Analysis

Adama NDOUR

2024-07-16

Load package

```r
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
```

Load the data

```r
df <- read_excel("uav_review_data.xlsx")
```

Overview of the data

```r
#str(df)
```

Data manipulation: create a model class variable

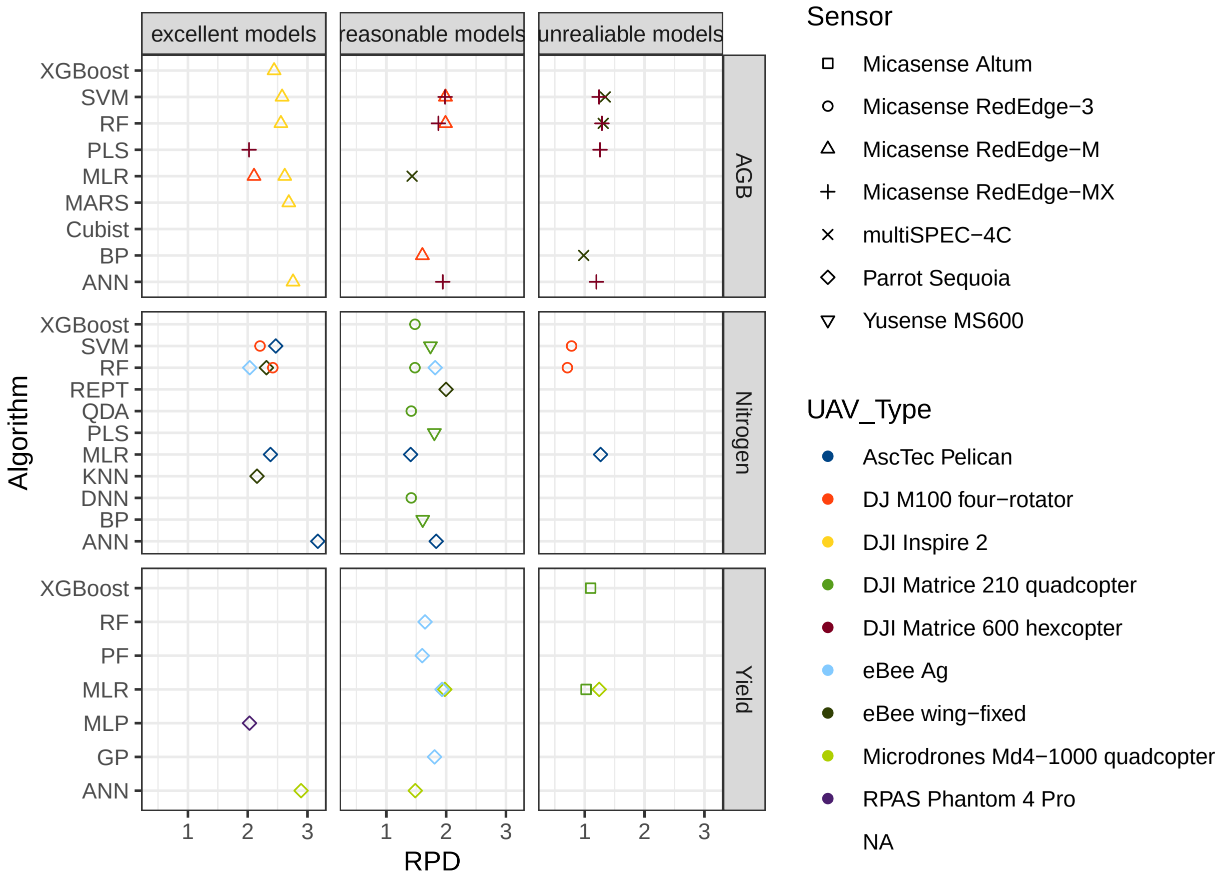
```r
df <- df %>% mutate(
  Model_Class = case_when(
    RPD < 1.4 ~ "unrealiable models",
    RPD >= 1.4 & RPD < 2 ~ "reasonable models",
    RPD >=2 ~ "excellent models"
  )
)

# 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
```

