The following report addresses four questions from the exercise protocol and discusses the methods used. The aims of this exercise were to develop regression models for predicting AGB of the Clocaenog forest site and select the appropriate model candidate based on their internal error rate. Section 1 provides a summary of data and methods used to develop eight candidate regression models. The following section evaluates the candidate models using accuracy assessments. Important to note that integrating remote sensing data into a multi-phase forest inventory sampling requires a statistical framework that is based on unbiased model-assisted estimators (20–22). Therefore, efforts are made to check all four type (I-IV) errors and to evaluate if models are over-fitted to the data based on k-fold validation techniques.

1. Building a LiDAR model for predicting above ground biomass (AGB)

Models were calibrated between field plot data and remote sensing statistics, which were linked using global navigation satellite systems (23). Field data was collected from among 50 randomly sampled concentric circular plots (10m radius), including measurements of tree diameters at breast height (dbh, cm). Remote sensing LiDAR data was generated from an airborne laser scanning survey (ALS), from which three predictor variables were generated: mean tree height (MeanH) (11,24–29), standard deviation (SD), and fractional canopy height (Cover) based on ratio of height returns above 1.3m to total returns (30–35). Descriptive statistics of these predictor variables are presented below in Table 1.

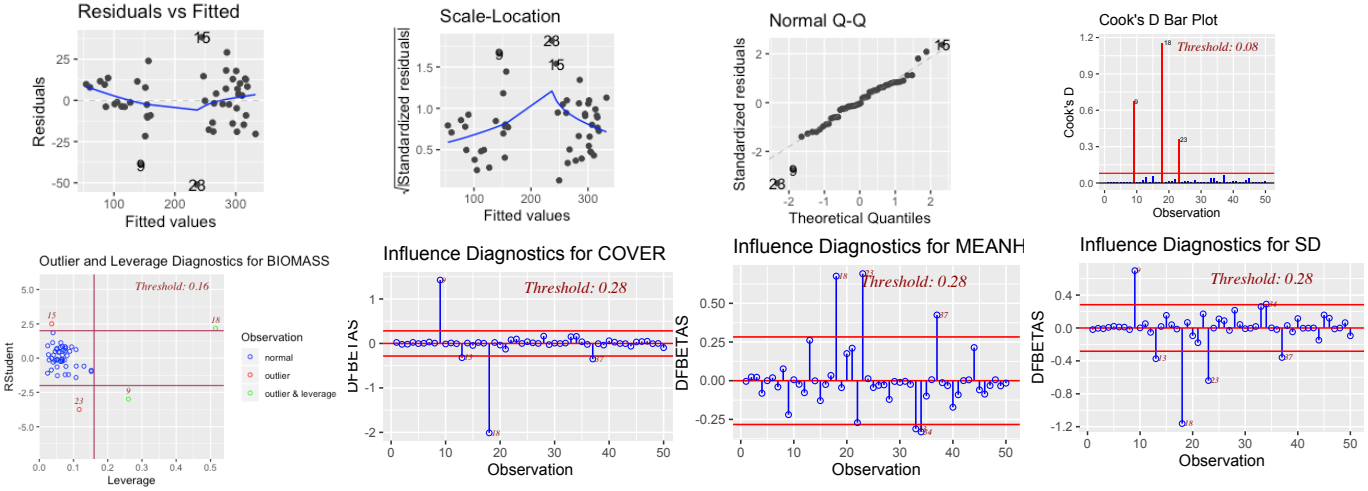
Table 1. Descriptives of predictors: Standard Deviation (SD), Standard Error (SE), Shapiro-Wilk (SW) score of normality.

Variable	Mean	SD	SE	Skewness	Kurtosis	SW
MeanH	13.05	5.77	0.82	-0.32	-1.40	***
Cover	0.92	0.08	0.01	-1.76	4.03	***
SD	5.41	2.84	0.40	-0.15	-1.47	***

A simple linear regression was conducted (M1) and its residuals were explored using regression diagnostics. These results are presented below in Table 2 and Figures 1 to 8. Results indicated signs of heteroscedasticity, multi-collinearity and influential outliers. Heteroscedasticity was observed by the Breusch-Pagan test ($p = 0.002$), and confirmed by the "Residuals vs Fitted" and "Scale Location". Although no collinearity problems were detected by Durba-Watson test, the Variance-Inflation-Factor identified Standard Deviation as a problematic predictor. The existence of influential outliers was observed using the Bonferroni test ($p = 0.025$). To explore measures of influence, analyses ran a Cook's Distance graph, a DFBETA panel, and a "Studentized Residuals vs Leverage" plot using the *lmtest* and *olsrr* packages in R. Two observations (obs 9 &18) were removed and a new dataset was developed "Clocaenog_Cleaned". To compare models, both datasets were used, and models corresponding to these are hereafter labeled "-clean" and "-clean".

Table 2. Hypothesis tests of linear regression: Shapiro-Wilk (SW) test of normality of residuals, Breusch-Pagan (BP) test of heteroskedasticity of residuals, Durbin-Watson (DW) test to search for multi-collinearity of data, Variance Inflation Factor (VIF) test to understand collinearity problems, Bonferroni (BF) test to search for outliers.

Model	SW	BP	DW	BF	Variance Inflation Factor		
					MeanH	SD	Cover
M1 (LM.comp)	0.131	0.002	0.345	0.025	3.234	5.451	2.886



Figures 1-8; Extensive diagnostics of M1 simple linear regression model

Based on these results, 7 new models were generated. M2 included a simple linear regression based on the cleaned data. M3 included a stepwise linear regression using two predictor variables (MeanH and SD) and complete dataset, which were selected from a stepwise backwards 10-kfold validation and the *caret* R package. M4 included a nonlinear least squares regression based on complete dataset and M5 included a nonlinear least squares regression using cleaned dataset. Using the *mass* R package, M6-8 models included iterated re-weighted least square regressions based on the Huber-loss estimator, the Tukey estimator and the Hampel estimator. Residuals of these 7 additional models were examined using same hypothesis tests as above. In Table 3 below, similar violations were observed. Due to its nonparametric nature, the model that showed fewer problems was M4.

