

Basal area by size class

Tim White

Packages, data, and functions

```
library(car)
library(tidyverse)

data_stems <- read_csv("data/data_stems.csv") %>%
  mutate(plot_id = as.factor(plot_id),
        size_class = as.factor(size_class),
        harvested = as.factor(harvested),
        vegetation_type = as.factor(vegetation_type),
        milpa = as.factor(milpa))
data_plots <- read_csv("data/data_plots.csv") %>%
  mutate(plot_id = as.factor(plot_id),
        harvested = as.factor(harvested),
        vegetation_type = as.factor(vegetation_type),
        milpa = as.factor(milpa))

compute_bootstrap_interval <- function(var, statistic = mean, num_iters = 50000,
                                         percentiles = c(0.025, 0.975)) {
  boot_samples <- numeric(num_iters)

  for (i in 1:num_iters) {
    boot_samples[i] <- statistic(sample(var, size = length(var), replace = TRUE))
  }

  return(quantile(boot_samples, percentiles))
}

plot_theme <- theme(plot.background = element_rect(fill = "white"),
                     plot.title = element_blank(),
                     plot.subtitle = element_text(family="sans", face="plain"),
                     axis.title.x = element_text(family="sans", face="bold"),
                     axis.title.y = element_text(family="sans", face="bold"),
                     axis.text.x = element_text(family="sans", face="plain"),
                     axis.text.y = element_text(family="sans", face="plain"),
                     panel.background = element_rect(fill="white"),
                     panel.grid.major.x = element_blank(),
                     panel.grid.major.y = element_line(color="gainsboro"),
                     panel.grid.minor = element_blank(),
                     axis.ticks = element_blank(),
                     legend.background = element_rect(color="black", fill = "white"),
                     legend.position = c(0.9, 0.85),
```

```

    legend.title = element_text(color = "black", face = "bold", hjust = 0.5),
    legend.text = element_text(color = "black"))

```

Figures

Average basal area by size class

Compute bootstrap intervals and organize data

```

harvested <- data_plots %>%
  filter(harvested == "yes") %>%
  select(ba_saplings, starts_with("ba_trees"))
unharvested <- data_plots %>%
  filter(harvested == "no") %>%
  select(ba_saplings, starts_with("ba_trees"))

set.seed(2)
boot_intervals <- bind_rows(compute_bootstrap_interval(unharvested$ba_saplings),
                            compute_bootstrap_interval(unharvested$ba_trees05to09),
                            compute_bootstrap_interval(unharvested$ba_trees10to14),
                            compute_bootstrap_interval(unharvested$ba_trees15plus),
                            compute_bootstrap_interval(harvested$ba_saplings),
                            compute_bootstrap_interval(harvested$ba_trees05to09),
                            compute_bootstrap_interval(harvested$ba_trees10to14),
                            compute_bootstrap_interval(harvested$ba_trees15plus))

ba_mean_data <- data_plots %>%
  select(harvested, ba_saplings, starts_with("ba_trees")) %>%
  group_by(harvested) %>%
  summarize_all(mean) %>%
  pivot_longer(cols = starts_with("ba"), names_prefix = "ba_",
               names_to = "size_class", values_to = "mean") %>%
  bind_cols(boot_intervals) %>%
  select(harvested, size_class, lower = `2.5%`, mean, upper = `97.5%`) %>%
  mutate(size_class = as.factor(size_class)) %>%
  mutate(harvested = fct_recode(harvested,
                                "Harvested" = "yes", "Unharvested" = "no")) %>%
  mutate(size_class = fct_recode(size_class,
                                 "Saplings (0-4 cm DBH)" = "saplings",
                                 "Trees (5-9 cm DBH)" = "trees05to09",
                                 "Trees (10-14 cm DBH)" = "trees10to14",
                                 "Trees (15+ cm DBH)" = "trees15plus"))