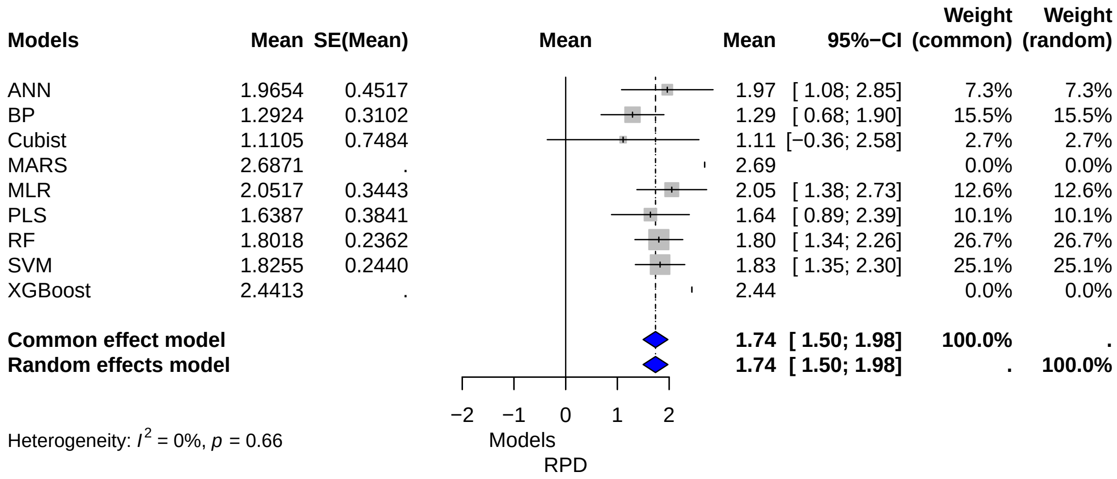


Forest plot for the biomass

```
# 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 x 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)"))
```



Models	Mean	SE(Mean)	Mean	95%−CI	Weight (common)	Weight (random)
ANN	1.9654	0.4517	1.97	[1.08; 2.85]	7.3%	7.3%
BP	1.2924	0.3102	1.29	[0.68; 1.90]	15.5%	15.5%
Cubist	1.1105	0.7484	1.11	[−0.36; 2.58]	2.7%	2.7%
MARS	2.6871	.	2.69		0.0%	0.0%
MLR	2.0517	0.3443	2.05	[1.38; 2.73]	12.6%	12.6%
PLS	1.6387	0.3841	1.64	[0.89; 2.39]	10.1%	10.1%
RF	1.8018	0.2362	1.80	[1.34; 2.26]	26.7%	26.7%
SVM	1.8255	0.2440	1.83	[1.35; 2.30]	25.1%	25.1%
XGBoost	2.4413	.	2.44		0.0%	0.0%
Common effect model			**1.74**	**[1.50; 1.98]**	**100.0%**	**.**
Random effects model			**1.74**	**[1.50; 1.98]**	**.**	**100.0%**

