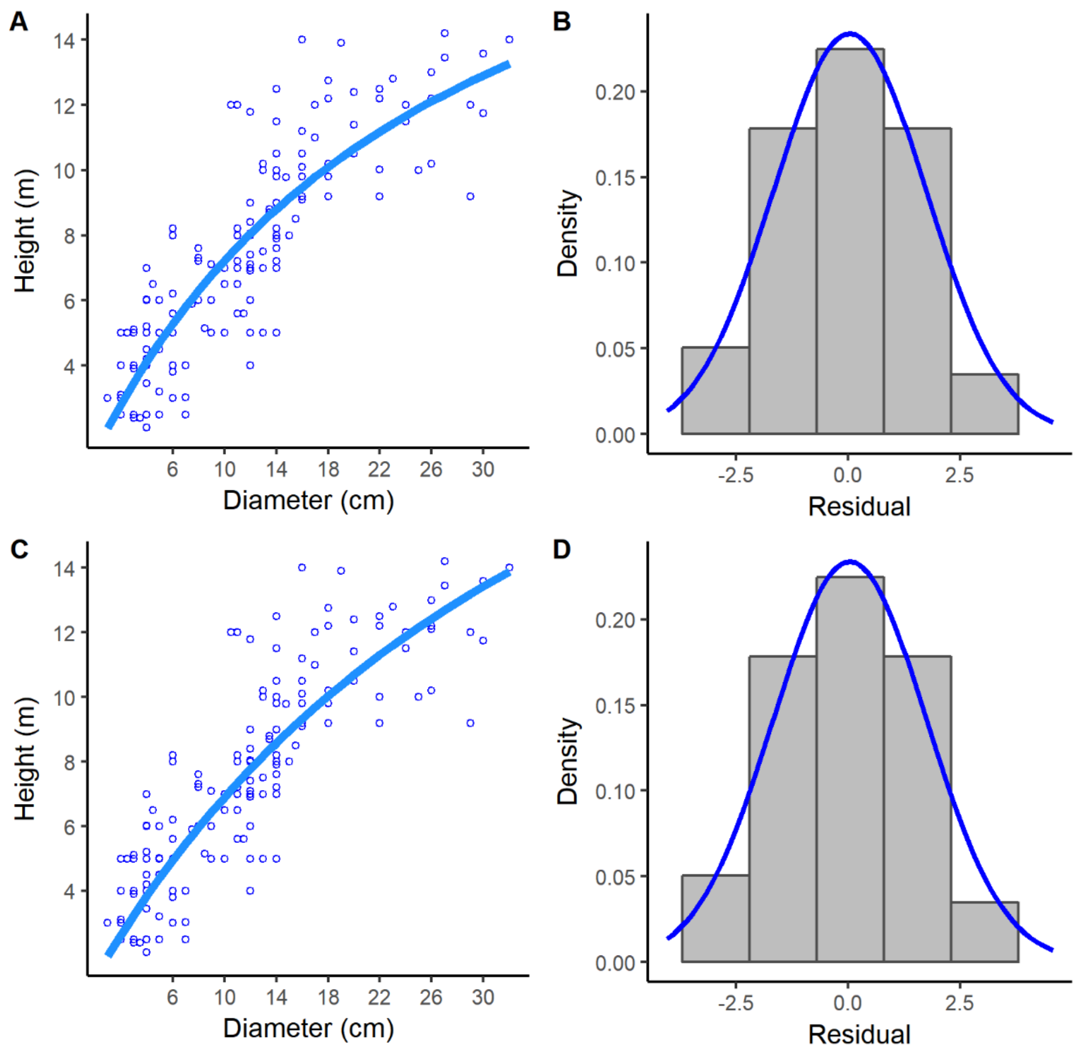

Figure SB.: Scatter plot of total heights from combined dataset against D superimposed with Model M4 (A) and model M12 (C), a histogram of individual residuals with the normal curve for model M4 (B) and M12 (D)

References:

Curtis RO (1967) Height-diameter and height-diameter-age equations for second-growth douglas-fir. For Sci 13:365–375.

Hosoda K, Iehara T (2010) Aboveground biomass equations for individual trees of *Cryptomeria japonica*, *Chamaecyparis obtusa* and *Larix kaempferi* in Japan. J For Res 15:299–306. doi: 10.1007/s10310-010-0192-y

Huang S, Price D, Titus SJ (2000) Development of ecoregion-based height - diameter models for white spruce in boreal forests. For Ecol Manage 129:125–141.

Huxley JS, Tesissier G (1936) Terminology of relative growth. Nature 137:780–781. doi: 10.1038/148225a0

Larson BC (1986) Development and growth of even-aged stands of Douglas-fir and grand fir. Can J For Res 16:367–372. doi: 10.1139/x86-063

Lei X, Peng C, Wang H, Zhou X (2009) Individual height–diameter models for young black spruce (*Picea mariana*) and jack pine (*Pinus banksiana*) plantations in New Brunswick, Canada. For Chron 85:43–56. doi: 10.5558/tfc85043-1

Loetsch F, Zöhrer F, Haller KE (1973) Forest Inventory. BLV Verlagsgesellschaft, Munich

Näslund M (1936) Skogsforsö ksastaltens gallringsforsök i tallskog. Medd från Statens Skogsförsöksanstalt 29:1–169.