Table 3. Regression diagnostics.

Model	SW	BP	DW	BF	Variance Inflation Factor		
					MeanH	SD	Cover
M2 (LM.clean)	0.949	0.187	0.303	0.001	3.743	4.951	9.493
M3 (LM.stepwise)	0.916	0.586	0.437	0.010	1.903	1.903	/
M4 (NLS.comp)	0.968	0.417	0.344	0.049	28.554	12.128	2.959
M5 (NLS.clean)	0.961	0.110	0.287	0.002	34.967	13.391	5.221
M6 (RR.huber)	0.002	0.002	0.345	0.025	3.234	5.451	2.886
M7 (RR.tukey)	0.000	0.002	0.345	0.025	3.234	5.451	2.886
M8 (RR.hampel)	0.005	0.002	0.345	0.025	3.234	5.451	2.886

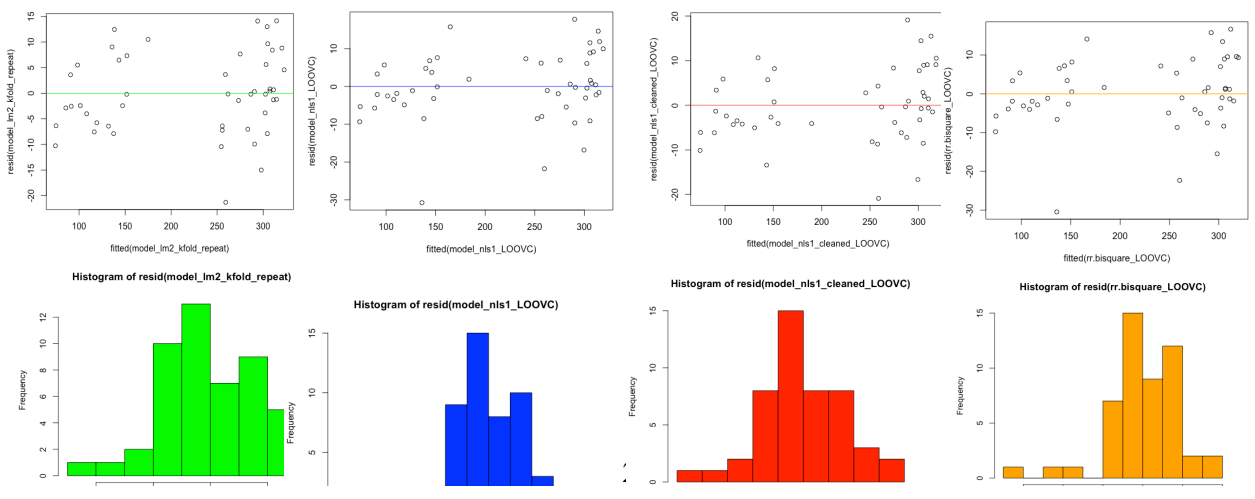
2. Accuracy assessments of LiDAR AGB models

As results above were not conclusive with regards to regression violations, all models were analysed using a resampling validation technique: the 10-Kfold cross validation (3-repeat). This method was used to provide estimates of the bias between resampling procedures and estimates of each model. It was also used to explore levels of precision in terms of the variation of results (RMSE). Using *DescTools* package in R, Theil's U estimate of error was provided to examine the level of unexplained variance in each model (Table 4.).

Table 4. Accuracy assessment of 8 candidate regressions using full models and cross validation estimates

	Full Models					Cross Validation	
Model	RMSE(%)	MAE(%)	Theil U_{error}	AIC	BIC	RMSE%	MAE%
1.LM.Full	16.44 (7.5)	11.95 (5.4)	0.998	427.727	437.286	16.63 (7.6)	13.67 (6.2)
2.LM.Cleaned	15.30 (6.9)	10.95 (4.9)	0.998	403.926	413.282	15.92 (7.1)	13.15 (5.9)
3.LM.Stepwise	17.86 (8.1)	12.47 (5.7)	0.997	435.041	442.689	16.44 (7.5)	13.34 (6.1)
4.NLS.Full	16.30 (7.4)	11.19 (5.1)	0.998	427.766	439.239	16.34 (7.4)	13.26 (6.0)
5.NLS.Cleaned	15.18 (6.8)	10.37 (4.6)	0.998	404.083	415.310	15.90 (7.1)	13.91 (6.2)
6.RR.Huber	13.11 (5.9)	11.17 (5.1)	1.001	430.087	429.647	17.19 (7.8)	14.08 (6.4)
7.RR.Tukey	12.14 (5.5)	11.17 (5.1)	1.006	435.415	444.975	17.31 (7.9)	13.94 (6.3)
8.RR.Hampel	13.34 (6.1)	11.17 (5.1)	1.001	428.938	438.499	17.48 (7.9)	14.08 (6.4)

Based on these results, the Tukey robust regression models provided highest accuracy based on its lowest RMSE score. However, all robust regressions showed signs of over-fitting the data. This was evident from the substantial difference observed between full model and cross validation error rates, as was as their higher AIC and BIC scores. Therefore, I selected the next best model M5: the nonlinear least squared model based on cleaned dataset (RMSE = 15.18). However, cleaning the dataset may not be sensible. As this remains uncertain, the residual spread of all four final trained candidate models are presented below (M2 = green, M4 = blue, M5 = red, M7 = orange).



Bibliography

1. Garcia M, Saatchi S, Ustin S, Balzter H. Modelling forest canopy height by integrating airborne LiDAR samples with satellite Radar and multispectral imagery. Int J Appl earth Obs Geoinf. 2018;66:159–73.

2. Guo Z, Chi H, Sun G. Estimating forest aboveground biomass using HJ-1 Satellite CCD and ICESat GLAS waveform data. Sci China Earth Sci. 2010;53(1):16–25.

3. Næsset E, Gobakken T, Solberg S, Gregoire TG, Nelson R, Ståhl G, et al. Model-assisted regional forest biomass estimation using LiDAR and InSAR as auxiliary data: A case study from a boreal forest area. Remote Sens Environ. 2011;115(12):3599–614.

