Analysis

Adama NDOUR

2024-07-16

## Load package

library(tidyverse)  
library(readxl)  
library(paletteer)  
library(meta) # meta-analysis  
library(nnet) # Multinomial logit model  
library(GGally) # plot model coefficients  
library(recipes)  
library(recipeselectors)  
library(embed) # encoding  
library(report) # report statistical results  
library(stargazer) # formatting statistical results  
library(kableExtra) # format table  
library(DHARMa) # plot residuals of a linear model

## Load the data

df <- read\_excel("uav\_review\_data.xlsx")

## Overview of the data

#str(df)

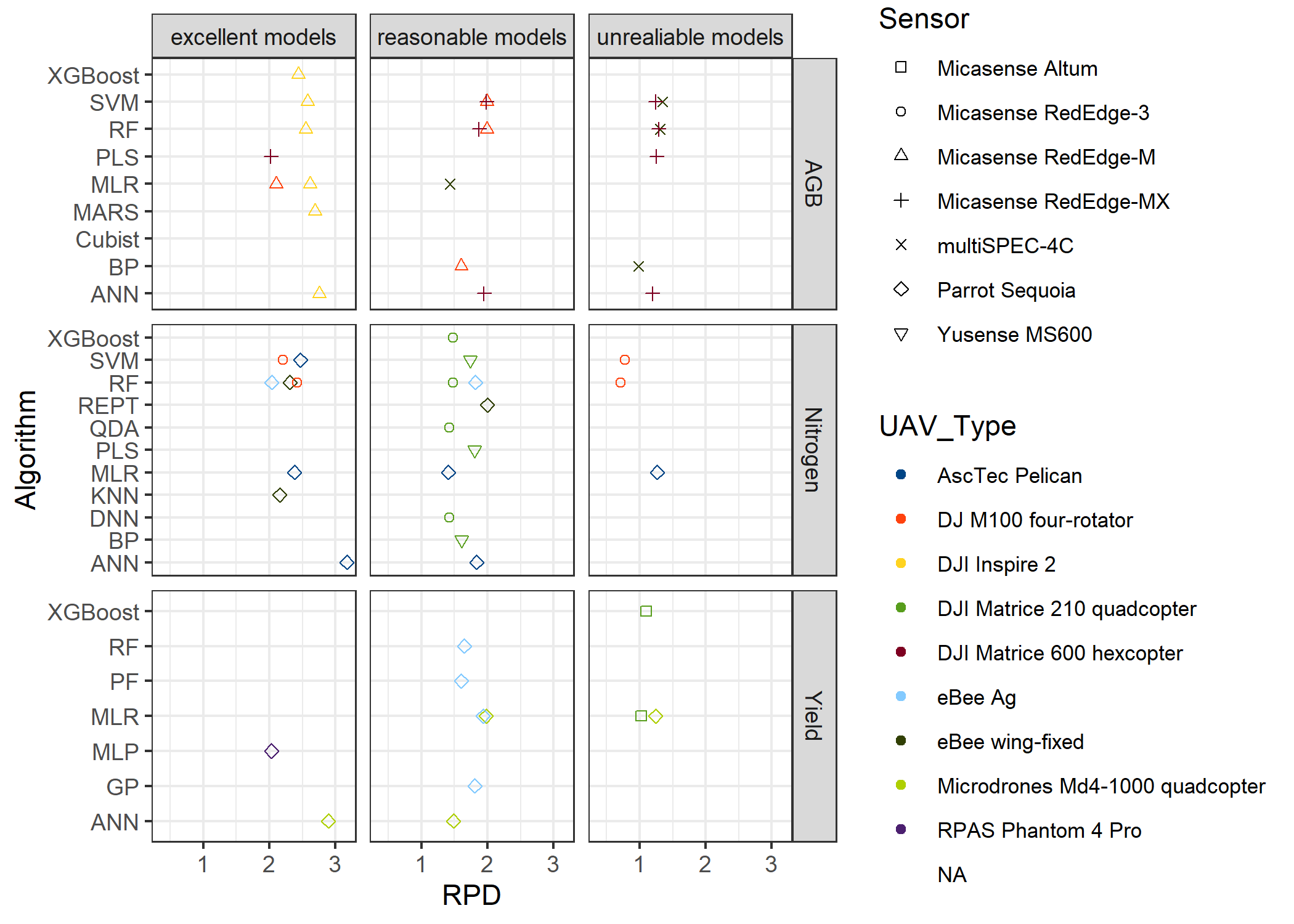
## Data manipulation: create a model class variable

df <- df %>% mutate(  
 Model\_Class = case\_when(  
 RPD < 1.4 ~ "unrealiable models",  
 RPD >= 1.4 & RPD < 2 ~ "reasonable models",  
 RPD >=2 ~ "excellent models"  
 )  
)  
  
# Remove special characters  
df$Sensor <- str\_replace\_all(df$Sensor, "\r", " ")  
df$Sensor <- str\_replace\_all(df$Sensor, "\n", " ")  
df$Sensor <- str\_replace\_all(df$Sensor, "\\s+", " ")

## Exploratory Data Analysis (EDA)

### Which UAV platform maximize the performance of ML models

df\_trait <- df %>% filter(Problem=="trait estimation")  
#df\_trait %>% group\_by(Crop,Algorithm)  
n\_shape\_var <- length(unique(df\_trait$Sensor))  
p<-df\_trait %>%   
 group\_by(DOI, Trait, UAV\_Type, Sensor, Algorithm,Model\_Class) %>%   
 summarise(RPD=mean(RPD)) %>%   
 ggplot(aes(y=Algorithm, x=RPD, color=UAV\_Type)) +  
 geom\_point(aes(shape=Sensor))+  
 scale\_shape\_manual(values = 0:n\_shape\_var) +  
 scale\_color\_paletteer\_d("ggthemes::calc")+  
 facet\_grid(Trait~ Model\_Class, scales = "free\_y")+  
 theme\_bw()+  
 theme(  
 legend.text = element\_text(size = 8.5)  
 )  
ggsave("output/figure1.png",plot = p,dpi = 300)  
p