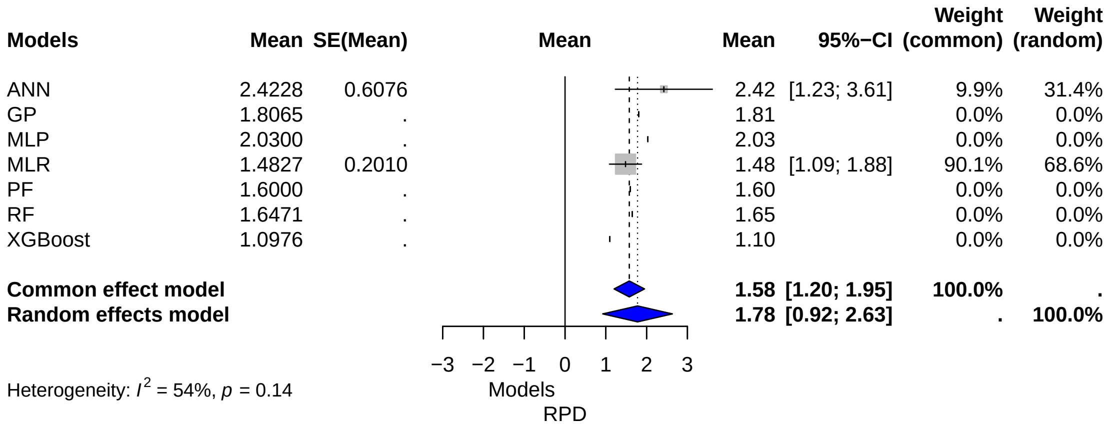
Heterogeneity: I^2 = 0%, p = 0.66

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 x 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)"))
```



Models	Mean	SE(Mean)	Mean	95%−CI	Weight (common)	Weight (random)
ANN	2.4228	0.6076	2.42	[1.23; 3.61]	9.9%	31.4%
GP	1.8065	.	1.81		0.0%	0.0%
MLP	2.0300	.	2.03		0.0%	0.0%
MLR	1.4827	0.2010	1.48	[1.09; 1.88]	90.1%	68.6%
PF	1.6000	.	1.60		0.0%	0.0%
RF	1.6471	.	1.65		0.0%	0.0%
XGBoost	1.0976	.	1.10		0.0%	0.0%
Common effect model			**1.58**	**[1.20; 1.95]**	**100.0%**	**.**
Random effects model			**1.78**	**[0.92; 2.63]**	**.**	**100.0%**

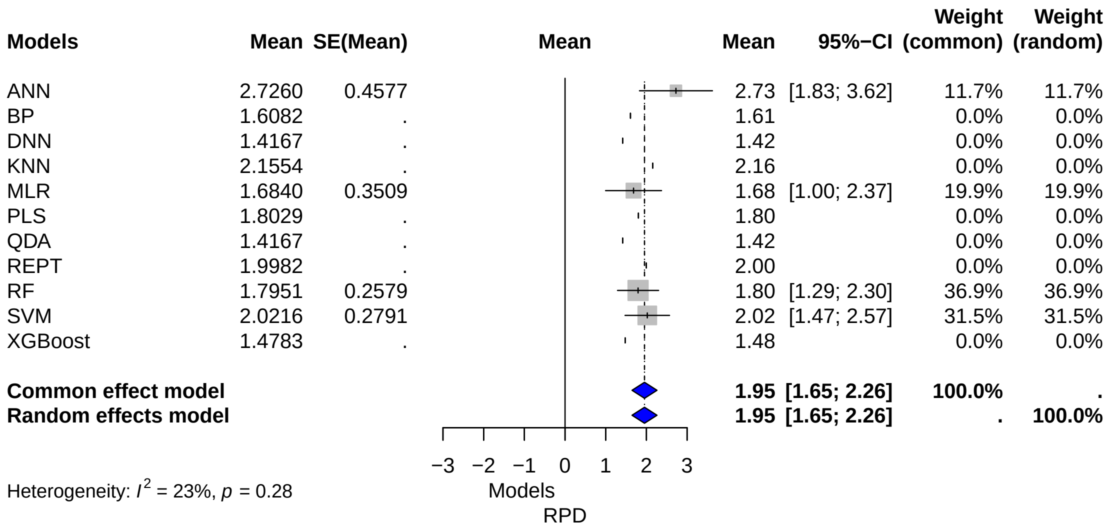
Heterogeneity: I^2 = 54%, p = 0.14

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 x 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
```

```r
# 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)"))
```