4. Nelson R, Margolis H, Montesano P, Sun G, Cook B, Corp L, et al. Lidar-based estimates of aboveground biomass in the continental US and Mexico using ground, airborne, and satellite observations. Remote Sens Environ. 2017;188:127–40.

5. Nelson R, Krabill W, MacLean G. Determining forest canopy characteristics using airborne laser data. Remote Sens Environ. 1984;15(3):201–12.

6. Maclean GA, Krabill WB. Gross-merchantable timber volume estimation using an airborne LIDAR system. Can J Remote Sens. 1986;12(1):7–18.

7. Ene LT, Gobakken T, Andersen H-E, Næsset E, Cook BD, Morton DC, et al. Large-area hybrid estimation of aboveground biomass in interior Alaska using airborne laser scanning data. Remote Sens Environ. 2018;204:741–55.

8. Laurin GV, Puletti N, Chen Q, Corona P, Papale D, Valentini R. Above ground biomass and tree species richness estimation with airborne lidar in tropical Ghana forests. Int J Appl earth Obs Geoinf. 2016;52:371–9.

9. Magnussen S, Nord-Larsen T, Riis-Nielsen T. Lidar supported estimators of wood volume and aboveground biomass from the Danish national forest inventory (2012--2016). Remote Sens Environ. 2018;211:146–53.

10. Shao G, Shao G, Gallion J, Saunders MR, Frankenberger JR, Fei S. Improving Lidar-based aboveground biomass estimation of temperate hardwood forests with varying site productivity. Remote Sens Environ. 2018;204:872–82.

11. Næsset E. Predicting forest stand characteristics with airborne scanning laser using a practical two-stage procedure and field data. Remote Sens Environ. 2002;80(1):88–99.

12. Nelson R, Oderwald R, Gregoire TG. Separating the ground and airborne laser sampling phases to estimate tropical forest basal area, volume, and biomass. Remote Sens Environ. 1997;60(3):311–26.

13. Nilsson M. Estimation of tree heights and stand volume using an airborne lidar system. Remote Sens Environ. 1996;56(1):1–7.

14. Gougeon FA, Leckie DG, others. Forest information extraction from high spatial resolution images using an individual tree crown approach. Pacific Forestry Centre; 2002.

15. Reutebuch SE, Andersen H-E, McGaughey RJ. Light detection and ranging (LIDAR): an emerging tool for multiple resource inventory. J For. 2005;103(6):286–92.

16. Lee H, Slatton KC, Roth BE, Cropper Jr WP. Adaptive clustering of airborne LiDAR data to segment individual tree crowns in managed pine forests. Int J Remote Sens. 2010;31(1):117–39.

17. Andersen H-E, McGaughey RJ, Reutebuch SE. Estimating forest canopy fuel parameters using LIDAR data. Remote Sens Environ. 2005;94(4):441–9.

18. Shi Y, Wang T, Skidmore AK, Heurich M. Important LiDAR metrics for discriminating forest tree species in Central Europe. ISPRS J Photogramm Remote Sens. 2018;137:163–74.

19. Brandtberg T. Classifying individual tree species under leaf-off and leaf-on conditions using airborne lidar. ISPRS J Photogramm Remote Sens. 2007;61(5):325–40.

20. Tojal L-T, Bastarrika A, Barrett B, Sanchez Espeso JM, Lopez-Guede JM, Graña M. Prediction of Aboveground Biomass from Low-Density LiDAR Data: Validation over P. radiata Data from a Region North of Spain. Forests. 2019;10(9):819.

21. Corona P. Integration of forest mapping and inventory to support forest management. iForest-Biogeosciences For. 2010;3(3):59.

22. Corona P. Consolidating new paradigms in large-scale monitoring and assessment of forest ecosystems. Environ Res. 2016;144:8–14.

23. Valbuena R, Mauro F, Rodr\'\iguez-Solano R, Manzanera JA. Partial least squares for discriminating variance components in global navigation satellite systems accuracy obtained under scots pine canopies. For Sci. 2012;58(2):139–53.

24. Hopkinson C, Chasmer L, Hall RJ. The uncertainty in conifer plantation growth prediction from multi-temporal lidar datasets. Remote Sens Environ. 2008;112(3):1168–80.

25. Magnussen S, Boudewyn P. Derivations of stand heights from airborne laser scanner data with canopy-based quantile estimators. Can J For Res. 1998;28(7):1016–31.

26. Means JE, Acker SA, Fitt BJ, Renslow M, Emerson L, Hendrix CJ, et al. Predicting forest

stand characteristics with airborne scanning lidar. Photogramm Eng Remote Sensing. 2000;66(11):1367–72.

27. Naesset E. Determination of mean tree height of forest stands using airborne laser scanner data. ISPRS J Photogramm Remote Sens. 1997;52(2):49–56.

28. Næsset E, Økland T. Estimating tree height and tree crown properties using airborne scanning laser in a boreal nature reserve. Remote Sens Environ. 2002;79(1):105–15.

29. Popescu SC. Estimating biomass of individual pine trees using airborne lidar. Biomass and Bioenergy. 2007;31(9):646–55.

30. Hopkinson C, Chasmer L, Lim K, Treitz P, Creed I. Towards a universal lidar canopy height indicator. Can J Remote Sens. 2006;32(2):139–52.

31. Hopkinson C, Chasmer L. Testing LiDAR models of fractional cover across multiple forest ecozones. Remote Sens Environ. 2009;113(1):275–88.

32. Morsdorf F, Koetz B, Meier E, Itten KI, Allgöwer B. The potential of discrete return, small footprint airborne laser scanning data for vegetation density estimation. Proc ISPRS WG III/3. 2005;3(4):3.

33. Hill RA, Broughton RK. Mapping the understorey of deciduous woodland from leaf-on and leaf-off airborne LiDAR data: A case study in lowland Britain. ISPRS J Photogramm Remote Sens. 2009;64(2):223–33.

34. Wasser L, Day R, Chasmer L, Taylor A. Influence of vegetation structure on lidar-derived canopy height and fractional cover in forested riparian buffers during leaf-off and leaf-on conditions. PLoS One. 2013;8(1).