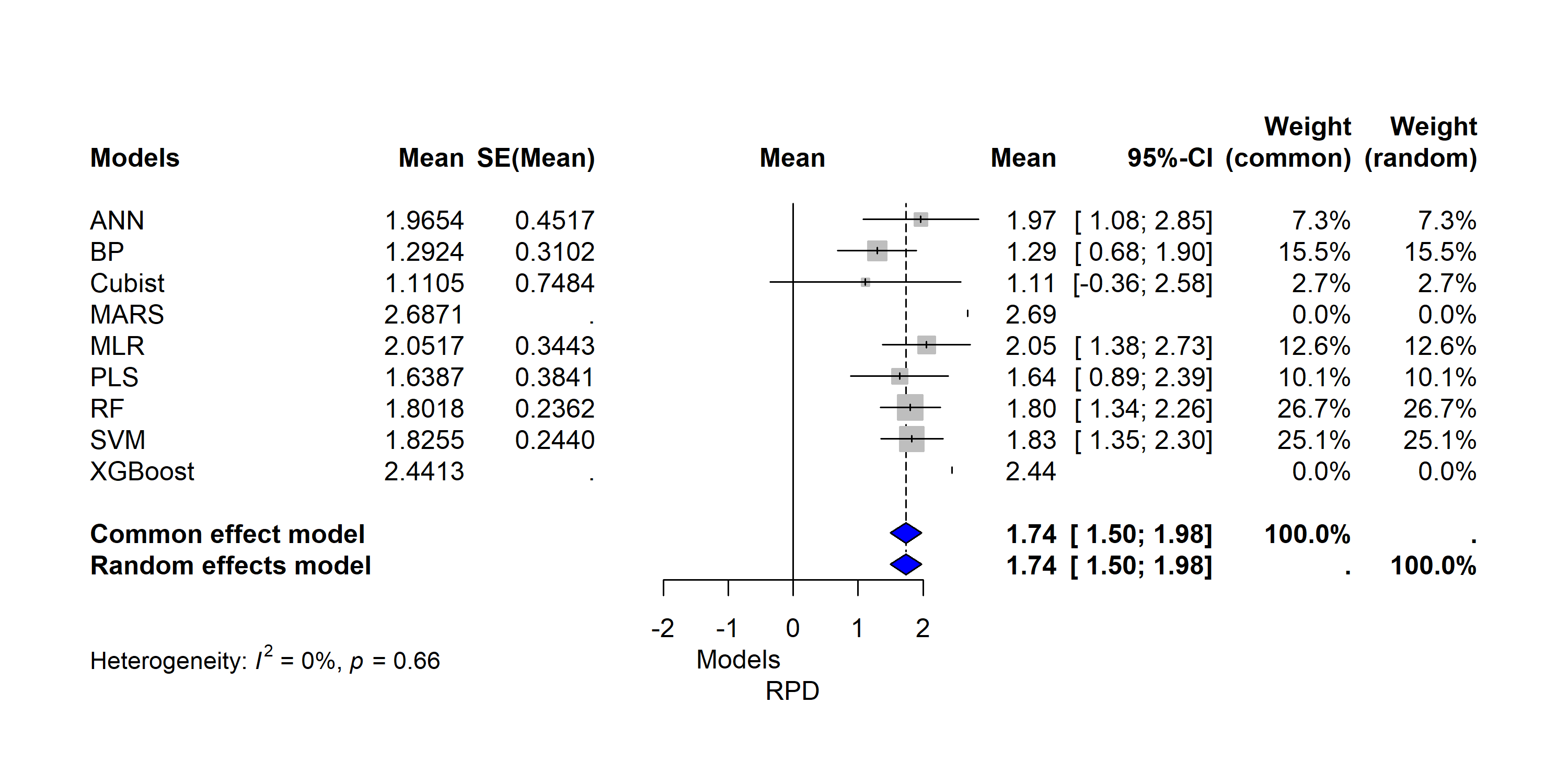


### Forest plot for the biomass

# Install and load necessary packages  
# Load necessary libraries  
  
# Example RPD data  
biomass\_data <- df\_trait %>% filter(Trait=="AGB")  
  
# Calculate summary statistics  
biomass\_rpd\_summary <- biomass\_data %>%  
 group\_by(Algorithm) %>%  
 summarize(  
 mean\_RPD = mean(RPD),  
 sd\_RPD = sd(RPD),  
 n = n(),  
 SEM\_RPD = sd\_RPD / sqrt(n),  
 CI\_Lower = mean\_RPD - 1.96 \* SEM\_RPD,  
 CI\_Upper = mean\_RPD + 1.96 \* SEM\_RPD  
 )  
  
# Print the summary  
print(biomass\_rpd\_summary)

## # A tibble: 9 × 7  
## Algorithm mean\_RPD sd\_RPD n SEM\_RPD CI\_Lower CI\_Upper  
## <chr> <dbl> <dbl> <int> <dbl> <dbl> <dbl>  
## 1 ANN 1.97 0.782 3 0.452 1.08 2.85  
## 2 BP 1.29 0.439 2 0.310 0.684 1.90  
## 3 Cubist 1.11 1.06 2 0.748 -0.356 2.58  
## 4 MARS 2.69 NA 1 NA NA NA   
## 5 MLR 2.05 0.596 3 0.344 1.38 2.73  
## 6 PLS 1.64 0.543 2 0.384 0.886 2.39  
## 7 RF 1.80 0.528 5 0.236 1.34 2.26  
## 8 SVM 1.83 0.546 5 0.244 1.35 2.30  
## 9 XGBoost 2.44 NA 1 NA NA NA

# Combine data for all models  
biomass\_meta\_combined <- metagen(  
 TE = biomass\_rpd\_summary$mean\_RPD,  
 lower = biomass\_rpd\_summary$CI\_Lower,  
 upper = biomass\_rpd\_summary$CI\_Upper,  
 studlab = biomass\_rpd\_summary$Algorithm,  
 sm = "Mean"  
)  
  
# Forest plot for all models  
# png(file = "output/forestplot\_biomass.png", width = 10, height = 5, res = 300, units = "in")  
forest(biomass\_meta\_combined,  
 main = "Forest Plot of RPD for All Models of Biomass Estimation",  
 xlab = "RPD",  
 label.left = "Models",  
 studlab = biomass\_rpd\_summary$Algorithm,  
 print.tau2 = FALSE,  
 col.diamond = "blue",  
 col.predict = "red",  
 leftlabs = c("Models", "Mean", "SE(Mean)"))