Models	Mean	SE(Mean)	Mean	95%−CI	Weight (common)	Weight (random)
ANN	2.7260	0.4577	2.73	[1.83; 3.62]	11.7%	11.7%
BP	1.6082	.	1.61		0.0%	0.0%
DNN	1.4167	.	1.42		0.0%	0.0%
KNN	2.1554	.	2.16		0.0%	0.0%
MLR	1.6840	0.3509	1.68	[1.00; 2.37]	19.9%	19.9%
PLS	1.8029	.	1.80		0.0%	0.0%
QDA	1.4167	.	1.42		0.0%	0.0%
REPT	1.9982	.	2.00		0.0%	0.0%
RF	1.7951	0.2579	1.80	[1.29; 2.30]	36.9%	36.9%
SVM	2.0216	0.2791	2.02	[1.47; 2.57]	31.5%	31.5%
XGBoost	1.4783	.	1.48		0.0%	0.0%
Common effect model			**1.95**	**[1.65; 2.26]**	**100.0%**	**.**
Random effects model			**1.95**	**[1.65; 2.26]**	**.**	**100.0%**

Heterogeneity: I^2 = 23%, p = 0.28

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 va
    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 va
    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 va
    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 Altitu
```

- 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],
##   - The effect of Crop is statistically significant and negative (beta = -1.10, 95% CI [-2.00, -0.19]
##   - The effect of Stage is statistically significant and positive (beta = 1.05, 95% CI [0.74, 1.36],
##   - The effect of UAV Type is statistically non-significant and positive (beta = 1.07, 95% CI [-0.20
##   - The effect of Sensor is statistically non-significant and negative (beta = -5.72e-03, 95% CI [-0
##   - The effect of Band is statistically non-significant and negative (beta = -0.03, 95% CI [-0.67, 0
##   - The effect of Altitude m is statistically significant and negative (beta = -1.10, 95% CI [-2.00,
```

- Summary

```
#stargazer(biomass_model, type = "text")
sjPlot::tab_model(biomass_model, show.p = TRUE, show.ci = FALSE)
```

RPD

Predictors

Estimates

p

(Intercept)

-0.03

0.913

Crop

-1.10

0.020

Stage

1.05

<0.001

UAV Type

1.07

0.093

Sensor

-0.01

0.988

Observations

22

R2 / R2 adjusted

0.949 / 0.937

Yield

- Model

```
report_model(yield_model)
```

```
## linear model (estimated using OLS) to predict RPD with Stage, Sensor, Band, Altitude_m and Algorithm
```

- Performance

```r
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

```r
report_parameters(yield_model)
```

```
##   - The intercept is statistically non-significant and negative (beta = -1.24, 95% CI [-13.26, 10.78]
##   - The effect of Stage is statistically non-significant and positive (beta = 1.77, 95% CI [-1.28, 4
##   - The effect of Sensor is statistically non-significant and negative (beta = -0.78, 95% CI [-6.34,
##   - The effect of Band is statistically non-significant and negative (beta = -9.25e-03, 95% CI [-0.0
##   - The effect of Altitude m is statistically non-significant and positive (beta = 1.11, 95% CI [-2.
##   - The effect of Algorithm is statistically non-significant and negative (beta = -1.24, 95% CI [-13
```

- Summary

```r
report_table(yield_model)
```

```
## Parameter   | Coefficient |          95% CI |  t(8) |     p | Std. Coef. | Std. Coef. 95% CI |   Fit
## ----------------------------------------------------------------------------------------------------
## (Intercept) |       -1.24 | [-13.26, 10.78] | -0.24 | 0.818 |   4.03e-16 |     [-0.52, 0.52] |
## Stage       |        1.77 | [ -1.28,  4.83] |  1.34 | 0.218 |       0.59 |     [-0.42, 1.60] |
## Sensor      |       -0.78 | [ -6.34,  4.78] | -0.32 | 0.755 |      -0.25 |     [-2.00, 1.51] |
## Altitude m  |   -9.25e-03 | [ -0.06,  0.04] | -0.45 | 0.662 |      -0.30 |     [-1.82, 1.22] |
## Algorithm   |        1.11 | [ -2.80,  5.02] |  0.66 | 0.531 |       0.22 |     [-0.56, 1.01] |
##             |             |                 |       |       |            |                   |
## AIC         |             |                 |       |       |            |                   | 26.55
## AICc        |             |                 |       |       |            |                   | 40.55
## BIC         |             |                 |       |       |            |                   | 29.94
## R2          |             |                 |       |       |            |                   |  0.56
## R2 (adj.)   |             |                 |       |       |            |                   |  0.34
## Sigma       |             |                 |       |       |            |                   |  0.54
```

Nitrogen

- Model