35. Morsdorf F, Kötz B, Meier E, Itten KI, Allgöwer B. Estimation of LAI and fractional cover from small footprint airborne laser scanning data based on gap fraction. Remote Sens Environ. 2006;104(1):50–61.

36. Cai T, Gao R, Hou J, Chen S, Wang D, He D, et al. A gram-gauss-newton method learning overparameterized deep neural networks for regression problems. arXiv Prepr arXiv190511675. 2019;

#5 steps that follow:
#1.Explore predictors and run linear model using full dataset. Run diagnostics & check violations in residuals
#2.Clean dataset removing biggest outliers
#3.Run 7 alternative models using full & cleaned dataset. Run diagnostics & accuracy assessment
#4.Choose model based on pre/post-diagnostics and accuracy assessment
#5.Plot residuals using benchmark diagnostics

R Syntax:

```
#### MSc Inventory Monitoring and Assessment ####
######  Assignment 3; LIDAR AGB Predict #########
############ S.Murphy 19-04-2020 ################

#using k-fold methods to estimate accuracy of model based on following:
#https://machinelearningmastery.com/how-to-estimate-model-accuracy-in-r-using-the-caret-package/
#library(caret)
#library(klaR)

library(readxl)
ClocaenogField <- read_excel("ClocaenogField.xlsx")
View(ClocaenogField)

#Explore data and normality of predictor variables:
describe(ClocaenogField$MEANH)
describe(ClocaenogField$COVER)
describe(ClocaenogField$SD)
shapiro.test(ClocaenogField$MEANH)
shapiro.test(ClocaenogField$COVER)
shapiro.test(ClocaenogField$SD)

#MODEL 1 (LM1): Linear regression
model_lm1 <- lm(BIOMASS ~ MEANH + COVER + SD, data = ClocaenogField)
summary (model_lm1)

#Run model diagnostics and check linear assumptions
ols_test_normality(model_lm1)
ols_test_breusch_pagan(model_lm1)
dwtest(model_lm1)
ols_coll_diag(model_lm1)
outlierTest(model_lm1, data=Duncan)
ols_test_outlier(model_lm1)
ols_vif_tol(model_lm1)
autoplot(model_lm1)
ols_plot_diagnostics(model_lm1)
ols_plot_cooksd_bar(model_lm1)
ols_plot_dfbetas(model_lm1)
ols_plot_resid_lev(model_lm1)
ols_plot_comp_plus_resid(model_lm1, print_plot = TRUE)

#Bonferroni results significant and influential outliers found.
#Create new dataframe removing outliers and re-run linear model.
#Check again for violations and compare model 1 & 2 with non-linear models.

#remove outliers from rows 9 and 18
```

```
ClocaenogField_cleaned <- ClocaenogField[-c(9,18),]

#MODEL 2 (LM_Cleaned): Linear regression using cleaned dataset
model_lm2 <- lm(BIOMASS ~ MEANH + COVER + SD, data = ClocaenogField_cleaned)
summary (model_lm2)

#run model diagnostics again
ols_test_normality(model_lm2)
ols_test_breusch_pagan(model_lm2)
dwtest(model_lm2)
ols_coll_diag(model_lm2)
outlierTest(model_lm2, data=Duncan)
ols_test_outlier(model_lm2)
ols_vif_tol(model_lm2)
autoplot(model_lm2)
ols_plot_diagnostics(model_lm2)
ols_plot_cooksd_bar(model_lm2)
ols_plot_dfbetas(model_lm2)
ols_plot_resid_lev(model_lm2)
ols_plot_comp_plus_resid(model_lm2, print_plot = TRUE)
#same violations found from bonferroni results and variance inflation factors
#resort to stepwise, non-linear and robust regression and compare...

#MODEL 3 (LM-STEP-Full): Stepwise Linear regression using full dataset
# Step-wise backwards 10-fold cross-validation test of model accuracy based on RMSE
# Set seed for reproducibility
# Set up repeated k-fold cross-validation
train.control_lm1 <- trainControl(method = "cv", number = 10)
# Train the model
step.model_lm1 <- train(BIOMASS ~ MEANH + COVER + SD, data = ClocaenogField, method =
"leapBackward", tuneGrid = data.frame(nvmax = 1:3), trControl = train.control_lm1)
step.model_lm1$results
summary(step.model_lm1$finalModel)
# Results suggest best 2-variable model contain MEANH and SD
# Step-wise model using most best 2-variables: MEANH and SD
model_lm_step_full <- lm(BIOMASS ~ MEANH + SD, data = ClocaenogField)
summary(model_lm_step_full)

#run diagnostics
ols_test_normality(model_lm_step_full)
ols_test_breusch_pagan(model_lm_step_full)
dwtest(model_lm_step_full)
ols_coll_diag(model_lm_step_full)
outlierTest(model_lm_step_full, data=Duncan)
ols_test_outlier(model_lm_step_full)
ols_vif_tol(model_lm_step_full)
autoplot(model_lm_step_full)
ols_plot_diagnostics(model_lm_step_full)
ols_plot_cooksd_bar(model_lm_step_full)
ols_plot_dfbetas(model_lm_step_full)
ols_plot_resid_lev(model_lm_step_full)
ols_plot_comp_plus_resid(model_lm_step_full, print_plot = TRUE)

# MODEL 4 (NLS) Non-linear least squares regression model using full dataset
# Non-linear least squares regression
```

```
model_nls1 <- lm(ClocaenogField$BIOMASS ~ ClocaenogField$MEANH + ClocaenogField$COVER + ClocaenogField$SD + I((ClocaenogField$MEANH + ClocaenogField$COVER + ClocaenogField$SD)^2))