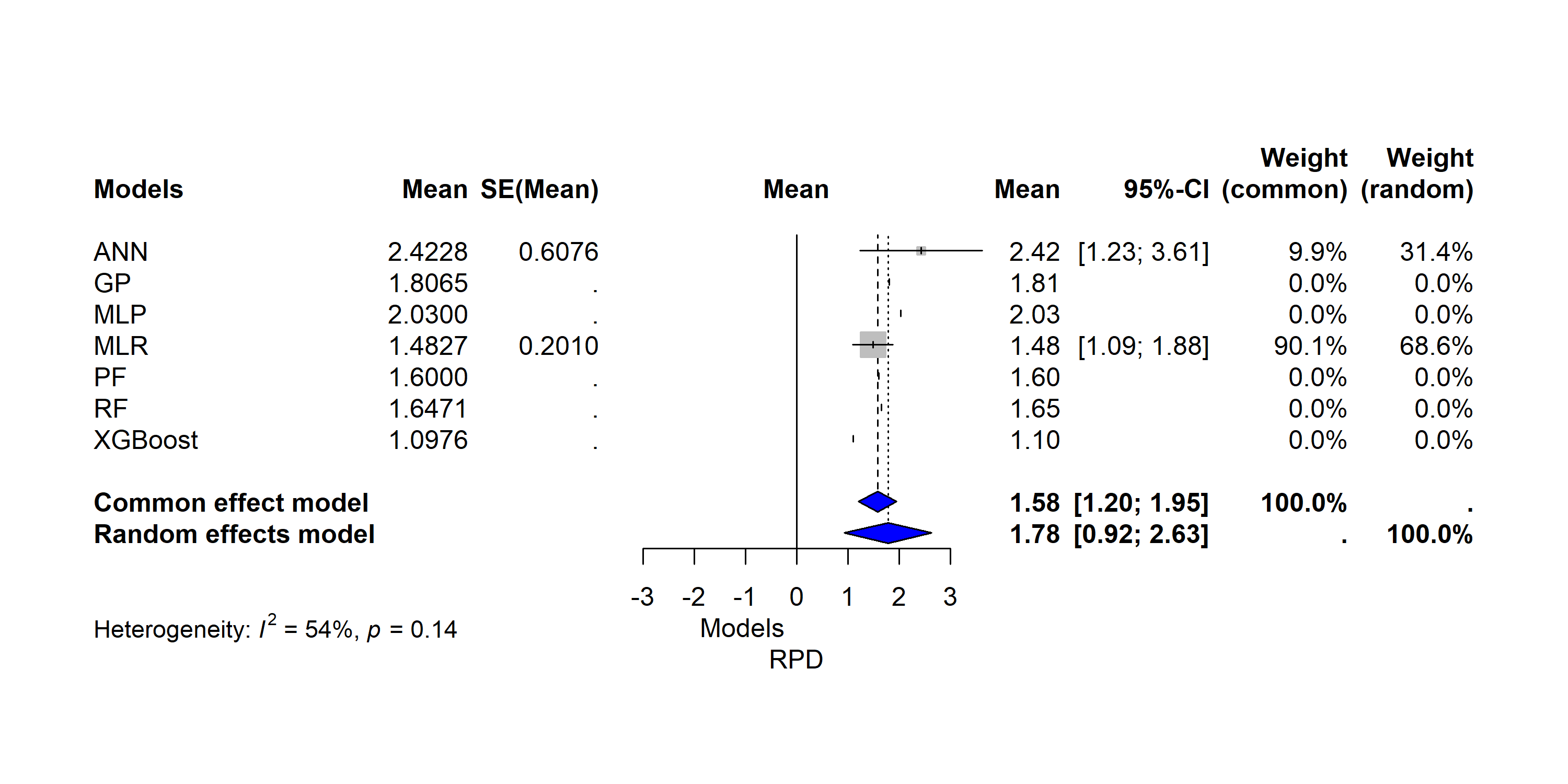


### Forest plot for the yield

# Install and load necessary packages  
# Load necessary libraries  
  
# Example RPD data  
yield\_data <- df\_trait %>% filter(Trait=="Yield")  
  
# Calculate summary statistics  
yield\_rpd\_summary <- yield\_data %>%  
 group\_by(Algorithm) %>%  
 summarize(  
 mean\_RPD = mean(RPD),  
 sd\_RPD = sd(RPD),  
 n = n(),  
 SEM\_RPD = sd\_RPD / sqrt(n),  
 CI\_Lower = mean\_RPD - 1.96 \* SEM\_RPD,  
 CI\_Upper = mean\_RPD + 1.96 \* SEM\_RPD  
 )  
  
# Print the summary  
print(yield\_rpd\_summary)

## # A tibble: 7 × 7  
## Algorithm mean\_RPD sd\_RPD n SEM\_RPD CI\_Lower CI\_Upper  
## <chr> <dbl> <dbl> <int> <dbl> <dbl> <dbl>  
## 1 ANN 2.42 1.05 3 0.608 1.23 3.61  
## 2 GP 1.81 NA 1 NA NA NA   
## 3 MLP 2.03 NA 1 NA NA NA   
## 4 MLR 1.48 0.449 5 0.201 1.09 1.88  
## 5 PF 1.6 NA 1 NA NA NA   
## 6 RF 1.65 NA 1 NA NA NA   
## 7 XGBoost 1.10 NA 1 NA NA NA

# Combine data for all models  
yield\_meta\_combined <- metagen(  
 TE = yield\_rpd\_summary$mean\_RPD,  
 lower = yield\_rpd\_summary$CI\_Lower,  
 upper = yield\_rpd\_summary$CI\_Upper,  
 studlab = yield\_rpd\_summary$Algorithm,  
 sm = "Mean"  
)  
  