Pearl R, Reed LJ (1920) On the rate of growth of the population of the United States since 1790 and its mathematical representation. Proc Natl Acad Sci 6:275. doi: 10.1373/clinchem.2009.133124

Ratkowsky DA, Giles DEA (1990) Handbook of nonlinear regression models. M. Dekker New York

Ratkowsky DA, Reedy TJ (1986) Choosing Near-Linear Parameters in the Four-Parameter Logistic Model for Radioligand and Related Assays. Biometrics 42:575. doi: 10.2307/2531207

Richards FJ (1959) A flexible growth function for empirical use. J Exp Bot 10:290–301. doi: 10.1093/jxb/10.2.290

Schumacher FX (1939) A new growth curve and its application to timber yield studies. J For 37:819–20.

Sharma RP (2011) Allometric models for total-tree and component-tree biomass of *Alnus* *nepalensis* D. Don in Nepal. Indian For 137:1386–90.

Sharma RP, Vacek Z, Vacek S (2016) Nonlinear mixed effect height-diameter model for mixed species forests in the central part of the Czech Republic. J For Sci 62:470–484. doi: 10.17221/41/2016-JFS

Winsor CP (1932) The gompertz curve as a growth curve. Proc Natl Acad Sci 18:1–8. doi: 10.1073/pnas.18.1.1