```r
report_model(nitrogen_model)
```

```
## linear model (estimated using OLS) to predict RPD with Crop, Stage, UAV_Type, Sensor, Band and Altitu
```

- Performance

```r
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

```r
report_parameters(nitrogen_model)
```

```
##   - The intercept is statistically non-significant and negative (beta = -0.20, 95% CI [-3.17, 2.77],
##   - The effect of Crop is statistically non-significant and negative (beta = -4.55, 95% CI [-55.47, 4
##   - The effect of Stage is statistically significant and positive (beta = 1.35, 95% CI [0.76, 1.93],
##   - The effect of UAV Type is statistically non-significant and positive (beta = 4.41, 95% CI [-47.18
##   - The effect of Sensor is statistically non-significant and positive (beta = 2.94, 95% CI [-31.09,
##   - The effect of Band is statistically non-significant and negative (beta = -3.04, 95% CI [-38.90, 3
##   - The effect of Altitude m is statistically non-significant and negative (beta = -4.25e-04, 95% CI
```

- Summary

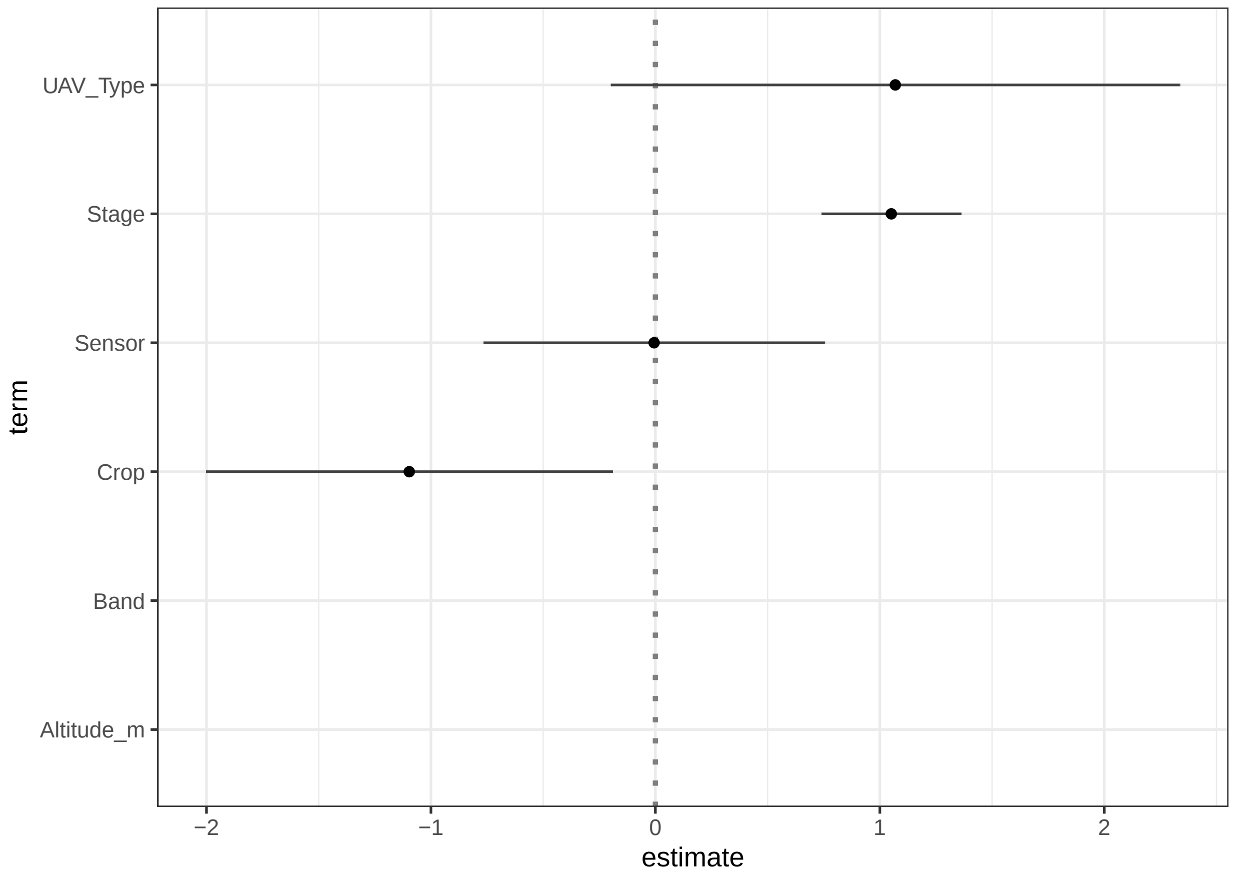
```r
report_table(nitrogen_model)
```

```
## Parameter   | Coefficient |          95% CI | t(18) |      p | Std. Coef. | Std. Coef. 95% CI |   Fi
## -----------------------------------------------------------------------------------------------------
## (Intercept) |       -0.20 | [ -3.17,  2.77] | -0.14 | 0.889  |   3.62e-16 |   [ -0.26,  0.26] |
## Crop        |       -4.55 | [-55.47, 46.37] | -0.19 | 0.853  |      -1.46 |   [-17.80, 14.88] |
## Stage       |        1.35 | [  0.76,  1.93] |  4.85 | < .001 |       0.87 |   [  0.49,  1.24] |
## UAV Type    |        4.41 | [-47.18, 56.00] |  0.18 | 0.859  |       1.56 |   [-16.70, 19.83] |
## Sensor      |        2.94 | [-31.09, 36.97] |  0.18 | 0.858  |       1.30 |   [-13.72, 16.31] |
## Band        |       -3.04 | [-38.90, 32.83] | -0.18 | 0.861  |      -1.44 |   [-18.46, 15.58] |
## Altitude m  |   -4.25e-04 | [ -0.01,  0.01] | -0.13 | 0.898  |      -0.02 |   [ -0.33,  0.29] |
##             |             |                 |       |        |            |                   |
## AIC         |             |                 |       |        |            |                   | 32.0
## AICc        |             |                 |       |        |            |                   | 41.0
## BIC         |             |                 |       |        |            |                   | 41.8
## R2          |             |                 |       |        |            |                   |  0.7
## R2 (adj.)   |             |                 |       |        |            |                   |  0.6
## Sigma       |             |                 |       |        |            |                   |  0.3
```