summary(model_nls1)

# MODEL 5 (NLS) Non-linear least squares regression model using cleaned dataset
# Non-linear least squares regression
model_nls1_cleaned <- lm(ClocaenogField_cleaned$BIOMASS ~ ClocaenogField_cleaned$MEANH + ClocaenogField_cleaned$COVER + ClocaenogField_cleaned$SD + I((ClocaenogField_cleaned$MEANH + ClocaenogField_cleaned$COVER + ClocaenogField_cleaned$SD)^2))
summary(model_nls1_cleaned)

# MODEL 6 (RLM.Huber) Robust regression using the Huber M-estimator to reduce outlier influence. Though this does not reduce outliers in predictor variables, so Tukey needed below
rr.huber <- rlm(BIOMASS ~ MEANH + COVER + SD, data = ClocaenogField)
summary(rr.huber)
f.robftest(rr.huber,var = "MEANH")
f.robftest(rr.huber, var = "SD")
f.robftest(rr.huber, var = "COVER")


# MODEL 7 (RLM.Tukey) Robust regression using the Tukey M-estimator that assigns a weight of zero to influential outliers
rr.bisquare <- rlm(BIOMASS ~ MEANH + COVER + SD, data = ClocaenogField, psi = psi.bisquare)
summary(rr.bisquare)
f.robftest(rr.bisquare,var = "MEANH")
f.robftest(rr.bisquare, var = "SD")
f.robftest(rr.bisquare, var = "COVER")

# MODEL 8 (RLM.Tukey) Robust regression using the Tukey M-estimator that assigns a weight of zero to influential outliers
rr.hampel <- rlm(BIOMASS ~ MEANH + COVER + SD, data = ClocaenogField, psi = psi.hampel)
summary(rr.hampel)
f.robftest(rr.hampel,var = "MEANH")
f.robftest(rr.hampel, var = "SD")
f.robftest(rr.hampel, var = "COVER")


# Explore altnerative accuracy assessments:
# sidak p value adjustment
ols_test_breusch_pagan(model_nls1, rhs = TRUE, multiple = TRUE, p.adj = 'sidak')
# holm's p value adjustment
ols_test_breusch_pagan(model_nls1, rhs = TRUE, multiple = TRUE, p.adj = 'holm')
# Global test of model assumptions
library(gvlma)
gvmodel <- gvlma(model_lm1)
summary(gvmodel)
gvmodel_del <- deletion.gvlma(gvmodel)
summary(gvmodel_del)
plot(gvmodel_del)
display.delstats
summary.gvlmaDel
summary(gvmodel_del, allstats = FALSE)


gvmodel_lm2 <- gvlma(model_lm2)
```

```
summary(gvmodel_lm2)
gvmodel_del_lm2 <- deletion.gvlma(gvmodel_lm2)
summary(gvmodel_del_lm2)
plot(gvmodel_del_lm2)
display.delstats
summary.gvlmaDel
summary(gvmodel_del_lm2, allstats = FALSE)

gvmodel_lm1 <- gvlma(model_lm1_kfold)
summary(model_lm1_kfold)
gvmodel_del_lm2 <- deletion.gvlma(model_lm1_kfold)
summary(gvmodel_del_lm2)
plot(gvmodel_del_lm2)
display.delstats
summary.gvlmaDel
summary(gvmodel_del_lm2, allstats = FALSE)

#compare models
ols_mallows_cp(model_lm1, model_lm2)
ols_fpe(model_lm1)
ols_hsp(model_lm1)

ols_mallows_cp(model_lm1, model_lm2)
ols_mallows_cp(model_lm1, model_lm_step_full)
ols_mallows_cp(model_lm1, model_nls1)
ols_mallows_cp(model_lm1, model_nls1_cleaned)
ols_mallows_cp(model_lm1, rr.huber)
ols_mallows_cp(model_lm1, rr.bisquare)
ols_mallows_cp(model_lm1, rr.hampel)

AIC(model_lm1)
AIC(model_lm2)
AIC(model_lm_step_full)
AIC(model_nls1)
AIC(model_nls1_cleaned)
AIC(rr.huber)
AIC(rr.bisquare)
AIC(rr.hampel)

BIC(model_lm1)
BIC(model_lm2)
BIC(model_lm_step_full)
BIC(model_nls1)
BIC(model_nls1_cleaned)
BIC(rr.huber)
BIC(rr.bisquare)
BIC(rr.hampel)

glance(model_lm1)
glance(model_lm2)
glance(model_lm_step_full)
glance(model_nls1)
glance(model_nls1_cleaned)
glance(rr.huber)
glance(rr.bisquare)
```

```
glance(rr.hampel)

#Explore residual diagnostics
ols_plot_resid_box
ols_plot_resid_fit
ols_plot_resid_hist
ols_plot_resid_qq


#big outliers so choose to run a robust regression using MASS pkg:
#from https://stats.idre.ucla.edu/r/dae/robust-regression/
#Robust regression is iterated re-weighted least squares (IRLS). The
#command is rlm in the MASS package. There are several
#weighting functions that can be used for IRLS. We
#first use the Huber weights and then bi-square weighting. We
#will then look at the final weights created by the IRLS process.

#robust methods due to variance of resids and measures of influence:
#from https://stats.idre.ucla.edu/r/dae/robust-regression/
#We can see that the weight given to Mississippi
#is dramatically lower using the bisquare weighting
#function than the Huber weighting function and
#the parameter estimates from these two different
#weighting methods differ. When comparing the
#results of a regular OLS regression and a robust
#regression, if the results are very different,
#you will most likely want to use the results
#from the robust regression. Large differences
#suggest that the model parameters are being
#highly influenced by outliers. Different functions
#have advantages and drawbacks. Huber weights can
#have difficulties with severe outliers, and