Zeide B (1993) Analysis of growth equations. For Sci 39:594–616. doi: 10.1111/j.1461-0248.2006.00883.x

```
#############  Height Diameter Model  #####################
#              Cinnamomum tamala                          #
#                                                         #
###########################################################

#### - Setting Global Options - ####
options(continue = "...",digits = 10,width = 80,timeout = 15,prompt = ">")

# Removing everything from environment
rm(list = ls())

####- Reading data - ####
# data cdat has three columns(sn,D,and ht)
cdat <- read.table (file = "combinedData.csv", header = TRUE, sep = ",")

####- Installing and loading library -####
# if foloowing packages are not installed, install them using following code
# package<- c("nls1","tidyverse","broom","gridExtra","ggpubr","cowplot")
# for(pack in package) {
# install.packages(pack)
# }

library(nls2)
library(tidyverse)
library(broom)
library(gridExtra)
library(ggpubr)
library(cowplot)

#### - Data Randomization -####
 #split the data into 2 groups
 Rand <- sample(1:dim(cdat),(dim(cdat)*8/10),replace = FALSE)
 sort(Rand)

 #select random trees
 cdat$Num <- 1:nrow(cdat)
 cdat$Selec <- match(cdat$Num,Rand)

 #trees to fit equation
 dat <- subset(cdat, cdat$Selec > 0,drop = TRUE)
 #trees to validate
 valdata <- subset(cdat, is.na(cdat$Selec))

# model building
# model building
nl1  <- nls2(ht~(1.3+ b0*D^b1),data = dat,
             start = list(b0 = 1,b1 = 0.1))
nl2  <- nls2(ht~(1.3 + b0*(1-(exp(-1*b1*D^b2)))),data = dat,
             start = list(b0 = 10.00,b1 = 0.001,b2 = 2.0))
nl3  <- nls2(ht~(1.3 + b0*(1-(exp(-1*b1*D)))^b2),data = dat,
             start = list(b0 = 14.0,b1 = 0.02,b2 = 1.0))
nl4  <- nls2(ht~(1.3 + D/(b0+b1*D)),data = dat,
             start = list(b0 = 10.00,b1 = 0.2))
nl5  <- nls2(ht~(1.3 + (exp(b0*D^b1))),data = dat,
             start = list(b0 = 10.00,b1 = 0.2))
nl6  <- nls2(ht~(1.3 + b0*(exp(-b1/D))),data = dat,
             start = list(b0 = 10.00,b1 = 0.2))
```

```
nl7 <- nls2(ht~(1.3 + b0*(D/(1+D))^b1),data = dat,
            start = list(b0 = 10.00,b1 = 0.2))
nl8 <- nls2(ht~(1.3 + (b0*D)/((1+D)+(b1*D))),data = dat,
            start = list(b0 = 10.00,b1 = 0.2))
nl9 <- nls2(ht~(1.3 + (D/(b0+b1*D))^3),data = dat,
            start = list(b0 = 10.00,b1 = 0.2))
nl10 <- nls2(ht~(1.3 + ((D^2)/(b0+b1*D+b2*D^2))),data = dat,
             start = list(b0 = 10.00,b1 = 0.2, b2 = 0.2))
nl11 <- nls2(ht~(1.3 + (b0/(1+(1/(b1*D^b2))))),data = dat,
             start = list(b0 = 10.00,b1 = 0.0002, b2 = 3))
nl12 <- nls2(ht~(1.3 + ((D^2)/(b0+b1*D)^2)),data = dat,
             start = list(b0 = 10.00,b1 = 0.2))
nl13 <- nls2(ht~(1.3 + (b0*D)/(b1+D)),data = dat,
             start = list(b0 = 10.00,b1 = 0.2))
nl14 <- nls2(ht~(1.3 + (b0*D^(b1*D^(-b2)))),data = dat,
             start = list(b0 = 11.00,b1 = 5.0, b2 = 0.2))
nl15 <- nls2(ht~(1.3 + (b0/(1+(b1*exp(-b2*D))))),data = dat,
             start = list(b0 = 11.6,b1 = 6.5, b2 = 0.20 ))
nl16 <- nls2(ht~(1.3+ (b0*exp(b1/(b2+D)))),data = dat,
             start = list(b0 = 6.99,b1 = -0.20, b2 = -2.29 ))
nl17 <- nls2(ht~(1.3+ (b0*exp(-b1*exp(-b2*D)))),data = dat,
             start = list(b0 = 21.00,b1 = 2.00, b2 = .029 ))
nl18 <- nls2(ht~(1.3+ (b0*exp(-b1*D^2))),data = dat,
             start = list(b0 = 6.00,b1 = -0.0002 ))

# Height estimate (H1 to H18)

H1<- function(D){
  1.3+coef(nl1)[[1]]*D^(coef(nl1)[[2]])
}
H2  <- function(D) {
  1.3 + coef(nl2)[[1]]*(1-(exp(-1*coef(nl2)[[2]]*D^coef(nl2)[[3]])))
}
H3  <- function(D) {
  1.3 + coef(nl3)[[1]]*(1-(exp(-1*coef(nl3)[[2]]*D)))^coef(nl3)[[3]]
}
H4  <- function(D) {
  1.3 + D/(coef(nl4)[[1]]+coef(nl4)[[2]]*D)
}
H5  <- function(D) {
  1.3 + (exp(coef(nl5)[[1]]*D^coef(nl5)[[2]]))
}
H6  <- function(D) {
  1.3 + coef(nl6)[[1]]*(exp(-coef(nl6)[[2]]/D))
}
H7 <- function(D) {
  1.3 + coef(nl7)[[1]]*(D/(1+D))^coef(nl7)[[2]]