Plot the multinomial regression coefficients

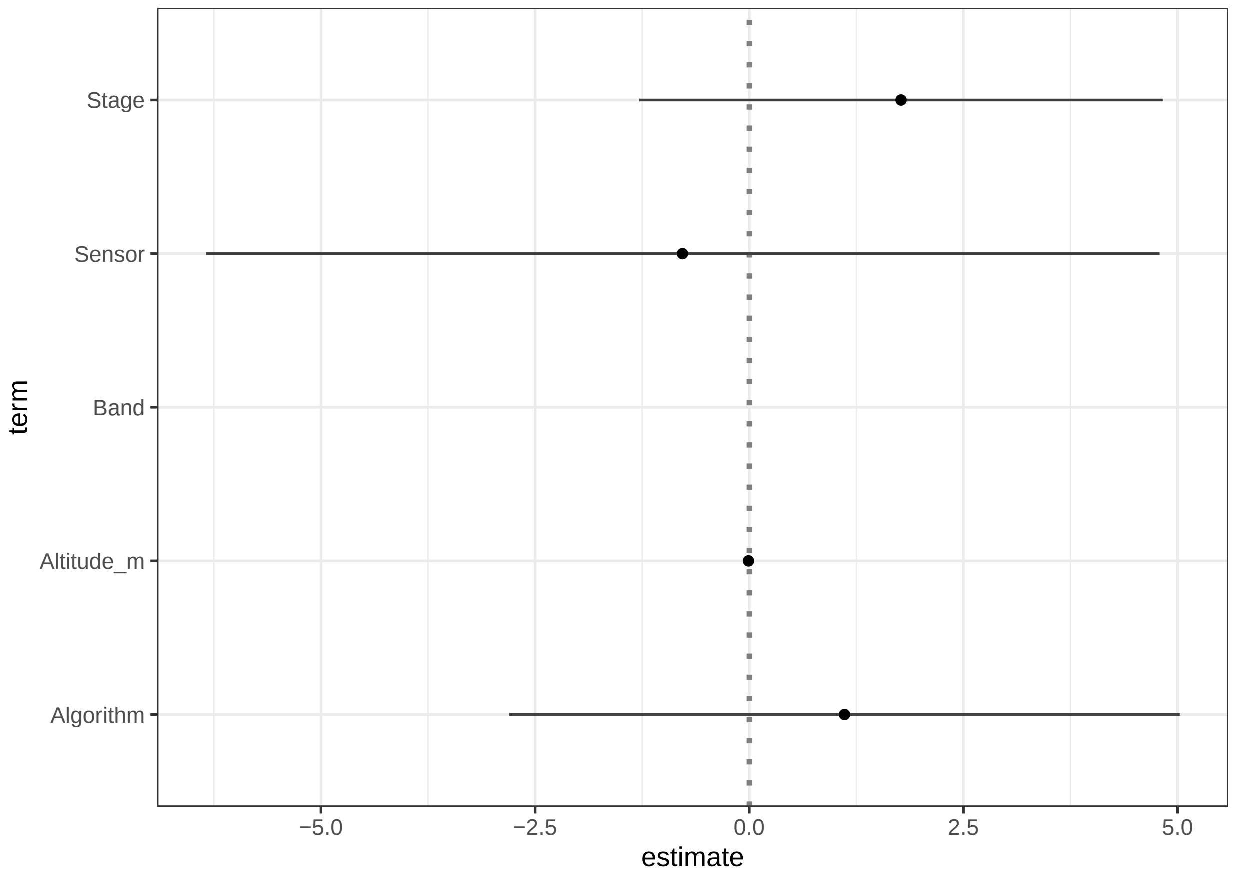
Biomass

```r
ggcoef(
  biomass_model,
  variable_labels = c(
    Crop = "Crop",
    Stage = "Phenological Stage",
    UAV_type = "UAV Type",
    Sensor = "Sensor",
    Band = "Bands",
    Altitude = "Flight Height",
    Algorithm = "ML Algorithm"
  ),
  show_p_values = T,
  signif_stars = T,
  exclude_intercept = TRUE
) + theme_bw()
```