#bisquare weights can have difficulties converging
#or may yield multiple solutions.

#(robust) sandwich variance estimator for linear regression:
#from:  https://thestatsgeek.com/2014/02/14/the-robust-sandwich-variance-estimator-for-linear-regression-
using-r/
#This method allowed us to estimate valid standard errors
#for our coefficients in linear regression, without requiring
#the usual assumption that the residual errors have constant variance.


# 10 k-fold cross validation
# define training control
train_control_kfold <- trainControl(method="cv", number=10)
metric <- "Accuracy"
# fix the parameters of the algorithm
grid <- expand.grid(.fL=c(0), .usekernel=c(FALSE))
# train the model
model_lm1_kfold <- train(BIOMASS ~ MEANH + SD + COVER, data=ClocaenogField,
trControl=train_control_kfold)
# summarize results
print(model_lm1_kfold)
View(model_lm1_kfold)
```

```
summary(model_lm1_kfold)

set.seed(7)
accuracy_model_lm1 <- train(BIOMASS ~ MEANH + SD + COVER, data=ClocaenogField, metric=metric, trControl=train_control_kfold)


# repeated 10 k-fold cross validation
# define training control
train_control_kfold_repeat <- trainControl(method="repeatedcv", number=10, repeats=3)
grid <- expand.grid(.fL=c(0), .usekernel=c(FALSE))
# train the model
model_lm1_kfold_repeat <- train(BIOMASS ~ MEANH + SD + COVER, data = ClocaenogField, trControl = train_control_kfold_repeat)
model_lm2_kfold_repeat <- train(BIOMASS ~ MEANH + SD + COVER, data = ClocaenogField_cleaned, trControl=train_control_kfold_repeat)
model_lm_step_full_kfold_repeat <- train(BIOMASS ~ MEANH + SD, data = ClocaenogField, trControl=train_control_kfold_repeat)
model_nls1_kfold_repeat <- train(BIOMASS ~ MEANH + SD + COVER, data = ClocaenogField, trControl=train_control_kfold_repeat)
model_nls1_cleaned_kfold_repeat <- train(BIOMASS ~ MEANH + SD + COVER, data = ClocaenogField_cleaned, trControl=train_control_kfold_repeat)
rr.huber_kfold_repeat <- train(BIOMASS ~ MEANH + SD + COVER, data = ClocaenogField, trControl=train_control_kfold_repeat)
rr.bisquare_kfold_repeat <- train(BIOMASS ~ MEANH + SD + COVER, data = ClocaenogField, trControl=train_control_kfold_repeat)
rr.hampel_kfold_repeat <-train(BIOMASS ~ MEANH + SD + COVER, data = ClocaenogField, trControl=train_control_kfold_repeat)

print(model_lm1_kfold_repeat)
print(model_lm2_kfold_repeat)
print(model_lm_step_full_kfold_repeat)
print(model_nls1_kfold_repeat)
print(model_nls1_cleaned_kfold_repeat)
print(rr.huber_kfold_repeat)
print(rr.bisquare_kfold_repeat)
print(rr.hampel_kfold_repeat)

# Leave-one-out cross validation
# define training control
train_control_LOOVC <- trainControl(method="LOOCV")
# train the model
model_lm1_LOOVC <- train(BIOMASS ~ MEANH + SD + COVER, data = ClocaenogField, trControl=train_control_LOOVC)
model_lm2_LOOVC <- train(BIOMASS ~ MEANH + SD + COVER, data = ClocaenogField_cleaned, trControl=train_control_LOOVC)
model_lm_step_full_LOOVC <- train(BIOMASS ~ MEANH + SD, data = ClocaenogField, trControl=train_control_LOOVC)
model_nls1_LOOVC <- train(BIOMASS ~ MEANH + SD + COVER, data = ClocaenogField, trControl=train_control_LOOVC)
model_nls1_cleaned_LOOVC <- train(BIOMASS ~ MEANH + SD + COVER, data = ClocaenogField_cleaned, trControl=train_control_LOOVC)
rr.huber_LOOVC <- train(BIOMASS ~ MEANH + SD + COVER, data = ClocaenogField, trControl=train_control_LOOVC)
```

```
rr.bisquare_LOOVC <- train(BIOMASS ~ MEANH + SD + COVER, data = ClocaenogField, trControl=train_control_LOOVC)
rr.hampel_LOOVC <-train(BIOMASS ~ MEANH + SD + COVER, data = ClocaenogField, trControl=train_control_LOOVC)

print(model_lm1_LOOVC)
summary(model_lm1_LOOVC)
View(model_lm1_LOOVC)

print(model_lm2_LOOVC)
print(model_lm_step_full_LOOVC)
print(model_nls1_LOOVC)
print(model_nls1_cleaned_LOOVC)
print(rr.huber_LOOVC)
print(rr.bisquare_LOOVC)
print(rr.hampel_LOOVC)

#run diagnostics
shapiro.test(resid(model_lm1_kfold_repeat))
shapiro.test(resid(model_lm2_kfold_repeat))
shapiro.test(resid(model_lm_step_full_kfold_repeat))
shapiro.test(resid(model_nls1_kfold_repeat))
shapiro.test(resid(model_nls1_cleaned_kfold_repeat))
shapiro.test(resid(rr.huber_kfold_repeat))
shapiro.test(resid(rr.bisquare_kfold_repeat))
shapiro.test(resid(rr.hampel_kfold_repeat))

bptest(model_lm1_kfold_repeat)
bptest(model_lm2_kfold_repeat)
bptest(model_lm_step_full_kfold_repeat)
bptest(model_nls1_kfold_repeat)
bptest(model_nls1_cleaned_kfold_repeat)
bptest(rr.huber_kfold_repeat)
bptest(rr.bisquare_kfold_repeat)
bptest(rr.hampel_kfold_repeat)

dwtest(rr.huber)
dwtest(rr.bisquare)
dwtest(rr.hampel)

outlierTest(lm(rr.huber))
outlierTest(lm(rr.bisquare))