}

H8 <- function(D) {
  1.3 + (coef(nl8)[[1]]*D)/((1+D)+((coef(nl8)[[2]])*D))
}
H9 <- function(D) {
  1.3 + (D/(coef(nl9)[[1]]+coef(nl9)[[2]]*D))^3
}
H10 <- function(D) {
  1.3 + (D^2/(coef(nl10)[[1]]+coef(nl10)[[2]]*D+coef(nl10)[[3]]*D^2))
```

```
}
H11 <- function(D) {
  1.3 + (coef(nl11)[[1]]/(1+(1/(coef(nl11)[[2]]*D^coef(nl11)[[3]]))))
}
H12 <- function(D) {
  1.3 + (D^2/(coef(nl12)[[1]]+coef(nl12)[[2]]*D)^2)
}
H13 <- function(D) {
  1.3 + (coef(nl13)[[1]]*D)/(coef(nl13)[[2]]+D)
}
H14 <- function(D) {
  1.3 + (coef(nl14)[[1]]*D^(coef(nl14)[[2]]*D^(-1*coef(nl14)[[3]])))
}
H15 <- function(D) {
  1.3 + (coef(nl15)[[1]]/(1+(coef(nl15)[[2]]*exp(D*-coef(nl15)[[3]]))))
}
H16 <- function(D) {
  1.3+ (coef(nl16)[[1]]*exp(coef(nl16)[[2]]/(coef(nl16)[[3]]+D)))
}
H17 <- function(D) {
  1.3+ (coef(nl17)[[1]]*exp(-coef(nl17)[[2]]*exp(-coef(nl17)[[3]]*D)))
}
H18 <- function(D) {
  1.3+ (coef(nl18)[[1]]*exp(-coef(nl18)[[2]]*D^2))
}

# place estimate height in dataset table
dat$nl1  <- mapply(H1, dat$D)
dat$nl2  <- mapply(H2, dat$D)
dat$nl3  <- mapply(H3, dat$D)
dat$nl4  <- mapply(H4, dat$D)
dat$nl5  <- mapply(H5, dat$D)
dat$nl6  <- mapply(H6, dat$D)
dat$nl7  <- mapply(H7, dat$D)
dat$nl8  <- mapply(H8, dat$D)
dat$nl9  <- mapply(H9, dat$D)
dat$nl10 <- mapply(H10,dat$D)
dat$nl11 <- mapply(H11,dat$D)
dat$nl12 <- mapply(H12,dat$D)
dat$nl13 <- mapply(H13,dat$D)
dat$nl14 <- mapply(H14,dat$D)
dat$nl15 <- mapply(H15,dat$D)
dat$nl16 <- mapply(H16,dat$D)
dat$nl17 <- mapply(H17,dat$D)
dat$nl18 <- mapply(H18,dat$D)

# Bias estimate
(BiasM1  <- sum(dat$ht - dat$nl1)/nrow(dat))
(BiasM2  <- sum(dat$ht - dat$nl2)/nrow(dat))
(BiasM3  <- sum(dat$ht - dat$nl3)/nrow(dat))
(BiasM4  <- sum(dat$ht - dat$nl4)/nrow(dat))
(BiasM5  <- sum(dat$ht - dat$nl5)/nrow(dat))
(BiasM6  <- sum(dat$ht - dat$nl6)/nrow(dat))
(BiasM7  <- sum(dat$ht - dat$nl7)/nrow(dat))
(BiasM8  <- sum(dat$ht - dat$nl8)/nrow(dat))
(BiasM9  <- sum(dat$ht - dat$nl9)/nrow(dat))
(BiasM10 <- sum(dat$ht - dat$nl10)/nrow(dat))
```

```
(BiasM11 <- sum(dat$ht - dat$nl11)/nrow(dat))
(BiasM12 <- sum(dat$ht - dat$nl12)/nrow(dat))
(BiasM13 <- sum(dat$ht - dat$nl13)/nrow(dat))
(BiasM14 <- sum(dat$ht - dat$nl14)/nrow(dat))
(BiasM15 <- sum(dat$ht - dat$nl15)/nrow(dat))
(BiasM16 <- sum(dat$ht - dat$nl16)/nrow(dat))
(BiasM17 <- sum(dat$ht - dat$nl17)/nrow(dat))
(BiasM18 <- sum(dat$ht - dat$nl18)/nrow(dat))
# create list of biases
Bias <- round(rbind(BiasM1,BiasM2,BiasM3,BiasM4,BiasM5,BiasM6,
                    BiasM7,BiasM8,BiasM9,BiasM10,BiasM11,
                    BiasM12,BiasM13,BiasM14,BiasM15,BiasM16,
                    BiasM17,BiasM18),digits = 3)
### Mean Absolute Error
(MAEM1   <- sum(abs(dat$ht - dat$nl1))/nrow(dat))
(MAEM2   <- sum(abs(dat$ht - dat$nl2))/nrow(dat))
(MAEM3   <- sum(abs(dat$ht - dat$nl3))/nrow(dat))
(MAEM4   <- sum(abs(dat$ht - dat$nl4))/nrow(dat))
(MAEM5   <- sum(abs(dat$ht - dat$nl5))/nrow(dat))
(MAEM6   <- sum(abs(dat$ht - dat$nl6))/nrow(dat))
(MAEM7   <- sum(abs(dat$ht - dat$nl7))/nrow(dat))
(MAEM8   <- sum(abs(dat$ht - dat$nl8))/nrow(dat))
(MAEM9   <- sum(abs(dat$ht - dat$nl9))/nrow(dat))
(MAEM10  <- sum(abs(dat$ht - dat$nl10))/nrow(dat))
(MAEM11  <- sum(abs(dat$ht - dat$nl11))/nrow(dat))
(MAEM12  <- sum(abs(dat$ht - dat$nl12))/nrow(dat))
(MAEM13  <- sum(abs(dat$ht - dat$nl13))/nrow(dat))
(MAEM14  <- sum(abs(dat$ht - dat$nl14))/nrow(dat))
(MAEM15  <- sum(abs(dat$ht - dat$nl15))/nrow(dat))
(MAEM16  <- sum(abs(dat$ht - dat$nl16))/nrow(dat))
(MAEM17  <- sum(abs(dat$ht - dat$nl17))/nrow(dat))
(MAEM18  <- sum(abs(dat$ht - dat$nl18))/nrow(dat))
# create list of MAEs
MAE <- round(rbind(MAEM1,MAEM2,MAEM3,MAEM4,MAEM5,
                   MAEM6,MAEM7,MAEM8,MAEM9,MAEM10,
                   MAEM11,MAEM12,MAEM13,MAEM14,MAEM15,
                   MAEM16,MAEM17,MAEM18),digits = 3)
#RMSE calculation
(RMSEM1  <- sqrt(sum((dat$nl1 -dat$ht)^2)/(nrow(dat)-length(summary(nl1