# Forest plot for all models  
# png(file = "output/forestplot\_yield.png", width = 10, height = 5, res = 300, units = "in")  
forest(yield\_meta\_combined,  
 main = "Forest Plot of RPD for All Models for Yield Estimation",  
 xlab = "RPD",  
 label.left = "Models",  
 studlab = yield\_rpd\_summary$Algorithm,  
 print.tau2 = FALSE,  
 col.diamond = "blue",  
 col.predict = "red",  
 leftlabs = c("Models", "Mean", "SE(Mean)"))

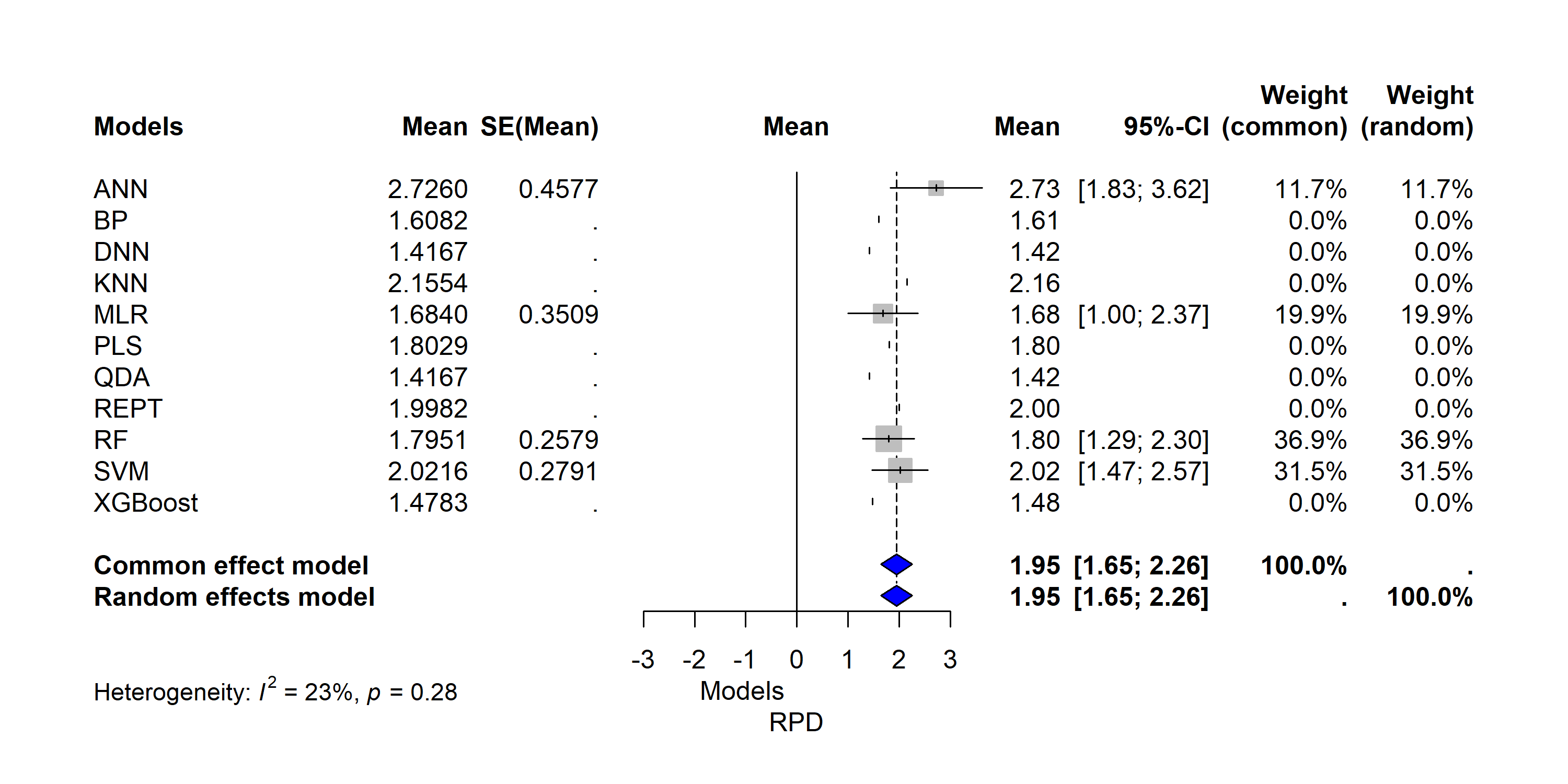


### Forest plot for the nitrogen

# Install and load necessary packages  
# Load necessary libraries  
library(meta)  
  
# Example RPD data  
nitrogen\_data <- df\_trait %>% filter(Trait=="Nitrogen")  
  
# Calculate summary statistics  
nitrogen\_rpd\_summary <- nitrogen\_data %>%  
 group\_by(Algorithm) %>%  
 summarize(  
 mean\_RPD = mean(RPD),  
 sd\_RPD = sd(RPD),  
 n = n(),  
 SEM\_RPD = sd\_RPD / sqrt(n),  
 CI\_Lower = mean\_RPD - 1.96 \* SEM\_RPD,  
 CI\_Upper = mean\_RPD + 1.96 \* SEM\_RPD  
 )  
  
# Print the summary  
print(nitrogen\_rpd\_summary)

## # A tibble: 11 × 7  
## Algorithm mean\_RPD sd\_RPD n SEM\_RPD CI\_Lower CI\_Upper  
## <chr> <dbl> <dbl> <int> <dbl> <dbl> <dbl>  
## 1 ANN 2.73 0.793 3 0.458 1.83 3.62  
## 2 BP 1.61 NA 1 NA NA NA   
## 3 DNN 1.42 NA 1 NA NA NA   
## 4 KNN 2.16 NA 1 NA NA NA   
## 5 MLR 1.68 0.608 3 0.351 0.996 2.37  
## 6 PLS 1.80 NA 1 NA NA NA   
## 7 QDA 1.42 NA 1 NA NA NA   
## 8 REPT 2.00 NA 1 NA NA NA   
## 9 RF 1.80 0.632 6 0.258 1.29 2.30  
## 10 SVM 2.02 0.684 6 0.279 1.47 2.57  
## 11 XGBoost 1.48 NA 1 NA NA NA