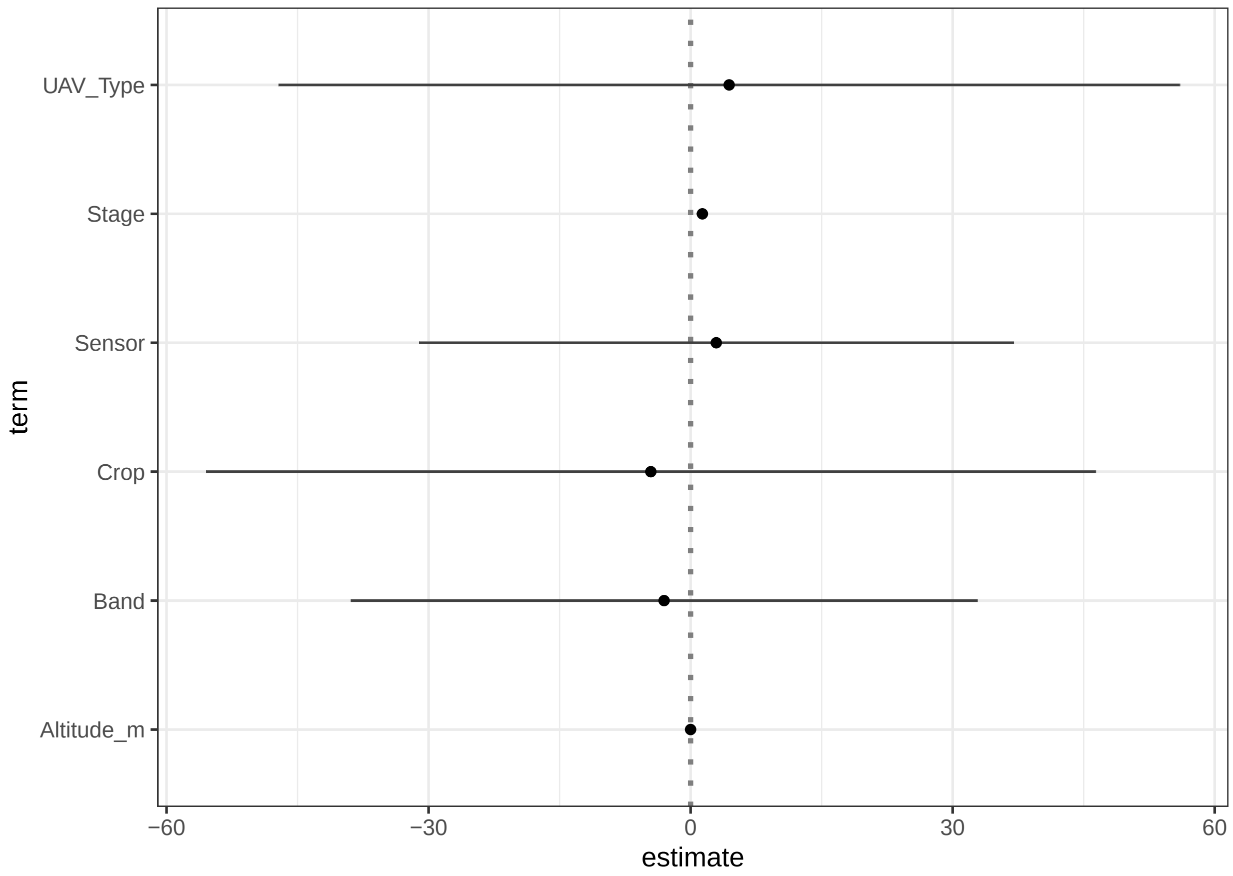
Yield

```
ggcoef(
  yield_model,
  variable_labels = c(
    Crop = "Crop",
    Stage = "Phenological Stage",
    UAV_type = "UAV Type",
    Sensor = "Sensor",
    Band = "Bands",
    Altitude = "Flight Height",
    Algorithm = "ML Algorithm"
  ),
  show_p_values = T,
  signif_stars = T,
  exclude_intercept = TRUE
) + theme_bw()
```



Nitrogen

```
ggcoef(
  nitrogen_model,
  variable_labels = c(
    Crop = "Crop",
    Stage = "Phenological Stage",
    UAV_type = "UAV Type",
    Sensor = "Sensor",
    Band = "Bands",
    Altitude = "Flight Height",
    Algorithm = "ML Algorithm"
  ),
  show_p_values = T,
  signif_stars = T,
  exclude_intercept = TRUE
) + theme_bw()
```


UAV_Type
Stage
Sensor
Crop
Band
Altitude_m
term
−60
−30
0
30
60
estimate