)$coef[,1])-1)))
(RMSEM2  <- sqrt(sum((dat$nl2 -dat$ht)^2)/(nrow(dat)-length(summary(nl2
)$coef[,1])-1)))
(RMSEM3  <- sqrt(sum((dat$nl3 -dat$ht)^2)/(nrow(dat)-length(summary(nl3
)$coef[,1])-1)))
(RMSEM4  <- sqrt(sum((dat$nl4 -dat$ht)^2)/(nrow(dat)-length(summary(nl4
)$coef[,1])-1)))
(RMSEM5  <- sqrt(sum((dat$nl5 -dat$ht)^2)/(nrow(dat)-length(summary(nl5
)$coef[,1])-1)))
(RMSEM6  <- sqrt(sum((dat$nl6 -dat$ht)^2)/(nrow(dat)-length(summary(nl6
)$coef[,1])-1)))
(RMSEM7  <- sqrt(sum((dat$nl7 -dat$ht)^2)/(nrow(dat)-length(summary(nl7
)$coef[,1])-1)))
(RMSEM8  <- sqrt(sum((dat$nl8 -dat$ht)^2)/(nrow(dat)-length(summary(nl8
)$coef[,1])-1)))
(RMSEM9  <- sqrt(sum((dat$nl9 -dat$ht)^2)/(nrow(dat)-length(summary(nl9
)$coef[,1])-1)))
```

```
(RMSEM10 <- sqrt(sum((dat$nl10 -dat$ht)^2)/(nrow(dat)-length(summary(nl10
)$coef[,1])-1)))
(RMSEM11 <- sqrt(sum((dat$nl11 -dat$ht)^2)/(nrow(dat)-length(summary(nl11
)$coef[,1])-1)))
(RMSEM12 <- sqrt(sum((dat$nl12 -dat$ht)^2)/(nrow(dat)-length(summary(nl12
)$coef[,1])-1)))
(RMSEM13 <- sqrt(sum((dat$nl13 -dat$ht)^2)/(nrow(dat)-length(summary(nl13
)$coef[,1])-1)))
(RMSEM14 <- sqrt(sum((dat$nl14 -dat$ht)^2)/(nrow(dat)-length(summary(nl14
)$coef[,1])-1)))
(RMSEM15 <- sqrt(sum((dat$nl15 -dat$ht)^2)/(nrow(dat)-length(summary(nl15
)$coef[,1])-1)))
(RMSEM16 <- sqrt(sum((dat$nl16 -dat$ht)^2)/(nrow(dat)-length(summary(nl16
)$coef[,1])-1)))
(RMSEM17 <- sqrt(sum((dat$nl17 -dat$ht)^2)/(nrow(dat)-length(summary(nl17
)$coef[,1])-1)))
(RMSEM18 <- sqrt(sum((dat$nl18 -dat$ht)^2)/(nrow(dat)-length(summary(nl18
)$coef[,1])-1)))
# list of RMSEs
RMSE <- round(rbind(RMSEM1,RMSEM2,RMSEM3,RMSEM4,RMSEM5,RMSEM6,
                    RMSEM7,RMSEM8,RMSEM9,RMSEM10,RMSEM11,
                    RMSEM12,RMSEM13,RMSEM14,RMSEM15,RMSEM16,
                    RMSEM17,RMSEM18),digits = 3)
# calculate AIC
(AICM1   <-  AIC(nl1))
(AICM2   <-  AIC(nl2))
(AICM3   <-  AIC(nl3))
(AICM4   <-  AIC(nl4))
(AICM5   <-  AIC(nl5))
(AICM6   <-  AIC(nl6))
(AICM7   <-  AIC(nl7))
(AICM8   <-  AIC(nl8))
(AICM9   <-  AIC(nl9))
(AICM10  <-  AIC(nl10))
(AICM11  <-  AIC(nl11))
(AICM12  <-  AIC(nl12))
(AICM13  <-  AIC(nl13))
(AICM14  <-  AIC(nl14))
(AICM15  <-  AIC(nl15))
(AICM16  <-  AIC(nl16))
(AICM17  <-  AIC(nl17))
(AICM18  <-  AIC(nl18))

# list of AICs
AIC <- round(rbind(AICM1,AICM2,AICM3,AICM4,AICM5,AICM6,AICM7,
                   AICM8,AICM9,AICM10,AICM11,AICM12,AICM13,
                   AICM14,AICM15,AICM16,AICM17,AICM18),digits = 3)
# calculate BIC
(BICM1   <-  BIC(nl1))
(BICM2   <-  BIC(nl2))
(BICM3   <-  BIC(nl3))
(BICM4   <-  BIC(nl4))
(BICM5   <-  BIC(nl5))
(BICM6   <-  BIC(nl6))
(BICM7   <-  BIC(nl7))
(BICM8   <-  BIC(nl8))
(BICM9   <-  BIC(nl9))
```

```
(BICM10  <-  BIC(nl10))
(BICM11  <-  BIC(nl11))
(BICM12  <-  BIC(nl12))
(BICM13  <-  BIC(nl13))
(BICM14  <-  BIC(nl14))
(BICM15  <-  BIC(nl15))
(BICM16  <-  BIC(nl16))
(BICM17  <-  BIC(nl17))
(BICM18  <-  BIC(nl18))

# list of BICs
BIC <- round(rbind(BICM1,BICM2,BICM3,BICM4,BICM5,BICM6,
                   BICM7,BICM8,BICM9,BICM10,BICM11,
                   BICM12,BICM13,BICM14,BICM15,BICM16,
                   BICM17,BICM18),digits = 3)

# combine fit statistics

fitstat<-cbind(Bias,RMSE,MAE,AIC,BIC)

# convert fitstat to dataframe
fitstat<-data.frame(fitstat)

# column name
colfitstat<-c("Bias","RMSE","MAE","AIC","BIC")
names(fitstat)<-colfitstat

 # validation dataset estimate
valdata$vnl1  <- mapply(H1,valdata$D)
valdata$vnl2  <- mapply(H2,valdata$D)
valdata$vnl3  <- mapply(H3,valdata$D)
valdata$vnl4  <- mapply(H4,valdata$D)
valdata$vnl5  <- mapply(H5,valdata$D)
valdata$vnl6  <- mapply(H6,valdata$D)
valdata$vnl7  <- mapply(H7,valdata$D)
valdata$vnl8  <- mapply(H8,valdata$D)
valdata$vnl9  <- mapply(H9,valdata$D)
valdata$vnl10 <- mapply(H10,valdata$D)
valdata$vnl11 <- mapply(H11,valdata$D)
valdata$vnl12 <- mapply(H12,valdata$D)
valdata$vnl13 <- mapply(H13,valdata$D)