# Combine data for all models  
nitrogen\_meta\_combined <- metagen(  
 TE = nitrogen\_rpd\_summary$mean\_RPD,  
 lower = nitrogen\_rpd\_summary$CI\_Lower,  
 upper = nitrogen\_rpd\_summary$CI\_Upper,  
 studlab = nitrogen\_rpd\_summary$Algorithm,  
 sm = "Mean"  
)  
  
# Forest plot for all models  
#png(file = "output/forestplot\_nitrogen.png", width = 10, height = 5, res = 300, units = "in")  
forest(nitrogen\_meta\_combined,  
 main = "Forest Plot of RPD for All Models of nitrogen Estimation",  
 xlab = "RPD",  
 label.left = "Models",  
 studlab = nitrogen\_rpd\_summary$Algorithm,  
 print.tau2 = FALSE,  
 comb.random = FALSE,  
 col.diamond = "blue",  
 col.predict = "red",  
 leftlabs = c("Models", "Mean", "SE(Mean)"))



## Multivariate Linear Regression: Key drivers of ML model performance

# Recode RPD variable to convert to factor predictors  
df\_trait\_model <- df\_trait %>%   
 mutate(  
 RPD\_rec = recode\_factor(Model\_Class,   
 "unrealiable models" = "Bad",  
 "reasonable models" = "Reliable",  
 "excellent models" = "Excellent"),  
 Crop = as\_factor(Crop),  
 Stage = as\_factor(Stage),  
 Trait = as\_factor(Trait),  
 UAV\_Type = as\_factor(UAV\_Type),  
 Sensor = as\_factor(Sensor),  
 Band = as\_factor(Band),  
 Algorithm = as\_factor(Algorithm)  
 )  
df\_trait\_model <- df\_trait\_model %>%   
 select(RPD, Crop, Stage, Trait, UAV\_Type, Sensor, Band, Altitude\_m, Algorithm)

# Drop Na  
df\_trait\_model <- df\_trait\_model %>% drop\_na(UAV\_Type)

### Biomass

# Example data  
biomass\_model\_data <- df\_trait\_model %>% filter(Trait=="AGB")  
  
# Remove trait  
biomass\_model\_data <- biomass\_model\_data %>% select(-Trait)  
  
# Standardize predictor variables  
recipe <- recipe(RPD ~ ., data=biomass\_model\_data) %>%  
 # convert string to factor  
 #step\_string2factor(all\_nominal()) %>%  
 # remove no variance predictors   
 #recipes::step\_nzv(all\_nominal()) %>%  
 # factor to dummy variables  
 #step\_dummy(all\_nominal(), one\_hot=T) %>%  
 step\_lencode\_mixed(all\_nominal\_predictors() , outcome=vars(RPD)) %>%  
 # remove non-variance variables  
 step\_nzv(where(is.numeric)) %>%  
 #step\_dummy(all\_nominal\_predictors(), one\_hot=F) %>% # Convert categorical variables to dummy variables  
 prep()

## boundary (singular) fit: see help('isSingular')

# juice recipe  
biomass\_model\_data\_final <- juice(recipe)  
  
# Fit the multinomial logistic regression model  
biomass\_model <- lm(RPD ~. , data = biomass\_model\_data\_final)

### Yield

# Example data  
yield\_model\_data <- df\_trait\_model %>% filter(Trait=="Yield")  
  
# Remove trait  
yield\_model\_data <- yield\_model\_data %>% select(-Trait)  
  
# Standardize predictor variables  
recipe <- recipe(RPD ~ ., data=yield\_model\_data) %>%  
 # convert string to factor  
 #step\_string2factor(all\_nominal()) %>%  
 # remove no variance predictors   
 #recipes::step\_nzv(all\_nominal()) %>%  
 # factor to dummy variables  
 #step\_dummy(all\_nominal(), one\_hot=T) %>%  
 step\_lencode\_mixed(all\_nominal\_predictors() , outcome=vars(RPD)) %>%  
 # remove non-variance variables  
 step\_nzv(where(is.numeric)) %>%  
 #step\_dummy(all\_nominal\_predictors(), one\_hot=F) %>% # Convert categorical variables to dummy variables  
 prep()

## boundary (singular) fit: see help('isSingular')  
## boundary (singular) fit: see help('isSingular')

# juice recipe  
yield\_model\_data\_final <- juice(recipe)  
  
# Fit the multinomial logistic regression model  
yield\_model <- lm(RPD ~. , data = yield\_model\_data\_final)

### Nitrogen

# Example data  
nitrogen\_model\_data <- df\_trait\_model %>% filter(Trait=="Nitrogen")  
  
# Remove trait  
nitrogen\_model\_data <- nitrogen\_model\_data %>% select(-Trait)  
  
# Standardize predictor variables  
recipe <- recipe(RPD ~ ., data=nitrogen\_model\_data) %>%  
 # convert string to factor  
 #step\_string2factor(all\_nominal()) %>%  
 # remove no variance predictors   
 #recipes::step\_nzv(all\_nominal()) %>%  
 # factor to dummy variables  
 #step\_dummy(all\_nominal(), one\_hot=T) %>%  
 step\_lencode\_mixed(all\_nominal\_predictors() , outcome=vars(RPD)) %>%  
 # remove non-variance variables  
 step\_nzv(where(is.numeric)) %>%  
 #step\_dummy(all\_nominal\_predictors(), one\_hot=F) %>% # Convert categorical variables to dummy variables  
 prep()

## boundary (singular) fit: see help('isSingular')

# juice recipe  
nitrogen\_model\_data\_final <- juice(recipe)  
  
# Fit the multinomial logistic regression model  
nitrogen\_model <- lm(RPD ~. , data = nitrogen\_model\_data\_final)

## Report results

### Biomass

* Model

report\_model(biomass\_model)

## linear model (estimated using OLS) to predict RPD with Crop, Stage, UAV\_Type, Sensor, Band and Altitude\_m (formula: RPD ~ Crop + Stage + UAV\_Type + Sensor + Band + Altitude\_m)

* Performance

report\_performance(biomass\_model)

## The model explains a statistically significant and substantial proportion of  
## variance (R2 = 0.95, F(4, 17) = 79.57, p < .001, adj. R2 = 0.94)

* Parameters

report\_parameters(biomass\_model)