outlierTest(lm(rr.hampel))

vif(rr.huber)
vif(rr.bisquare)
vif(rr.hampel)

AIC(model_lm1)
AIC(model_lm2)
AIC(model_lm_step_full)
AIC(model_nls1)
AIC(model_nls1_cleaned)
AIC(rr.huber)
AIC(rr.bisquare)
```

```
AIC(rr.hampel)


BIC(model_lm1)
BIC(model_lm2)
BIC(model_lm_step_full)
BIC(model_nls1)
BIC(model_nls1_cleaned)
BIC(rr.huber)
BIC(rr.bisquare)
BIC(rr.hampel)

TheilU(ClocaenogField$BIOMASS, model_lm1, type=1)
theil.wtd(model_lm1, weights = NULL)


bias_training <- rnorm(40, 2, sd = 0.5)
bias(bias_training, ClocaenogField)
bias(samp, pop)
bias(samp, pop, type = 'relative')
bias(samp, pop, type = 'standardized')
dev <- samp - pop
bias(dev)
nom.uncertainty(model_lm1)
lambda3(model_nls1)
CronbachAlpha(model_lm1)
guttman(model_lm1, missing = "complete", standardize = FALSE)


loq(model_lm1)



ols_plot_response(model_lm1)

u_theil_train <- createDataPartition(ClocaenogField$BIOMASS, p=0.5, list = FALSE)
u_theil_trainingData <- ClocaenogField[u_theil_train,]

u_theil_knn_model = train(model_lm1, data = u_theil_train, method = "knn", trControl = trainControl(method
= "cv", number = 5), tuneGrid = expand.grid((k = seq(1, 21, by = 2)))
u_theil_knn_model$modelType
summary(u_theil_knn_model)

u_theil_testData <- ClocaenogField[-u_theil_train,]
TheilU(actual_biomass, lm1_residuals)


dim(u_theil_trainingData)

lm1_residuals <-data.frame(lm=model_lm1$residuals)
training_sample <- rnorm (100, 2, sd = 0.5)
dev <- training_sample - ClocaenogField
bias(dev)
bias(mean(training_sample), ClocaenogField)
percent_bias(model_lm1, ClocaenogField)
```

```
lod(model_lm1, data = ClocaenogField)

summary(model_lm1)
computeBoundary(b1=2.8, b0=3.3, p=c(.5, .75))

m.nn_lm1        <- matchit(BIOMASS ~ MEANH + SD + COVER, data = ClocaenogField,        method=
"nearest",        ratio     = 1)
summary(m.nn)


ape(actual_biomass, lm1_residuals)


MAE(model_lm1)
MAE(model_lm2)
MAE(model_lm_step_full)
MAE(model_nls1)
MAE(model_nls1_cleaned)
MAE(rr.bisquare)
MAE(rr.bisquare)
MAE(rr.bisquare)

coxtest(model_lm1, model_lm1_kfold_repeat, data = ClocaenogField)

rsq.kl(model_lm1_kfold_repeat)
rsq.n(model_lm1_kfold_repeat)
rsq.partial(model_lm1_kfold_repeat)
rsq.sse(model_lm1_kfold_repeat)
rsq.v(model_lm1_kfold_repeat)

AIC(model_lm2)
AIC(model_lm_step_full)
AIC(model_nls1)
AIC(model_nls1_cleaned)
AIC(rr.huber)
AIC(rr.bisquare)
AIC(rr.hampel)


actual_biomass <- data.frame(ClocaenogField$BIOMASS)
actual_biomass_cleaned <- data.frame(ClocaenogField_cleaned$BIOMASS)


kfoldrepeat_residuals_lm1 <-data.frame(lm=model_lm1_kfold_repeat$residuals)
kfoldrepeat_residuals_lm2 <-data.frame(lm=model_lm2_kfold_repeat$residuals)
kfoldrepeat_residuals_lmstep <-data.frame(lm=model_lm_step_full_kfold_repeat$residuals)
kfoldrepeat_residuals_nls <-data.frame(lm=model_nls1_LOOVC$residuals)
kfoldrepeat_residuals_nlsclean <-data.frame(lm=model_nls1_cleaned_LOOVC$residuals)
kfoldrepeat_residuals_rrhuber <-data.frame(lm=rr.huber_LOOVC$residuals)
kfoldrepeat_residuals_rrbisq <-data.frame(lm=rr.bisquare_LOOVC$residuals)
kfoldrepeat_residuals_rrhampel <-data.frame(lm=rr.hampel_LOOVC$residuals)

lmFull_residuals <-data.frame(lm=model_lm1$residuals)
lmClean_residuals <-data.frame(lm=model_lm2$residuals)
```

```
lmStep_residuals <-data.frame(lm=model_lm_step_full$residuals)
nlsFull_residuals <-data.frame(lm=model_nls1$residuals)
nlsClean_residuals <-data.frame(lm=model_nls1_cleaned$residuals)
rrHuber_residuals <-data.frame(lm=rr.huber$residuals)
rrTukey_residuals <-data.frame(lm=rr.bisquare$residuals)
rrHampel_residuals <-data.frame(lm=rr.hampel$residuals)


library(DescTools)
TheilU(actual_biomass, lmFull_residuals)
TheilU(actual_biomass_cleaned, lmClean_residuals)
TheilU(actual_biomass, lmStep_residuals)
TheilU(actual_biomass, nlsFull_residuals)
TheilU(actual_biomass_cleaned, nlsClean_residuals)
TheilU(actual_biomass, rrHuber_residuals)
TheilU(actual_biomass, rrTukey_residuals)
TheilU(actual_biomass, rrHampel_residuals)

TheilU(actual_biomass, kfoldrepeat_residuals_lm1)
TheilU(actual_biomass_cleaned, kfoldrepeat_residuals_lm2)
TheilU(actual_biomass, kfoldrepeat_residuals_lmstep)
TheilU(actual_biomass, kfoldrepeat_residuals_nls)
TheilU(actual_biomass_cleaned, kfoldrepeat_residuals_nlsclean)
TheilU(actual_biomass, kfoldrepeat_residuals_rrhuber)
TheilU(actual_biomass, kfoldrepeat_residuals_rrbisq)
TheilU(actual_biomass, kfoldrepeat_residuals_rrhampel)

shapiro.test(resid(model_lm1))
shapiro.test(resid(model_lm2_kfold_repeat))
shapiro.test(resid(model_lm_step_full_kfold_repeat))
shapiro.test(resid(model_nls1_LOOVC))
shapiro.test(resid(model_nls1_cleaned_LOOVC))
shapiro.test(resid(rr.huber_LOOVC))