valdata$vnl14 <- mapply(H14,valdata$D)
valdata$vnl15 <- mapply(H15,valdata$D)
valdata$vnl16 <- mapply(H16,valdata$D)
valdata$vnl17 <- mapply(H17,valdata$D)
valdata$vnl18 <- mapply(H18,valdata$D)

# Validation dataset BIAS
# RBias estimate
(RBiasM1 <- sum(valdata$ht - valdata$vnl1)/nrow(valdata))
(RBiasM2 <- sum(valdata$ht - valdata$vnl2)/nrow(valdata))
(RBiasM3 <- sum(valdata$ht - valdata$vnl3)/nrow(valdata))
(RBiasM4 <- sum(valdata$ht - valdata$vnl4)/nrow(valdata))
(RBiasM5 <- sum(valdata$ht - valdata$vnl5)/nrow(valdata))
(RBiasM6 <- sum(valdata$ht - valdata$vnl6)/nrow(valdata))
(RBiasM7 <- sum(valdata$ht - valdata$vnl7)/nrow(valdata))
(RBiasM8 <- sum(valdata$ht - valdata$vnl8)/nrow(valdata))
```

```
(RBiasM9 <- sum(valdata$ht - valdata$vnl9)/nrow(valdata))
(RBiasM10 <- sum(valdata$ht - valdata$vnl10)/nrow(valdata))
(RBiasM11 <- sum(valdata$ht - valdata$vnl11)/nrow(valdata))
(RBiasM12 <- sum(valdata$ht - valdata$vnl12)/nrow(valdata))
(RBiasM13 <- sum(valdata$ht - valdata$vnl13)/nrow(valdata))
(RBiasM14 <- sum(valdata$ht - valdata$vnl14)/nrow(valdata))
(RBiasM15 <- sum(valdata$ht - valdata$vnl15)/nrow(valdata))
(RBiasM16 <- sum(valdata$ht - valdata$vnl16)/nrow(valdata))
(RBiasM17 <- sum(valdata$ht - valdata$vnl17)/nrow(valdata))
(RBiasM18 <- sum(valdata$ht - valdata$vnl18)/nrow(valdata))

# list of RBias
RBias <- round(rbind(RBiasM1, RBiasM2,RBiasM3,RBiasM4,RBiasM5,RBiasM6,
                     RBiasM7,RBiasM8,RBiasM9,RBiasM10,RBiasM11,
                     RBiasM12,RBiasM13,RBiasM14,RBiasM15,RBiasM16,
                     RBiasM17,RBiasM18),digits = 3)

# Mean absolute Error
(RMAEM1   <- sum(abs(valdata$ht - valdata$vnl1))/nrow(valdata))
(RMAEM2   <- sum(abs(valdata$ht - valdata$vnl2))/nrow(valdata))
(RMAEM3   <- sum(abs(valdata$ht - valdata$vnl3))/nrow(valdata))
(RMAEM4   <- sum(abs(valdata$ht - valdata$vnl4))/nrow(valdata))
(RMAEM5   <- sum(abs(valdata$ht - valdata$vnl5))/nrow(valdata))
(RMAEM6   <- sum(abs(valdata$ht - valdata$vnl6))/nrow(valdata))
(RMAEM7   <- sum(abs(valdata$ht - valdata$vnl7))/nrow(valdata))
(RMAEM8   <- sum(abs(valdata$ht - valdata$vnl8))/nrow(valdata))
(RMAEM9   <- sum(abs(valdata$ht - valdata$vnl9))/nrow(valdata))
(RMAEM10  <- sum(abs(valdata$ht - valdata$vnl10))/nrow(valdata))
(RMAEM11  <- sum(abs(valdata$ht - valdata$vnl11))/nrow(valdata))
(RMAEM12  <- sum(abs(valdata$ht - valdata$vnl12))/nrow(valdata))
(RMAEM13  <- sum(abs(valdata$ht - valdata$vnl13))/nrow(valdata))
(RMAEM14  <- sum(abs(valdata$ht - valdata$vnl14))/nrow(valdata))
(RMAEM15  <- sum(abs(valdata$ht - valdata$vnl15))/nrow(valdata))
(RMAEM16  <- sum(abs(valdata$ht - valdata$vnl16))/nrow(valdata))
(RMAEM17  <- sum(abs(valdata$ht - valdata$vnl17))/nrow(valdata))
(RMAEM18  <- sum(abs(valdata$ht - valdata$vnl18))/nrow(valdata))

# list of RMAEs
RMAE <- round(rbind(RMAEM1,RMAEM2,RMAEM3,RMAEM4,RMAEM5,RMAEM6,
                    RMAEM7,RMAEM8,RMAEM9,RMAEM10,RMAEM11,
                    RMAEM12,RMAEM13,RMAEM14,RMAEM15,RMAEM16,
                    RMAEM17,RMAEM18),digits = 3)
### VRMSE
(VRMSEM1  <- sqrt(sum((valdata$vnl1 - valdata$ht)^2)/(nrow(valdata)-
length(summary(nl1 )$coef[,1])-1)))
(VRMSEM2  <- sqrt(sum((valdata$vnl2 - valdata$ht)^2)/(nrow(valdata)-
length(summary(nl2 )$coef[,1])-1)))
(VRMSEM3  <- sqrt(sum((valdata$vnl3 - valdata$ht)^2)/(nrow(valdata)-
length(summary(nl3 )$coef[,1])-1)))
(VRMSEM4  <- sqrt(sum((valdata$vnl4 - valdata$ht)^2)/(nrow(valdata)-
length(summary(nl4 )$coef[,1])-1)))
(VRMSEM5  <- sqrt(sum((valdata$vnl5 - valdata$ht)^2)/(nrow(valdata)-
length(summary(nl5 )$coef[,1])-1)))
(VRMSEM6  <- sqrt(sum((valdata$vnl6 - valdata$ht)^2)/(nrow(valdata)-
length(summary(nl6 )$coef[,1])-1)))
(VRMSEM7  <- sqrt(sum((valdata$vnl7 - valdata$ht)^2)/(nrow(valdata)-
length(summary(nl7 )$coef[,1])-1)))
```

```
(VRMSEM8  <- sqrt(sum((valdata$vnl8 - valdata$ht)^2)/(nrow(valdata)-
length(summary(nl8 )$coef[,1])-1)))