## - The intercept is statistically non-significant and negative (beta = -0.03, 95% CI [-0.67, 0.60], t(17) = -0.11, p = 0.913; Std. beta = -1.12e-16, 95% CI [-0.11, 0.11])  
## - The effect of Crop is statistically significant and negative (beta = -1.10, 95% CI [-2.00, -0.19], t(17) = -2.56, p = 0.020; Std. beta = -0.85, 95% CI [-1.55, -0.15])  
## - The effect of Stage is statistically significant and positive (beta = 1.05, 95% CI [0.74, 1.36], t(17) = 7.14, p < .001; Std. beta = 0.92, 95% CI [0.65, 1.19])  
## - The effect of UAV Type is statistically non-significant and positive (beta = 1.07, 95% CI [-0.20, 2.34], t(17) = 1.78, p = 0.093; Std. beta = 0.91, 95% CI [-0.17, 2.00])  
## - The effect of Sensor is statistically non-significant and negative (beta = -5.72e-03, 95% CI [-0.77, 0.75], t(17) = -0.02, p = 0.988; Std. beta = -4.25e-03, 95% CI [-0.57, 0.56])  
## - The effect of Band is statistically non-significant and negative (beta = -0.03, 95% CI [-0.67, 0.60], t(17) = -0.11, p = 0.913; Std. beta = -1.12e-16, 95% CI [-0.11, 0.11])  
## - The effect of Altitude m is statistically significant and negative (beta = -1.10, 95% CI [-2.00, -0.19], t(17) = -2.56, p = 0.020; Std. beta = -0.85, 95% CI [-1.55, -0.15])

* Summary

library(flextable)

##   
## Attaching package: 'flextable'

## The following objects are masked from 'package:kableExtra':  
##   
## as\_image, footnote

## The following object is masked from 'package:purrr':  
##   
## compose

library(officer)

##   
## Attaching package: 'officer'

## The following object is masked from 'package:readxl':  
##   
## read\_xlsx

#stargazer(biomass\_model, type = "text")  
#sjPlot::tab\_model(biomass\_model, show.p = T, show.ci = T)  
# Create a summary of the model  
biomass\_model\_summary <- summary(biomass\_model)  
  
# Extract coefficients  
biomass\_coefficients <- as.data.frame(biomass\_model\_summary$coefficients)  
  
# Create a beautiful table  
  
# Create a flextable object  
# ft <- flextable(biomass\_coefficients)  
#   
# # Customize the flextable  
# ft <- theme\_vanilla(ft)  
# ft <- autofit(ft)  
# ft <- set\_caption(ft, caption = "Customized Sample Table")  
#   
# # Additional styling  
# ft <- bold(ft, part = "header")  
# ft <- bg(ft, part = "header", bg = "lightblue")  
# ft <- color(ft, part = "header", color = "white")  
# ft <- border\_remove(ft)  
# ft <- border\_outer(ft, border = fp\_border(color = "black", width = 1))  
# ft <- border\_inner\_h(ft, border = fp\_border(color = "gray", width = 0.5))  
# ft <- border\_inner\_v(ft, border = fp\_border(color = "gray", width = 0.5))  
#   
# # Display the flextable  
# ft  
# Create beautiful table  
kable(biomass\_coefficients, format = "simple")

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Estimate** | **Std. Error** | **t value** | **Pr(>|t|)** |
| (Intercept) | -0.03 | 0.30 | -0.11 | 0.9130069 |
| Crop | -1.10 | 0.43 | -2.56 | 0.0204846\* |
| Stage | 1.05 | 0.15 | 7.14 | 0.0000017\*\*\* |
| UAV\_Type | 1.07 | 0.60 | 1.78 | 0.0930132 |
| Sensor | -0.01 | 0.36 | -0.02 | 0.9875179 |
| Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1 | | | | |
| Residual standard error: 0.1409 on 17 degrees of freedom  Multiple R-squared: 0.9493, Adjusted R-squared: 0.9374  F-statistic: 79.57 on 4 and 17 DF, p-value: 8.917e-11 | | | | |

### Yield

* Model

report\_model(yield\_model)

## linear model (estimated using OLS) to predict RPD with Stage, Sensor, Band, Altitude\_m and Algorithm (formula: RPD ~ Stage + Sensor + Band + Altitude\_m + Algorithm)

* Performance

report\_performance(yield\_model)

## The model explains a statistically not significant and substantial proportion  
## of variance (R2 = 0.56, F(4, 8) = 2.53, p = 0.123, adj. R2 = 0.34)

* Parameters

report\_parameters(yield\_model)

## - The intercept is statistically non-significant and negative (beta = -1.24, 95% CI [-13.26, 10.78], t(8) = -0.24, p = 0.818; Std. beta = 4.03e-16, 95% CI [-0.52, 0.52])  
## - The effect of Stage is statistically non-significant and positive (beta = 1.77, 95% CI [-1.28, 4.83], t(8) = 1.34, p = 0.218; Std. beta = 0.59, 95% CI [-0.42, 1.60])  
## - The effect of Sensor is statistically non-significant and negative (beta = -0.78, 95% CI [-6.34, 4.78], t(8) = -0.32, p = 0.755; Std. beta = -0.25, 95% CI [-2.00, 1.51])  
## - The effect of Band is statistically non-significant and negative (beta = -9.25e-03, 95% CI [-0.06, 0.04], t(8) = -0.45, p = 0.662; Std. beta = -0.30, 95% CI [-1.82, 1.22])  
## - The effect of Altitude m is statistically non-significant and positive (beta = 1.11, 95% CI [-2.80, 5.02], t(8) = 0.66, p = 0.531; Std. beta = 0.22, 95% CI [-0.56, 1.01])  
## - The effect of Algorithm is statistically non-significant and negative (beta = -1.24, 95% CI [-13.26, 10.78], t(8) = -0.24, p = 0.818; Std. beta = 4.03e-16, 95% CI [-0.52, 0.52])

* Summary

#stargazer(yield\_model, type = "text")  
#sjPlot::tab\_model(yield\_model, show.p = T, show.ci = T)  
# Create a summary of the model  
yield\_model\_summary <- summary(yield\_model)  
  