shapiro.test(resid(rr.bisquare_LOOVC))
shapiro.test(resid(rr.hampel_LOOVC))

bptest(model_lm1)
bptest(model_lm2_kfold_repeat)
bptest(resid(model_lm_step_full_kfold_repeat))
bptest(resid(model_nls1_LOOVC))
bptest(resid(model_nls1_cleaned_LOOVC))
bptest(resid(rr.huber_LOOVC))
bptest(resid(rr.bisquare_LOOVC))
bptest(resid(rr.hampel_LOOVC))


#model diagnostics:
#check for trends
#check for equal distribution of predictors
plot (ClocaenogField$MEANH, resid (model_lm1))
abline (0,0)
plot (ClocaenogField$MEANH, resid (model_lm1))
abline (0,0)
```

```
plot (ClocaenogField$COVER, resid (model2))
abline (0,0)
plot (ClocaenogField$SD, resid (model2))
abline (0,0)



PlotCandlestick(model_lm1)

#homoskedasticity
#predicted values aginst fitted
PlotQQ(model_lm1)
autoplot(model_lm1_kfold_repeat, label.size = 3)

plot(fitted(model_lm1_kfold_repeat), resid(model_lm1_kfold_repeat))
abline(a = coef(model_lm1_kfold_repeat), b = 0, col="green")
plot(fitted(model_nls1_LOOVC), resid(model_nls1_LOOVC))
lines(abline(a = coef(model_nls1_LOOVC), b = 0, col="red"))
plot(fitted(model_nls1_cleaned_LOOVC), resid(model_nls1_cleaned_LOOVC))
lines(abline(a = coef(model_nls1_cleaned_LOOVC), b = 0, col="orange"))
plot(fitted(rr.bisquare_LOOVC), resid(rr.bisquare_LOOVC))
lines(abline(a = coef(rr.bisquare_LOOVC), b = 0, col="yellow"))


plot(fitted(model_lm2_kfold_repeat), resid(model_lm2_kfold_repeat))
lines(abline(a = coef(model_lm2_kfold_repeat), b = 0, col="blue"))

legend(46, 15, legend = c("model1: linear", "model2: poly x^2", "model3: poly x^2 + x^3"),
      col=c("green", "blue", "red", "orange", "yellow"), lwd=3, bty="n", cex=0.9)
lines(smooth.spline(SBI_species$dbh_cm, predict(predict3.z05.55)), col="green", lwd=3, lty=3)


legend(46, 15, legend = c("model1: linear", "model2: poly x^2", "model3: poly x^2 + x^3"),
      col=c("red", "blue", "green"), lwd=3, bty="n", cex=0.9)



lines(smooth.spline(SS_species$dbh_cm, predict(predict2_C14.16)), lwd=3, col="blue")
predict3_C14.16 <- lm(SS_species$tree_agb_kg_C14.16 ~ SS_species$dbh_cm + I(SS_species$dbh_cm^2) +
I(SS_species$dbh_cm^3))
lines(smooth.spline(SS_species$dbh_cm, predict(predict3_C14.16)), col="green", lwd=3, lty=3)
legend(46, 15, legend = c("model1: linear", "model2: poly x^2", "model3: poly x^2 + x^3"),
#breusch-pagan test of homoskedasticity
bptest(model2)
bptest(model1)



#normality of residuals
qqnorm  (rstandard(model_lm1))
abline(0,1)
hist(resid(model2))
ols_test_normality(model2)
```

```
plot(data$BIOMASS, fitted(modmel2)
    abline(0,1)

    sqrt(mean(sqres))
    sqrt(mean(sqres))/mean(data$BIOMASS)*100



plot(SS_species$dbh_cm, SS_species$tree_agb_kg_C14.16)
abline(predict1_C14.16, col="red", lwd=3)
predict2_C14.16 <- lm(SS_species$tree_agb_kg_C14.16 ~ SS_species$dbh_cm + I(SS_species$dbh_cm^2))
lines(smooth.spline(SS_species$dbh_cm, predict(predict2_C14.16)), lwd=3, col="blue")
predict3_C14.16 <- lm(SS_species$tree_agb_kg_C14.16 ~ SS_species$dbh_cm + I(SS_species$dbh_cm^2) +
I(SS_species$dbh_cm^3))
lines(smooth.spline(SS_species$dbh_cm, predict(predict3_C14.16)), col="green", lwd=3, lty=3)
 <-data.frame(lm=model_lm1_kfold_repeat$residuals)


kfoldrepeat_residuals_lm2 <-data.frame(lm=model_lm2_kfold_repeat$residuals)
kfoldrepeat_residuals_lmstep <-data.frame(lm=model_lm_step_full_kfold_repeat$residuals)
kfoldrepeat_residuals_nls <-data.frame(lm=model_nls1_LOOVC$residuals)
kfoldrepeat_residuals_nlsclean <-data.frame(lm=model_nls1_cleaned_LOOVC$residuals)
kfoldrepeat_residuals_rrhuber <-data.frame(lm=rr.huber_LOOVC$residuals)
kfoldrepeat_residuals_rrbisq <-data.frame(lm=rr.bisquare_LOOVC$residuals)
kfoldrepeat_residuals_rrhampel <-data.frame(lm=rr.hampel_LOOVC$residuals)

shapiro.test(resid(model_lm1))
bptest(model_lm2_kfold_repeat)
bptest(model_lm_step_full_kfold_repeat)
bptest(model_nls1_kfold_repeat)
bptest(model_nls1_cleaned_LOOVC)
bptest(rr.huber_LOOVC)
bptest(rr.bisquare_LOOVC)
bptest(rr.hampel_LOOVC)

dwtest(rr.huber)
dwtest(rr.bisquare)
dwtest(rr.hampel)

outlierTest(lm(rr.huber))
outlierTest(lm(rr.bisquare))
outlierTest(lm(rr.hampel))




plot(fitted(model_lm2_kfold_repeat), resid(model_lm2_kfold_repeat))
abline(0,0, col = "green")
hist(resid(model_lm2_kfold_repeat), col = "green")
```

```
plot(fitted(model_nls1_LOOVC), resid(model_nls1_LOOVC))
abline(0,0, col = "blue")
hist(resid(model_nls1_LOOVC), col = "blue")

plot(fitted(model_nls1_cleaned_LOOVC), resid(model_nls1_cleaned_LOOVC))
abline(0,0, col = "red")
hist(resid(model_nls1_cleaned_LOOVC), col = "red")

plot(fitted(rr.bisquare_LOOVC), resid(rr.bisquare_LOOVC))
abline(0,0, col = "orange")
hist(resid(rr.bisquare_LOOVC), col = "orange")
```