(VRMSEM9  <- sqrt(sum((valdata$vnl9 - valdata$ht)^2)/(nrow(valdata)-
length(summary(nl9 )$coef[,1])-1)))
(VRMSEM10 <- sqrt(sum((valdata$vnl10 - valdata$ht)^2)/(nrow(valdata)-
length(summary(nl10 )$coef[,1])-1)))
(VRMSEM11 <- sqrt(sum((valdata$vnl11 - valdata$ht)^2)/(nrow(valdata)-
length(summary(nl11 )$coef[,1])-1)))
(VRMSEM12 <- sqrt(sum((valdata$vnl12 - valdata$ht)^2)/(nrow(valdata)-
length(summary(nl12 )$coef[,1])-1)))
(VRMSEM13 <- sqrt(sum((valdata$vnl13 - valdata$ht)^2)/(nrow(valdata)-
length(summary(nl13 )$coef[,1])-1)))
(VRMSEM14 <- sqrt(sum((valdata$vnl14 - valdata$ht)^2)/(nrow(valdata)-
length(summary(nl14 )$coef[,1])-1)))
(VRMSEM15 <- sqrt(sum((valdata$vnl15 - valdata$ht)^2)/(nrow(valdata)-
length(summary(nl15 )$coef[,1])-1)))
(VRMSEM16 <- sqrt(sum((valdata$vnl16 - valdata$ht)^2)/(nrow(valdata)-
length(summary(nl16 )$coef[,1])-1)))
(VRMSEM17 <- sqrt(sum((valdata$vnl17 - valdata$ht)^2)/(nrow(valdata)-
length(summary(nl17 )$coef[,1])-1)))
(VRMSEM18 <- sqrt(sum((valdata$vnl18 - valdata$ht)^2)/(nrow(valdata)-
length(summary(nl18 )$coef[,1])-1)))

# list of VRMSE
VRMSE <- round(rbind(VRMSEM1,VRMSEM2,VRMSEM3,VRMSEM4,VRMSEM5,VRMSEM6,VRMSEM7,
                     VRMSEM8,VRMSEM9,VRMSEM10,VRMSEM11,VRMSEM12,VRMSEM13,
                     VRMSEM14,VRMSEM15,VRMSEM16,VRMSEM17,VRMSEM18),digits = 3)
# combing fitstat

Vfitstat<-cbind(RBias,VRMSE,RMAE)

# convert fitstat to dataframe
Vfitstat<-data.frame(Vfitstat)

# column name
colfitstat1<-c("Bias","RMSE","MAE")
names(Vfitstat)<-colfitstat1

##coefficient estimates (Table 3)
coeff<-rbind(
  summary(nl1)$param,
  summary(nl2)$param,
  summary(nl3)$param,
  summary(nl4)$param,
  summary(nl5)$param,
  summary(nl6)$param,
  summary(nl7)$param,
  summary(nl8)$param,
  summary(nl9)$param,
  summary(nl10)$param,
  summary(nl11)$param,
  summary(nl12)$param,
  summary(nl13)$param,
  summary(nl14)$param,
  summary(nl15)$param,
  summary(nl16)$param,
  summary(nl17)$param,
```

```
  summary(nl18)$param
)

# Table 4
models<- paste("Model ",seq(1:18))
Table4<- cbind(models,fitstat[,-1],Vfitstat)

## sub-set data for residual analysis
rdat<- dat[,2:4]
rdat$Model01  <- summary(nl1)$resid
rdat$Model02  <- summary(nl2)$resid
rdat$Model03  <- summary(nl3)$resid
rdat$Model04  <- summary(nl4)$resid
rdat$Model05  <- summary(nl5)$resid
rdat$Model06  <- summary(nl6)$resid
rdat$Model07  <- summary(nl7)$resid
rdat$Model08  <- summary(nl8)$resid
rdat$Model09  <- summary(nl9)$resid
rdat$Model10  <- summary(nl10)$resid
rdat$Model11  <- summary(nl11)$resid
rdat$Model12  <- summary(nl12)$resid
rdat$Model13  <- summary(nl13)$resid
rdat$Model14  <- summary(nl14)$resid
rdat$Model15  <- summary(nl15)$resid
rdat$Model16  <- summary(nl16)$resid
rdat$Model17  <- summary(nl17)$resid
rdat$Model18  <- summary(nl18)$resid

# convert wide to long # need tidyr
residuals <- rdat %>%
  gather(Resid, Value, -sn,-D, -ht)
residuals$Resid <- as.factor(residuals$Resid)
# create label for model model 10, 14, and 16 are discarded hence received same
label
labels <- c(Model01 = "Model 1",
            Model02 = "Model 2",  Model03 = "Model 3",
            Model04 = "Model 4",  Model05 = "Model 5",
            Model06 = "Model 6",  Model07 = "Model 7",
            Model08 = "Model 8",  Model09 = "Model 9",
            Model10 = "Model 10", Model11 = "Model 10",
            Model12 = "Model 11", Model13 = "Model 12",
            Model14 = "Model 13", Model15 = "Model 13",
            Model16 = "Model 14", Model17 = "Model 14",
            Model18 = "Model 15"
            )

## plot residuals avoiding insignificant models: Figure 1
RvsD <- ggplot(data = filter(residuals,Resid != "Model10",
                      Resid != "Model14",Resid != "Model16"))+
    geom_point(mapping = aes(x = D, y = Value),shape = 21,color = "black",
             fill = "gray60", size = 1)+
    geom_hline(yintercept = 0, color = "Red", size = 0.8)+
    ylab("Residuals")+