# Extract coefficients  
yield\_coefficients <- as.data.frame(yield\_model\_summary$coefficients)  
  
# Create a beautiful table  
kable(yield\_coefficients, format = "simple")

|  | Estimate | Std. Error | t value | Pr(>|t|) |
| --- | --- | --- | --- | --- |
| (Intercept) | -1.2409698 | 5.2109957 | -0.2381445 | 0.8177554 |
| Stage | 1.7717394 | 1.3243760 | 1.3377918 | 0.2177445 |
| Sensor | -0.7797305 | 2.4126842 | -0.3231797 | 0.7548449 |
| Altitude\_m | -0.0092505 | 0.0203727 | -0.4540652 | 0.6618474 |
| Algorithm | 1.1117613 | 1.6962494 | 0.6554232 | 0.5305873 |

### Nitrogen

* Model

report\_model(nitrogen\_model)

## linear model (estimated using OLS) to predict RPD with Crop, Stage, UAV\_Type, Sensor, Band and Altitude\_m (formula: RPD ~ Crop + Stage + UAV\_Type + Sensor + Band + Altitude\_m)

* Performance

report\_performance(nitrogen\_model)

## The model explains a statistically significant and substantial proportion of  
## variance (R2 = 0.71, F(6, 18) = 7.21, p < .001, adj. R2 = 0.61)

* Parameters

report\_parameters(nitrogen\_model)

## - The intercept is statistically non-significant and negative (beta = -0.20, 95% CI [-3.17, 2.77], t(18) = -0.14, p = 0.889; Std. beta = 3.62e-16, 95% CI [-0.26, 0.26])  
## - The effect of Crop is statistically non-significant and negative (beta = -4.55, 95% CI [-55.47, 46.37], t(18) = -0.19, p = 0.853; Std. beta = -1.46, 95% CI [-17.80, 14.88])  
## - The effect of Stage is statistically significant and positive (beta = 1.35, 95% CI [0.76, 1.93], t(18) = 4.85, p < .001; Std. beta = 0.87, 95% CI [0.49, 1.24])  
## - The effect of UAV Type is statistically non-significant and positive (beta = 4.41, 95% CI [-47.18, 56.00], t(18) = 0.18, p = 0.859; Std. beta = 1.56, 95% CI [-16.70, 19.83])  
## - The effect of Sensor is statistically non-significant and positive (beta = 2.94, 95% CI [-31.09, 36.97], t(18) = 0.18, p = 0.858; Std. beta = 1.30, 95% CI [-13.72, 16.31])  
## - The effect of Band is statistically non-significant and negative (beta = -3.04, 95% CI [-38.90, 32.83], t(18) = -0.18, p = 0.861; Std. beta = -1.44, 95% CI [-18.46, 15.58])  
## - The effect of Altitude m is statistically non-significant and negative (beta = -4.25e-04, 95% CI [-7.29e-03, 6.44e-03], t(18) = -0.13, p = 0.898; Std. beta = -0.02, 95% CI [-0.33, 0.29])

* Summary

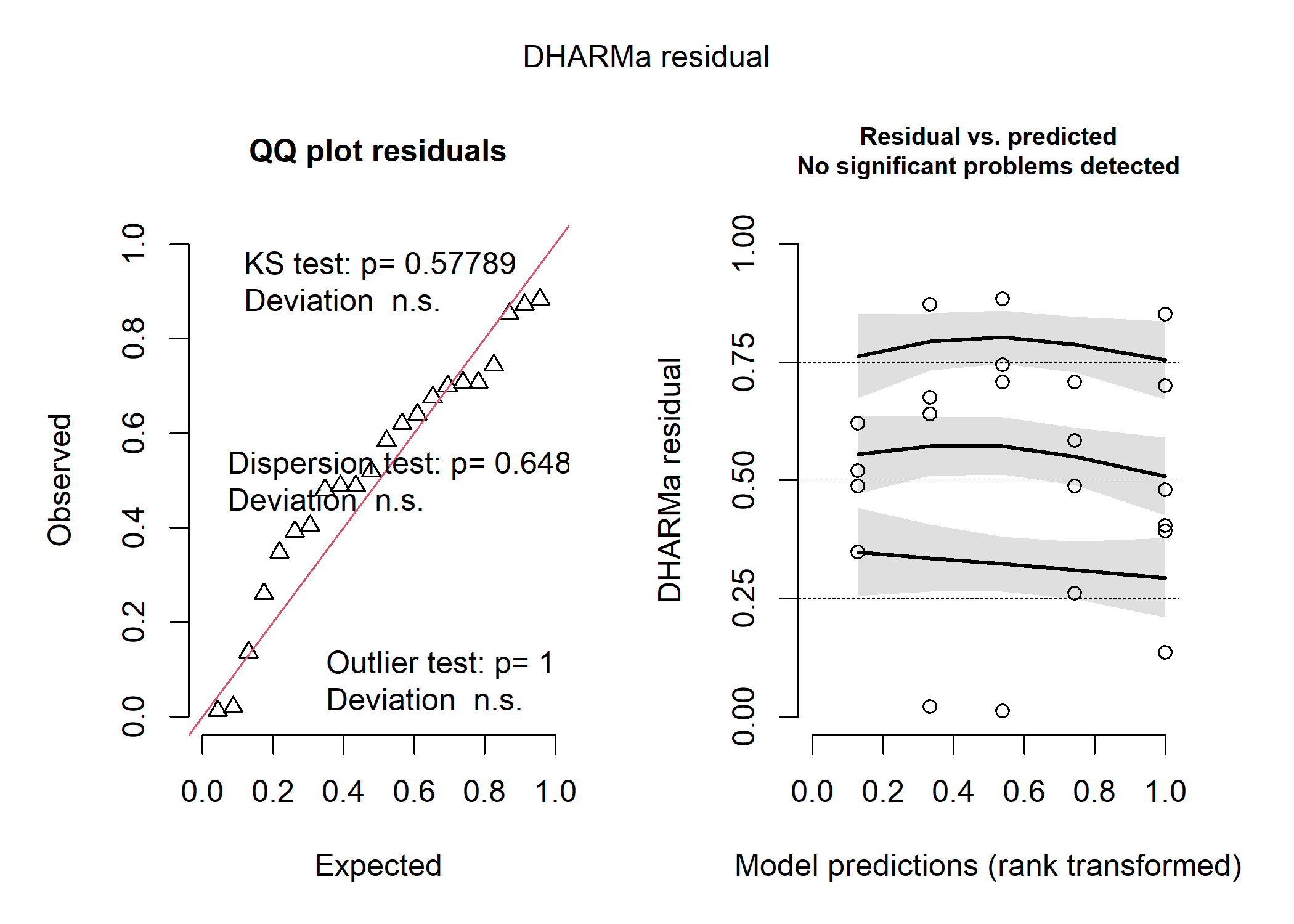
#stargazer(nitrogen\_model, type = "text")  
#sjPlot::tab\_model(nitrogen\_model, show.p = T, show.ci = T)  
# Create a summary of the model  
nitrogen\_model\_summary <- summary(nitrogen\_model)  
  