    xlab ("Diameter (cm)")+
    scale_y_continuous(breaks = c(-5,0,5))+
    theme_bw(10)+
    theme(
```

```
    panel.background = element_rect( fill = "white" , color = "white"),
    panel.grid = element_blank(),
    panel.grid.major = element_line(colour = "white", linetype = "dotted"),
    panel.grid.minor = element_line(colour = "white", linetype = "dotted"),
    strip.text = element_text(size = 8),
    strip.background = element_rect(colour = "black", fill = "#CCCCFF", size =
0.2),
    axis.title =  element_text(size = rel(1.05) , color = "black"),
    axis.text =   element_text(size = rel(1.05) , color = "black"),
    axis.text.x = element_text(size =  8, angle = 360, face = "plain"),
    axis.text.y = element_text(size = 8,face = "plain",angle = 360),
    axis.title.y = element_text(size = 10, face = "plain"))+
  facet_wrap(~Resid, nrow = 3,scales = "fixed",labeller = labeller(Resid =
labels))

# Convert (dat) data to Longitudinal Format as DatGat
DatGat <- dat %>%
  gather(nl1:nl18,key = "Model",value = "value")
#select best model
seldata <- DatGat %>%
  filter (Model %in% c("nl17"))

seldata$Model[seldata$Model == "nl17"] <- "M14"

# plotting them
selmodel <- ggplot(data = seldata,aes(x = D))+
  geom_point(mapping = aes(y = ht),size = 1.0,shape = 21,color = "blue")+
  geom_line(mapping = aes(y = value),color = '#1E90FF',size = 1.5,
            show.legend = FALSE)+
  scale_y_continuous(breaks = c(4,6,8,10,12,14))+
  scale_x_continuous(breaks = c(6,10,14,18,22,26,30))+
  xlab("Diameter (cm)")+
  ylab ("Height (m)")+
  theme_classic(10)

# residuals
reshisto <- ggplot(data = residuals %>% filter(Resid == "Model14"),aes(x =
Value))+
  geom_histogram(mapping = aes(y = ..density..),
                 breaks = (seq(-3.798224,4.947046, by = 1.749054)),
                 colour = "gray30",
                 fill = "gray")+
  stat_function(fun = dnorm, args = list(mean = mean(residuals$Value),
                                         sd = sd(residuals$Value)),
                color = "blue",size = 1)+
  theme_classic(10)+
  ylab("Density")+
  xlab ("Residual")

# validation dataset Analysis
vdata01 <- valdata
vDatGat <- vdata01 %>%
  gather(vnl1:vnl18,key = "Model",value = "value")

######### Residual distribution
vResData <- vDatGat %>%
  filter (Model %in% c("R17"))
```

```
vResData$Model[vResData$Model == "R17"] <- "M 14"
#
## making graph
vseldata <- vDatGat %>%
  filter (Model %in% c("vnl17"))

  vseldata$Model[vseldata$Model == "vnl17"]<- "M 14"

  # plotting M14
vselmodel <- ggplot(data = vseldata,aes(x = D))+
  geom_point(mapping = aes(y = ht),size = 1.5,shape = 21,color = "blue")+
  geom_line(mapping = aes(y = value),color = '#1E90FF',size = 1.5,
            show.legend = FALSE)+
  scale_y_continuous(breaks = c(4,6,8,10,12,14))+
  scale_x_continuous(breaks = c(6,10,14,18,22,26,30))+
  xlab("Diameter (cm)")+
  ylab ("Height (m)")+
  theme_classic(10)

# residual histogram
vseldata1 <- valdata %>% mutate (R14 = vnl7-ht)

  vreshisto <- ggplot(data = vseldata1,aes(x = R14))+
  geom_histogram(mapping = aes(y = ..density..),
                 breaks = (seq(-3.70,4.947046, by = 1.5)),
                 colour = "gray30",
                 fill = "gray")+
  stat_function(fun = dnorm, args = list(mean = mean(vseldata1$R14),
                                         sd = sd(vseldata1$R14)),
                color = "blue",size = 1)+
  theme_classic(10)+
  ylab("Density")+
  xlab ("Residual")
# Figure 2
gtp <- arrangeGrob(selmodel,
                 reshisto,vselmodel, vreshisto,
                 ncol = 2, nrow = 2 )
vp <- as_ggplot(gtp) +  # transform to a ggplot
  draw_plot_label(label = c("A", "C", "D","B"), size = 10,
                x = c(0, 0, 0.52,0.52), y = c(1, 0.52, 0.52,1))
vp

                ############ End ###################

                           contact:
                         suvedimukti@gmail.com
```