# Extract coefficients  
nitrogen\_coefficients <- as.data.frame(nitrogen\_model\_summary$coefficients)  
  
# Create a beautiful table  
kable(nitrogen\_coefficients, format = "simple")

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Estimate** | **Std. Error** | **t value** | **Pr(>|t|)** |
| (Intercept) | -0.1995 | 1.4136 | -0.1411 | 0.88934 |
| Crop | -4.5532 | 24.2372 | -0.1879 | 0.85309 |
| Stage | 1.3486 | 0.2779 | 4.8527 | 0.00013 |
| UAV\_Type | 4.4134 | 24.5559 | 0.1797 | 0.85937 |
| Sensor | 2.9388 | 16.1974 | 0.1814 | 0.85805 |
| Band | -3.0355 | 17.0713 | -0.1778 | 0.86086 |
| Altitude\_m | -0.0004 | 0.0033 | -0.1302 | 0.89783 |

## Plot the residuals

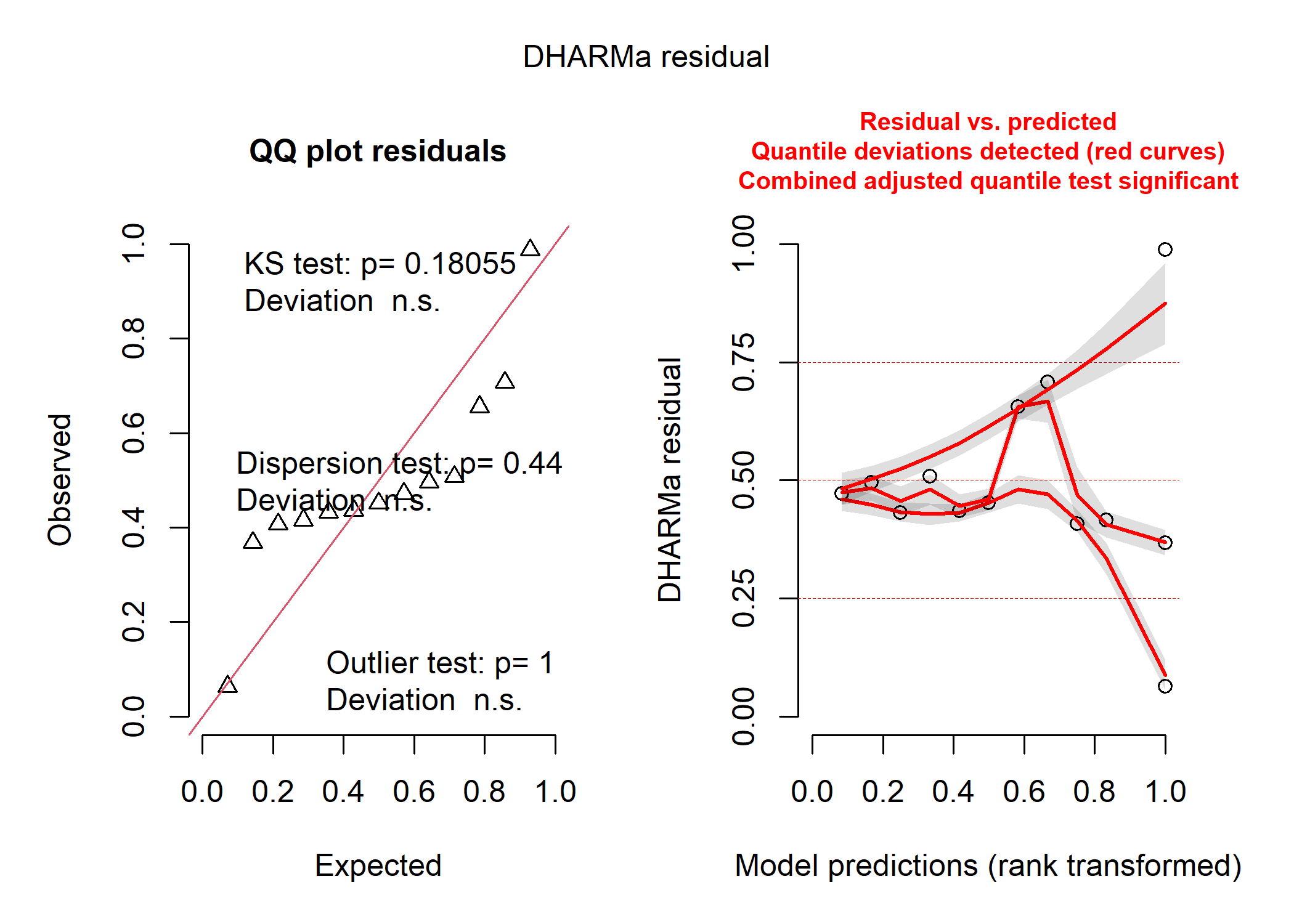
### Biomass

biomass\_simulationOutput <- simulateResiduals(fittedModel = biomass\_model)  
plot(biomass\_simulationOutput)



### Yield

yield\_simulationOutput <- simulateResiduals(fittedModel = yield\_model)  
plot(yield\_simulationOutput)



### Nitrogen

nitrogen\_simulationOutput <- simulateResiduals(fittedModel = nitrogen\_model)  
plot(nitrogen\_simulationOutput)