```

Pointrange plot

```

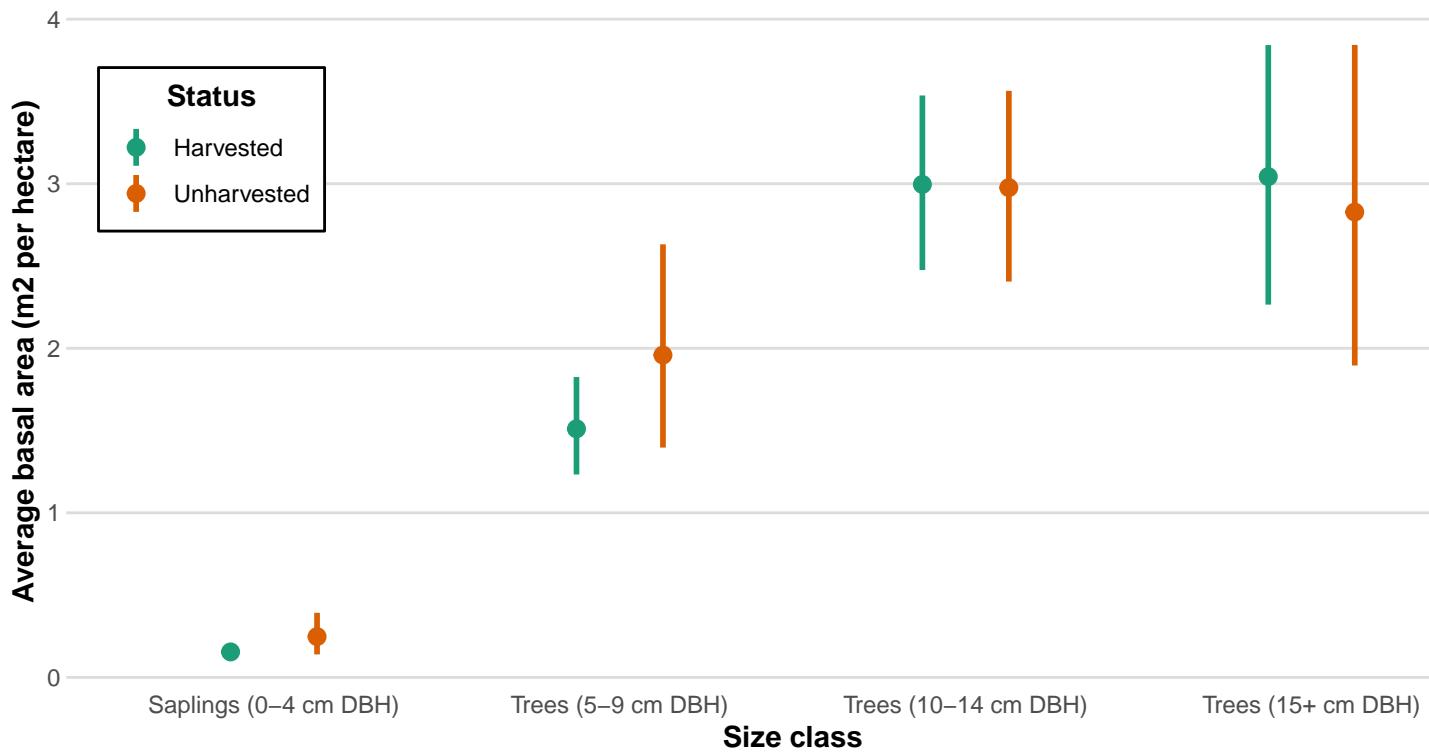
ba_sizeclass_fig1 <- ba_mean_data %>%
  ggplot() +
  geom_pointrange(aes(x = size_class, col = fct_relevel(harvested, "Harvested", "Unharvested")),

```

```

      ymin = lower, y = mean, ymax = upper),
      position = position_dodge(width = 0.5), size = 0.5, linewidth = 1) +
plot_theme +
theme(legend.position = c(0.1, 0.8)) +
scale_color_brewer(palette = "Dark2") +
labs(x = "Size class", y = "Average basal area (m2 per hectare)") +
guides(col=guide_legend(title = "Status"))
ggsave("figures/not_in_manuscript/basal_area_sizeclass_pointrange.png", height = 4, width = 8)
ba_sizeclass_fig1

```

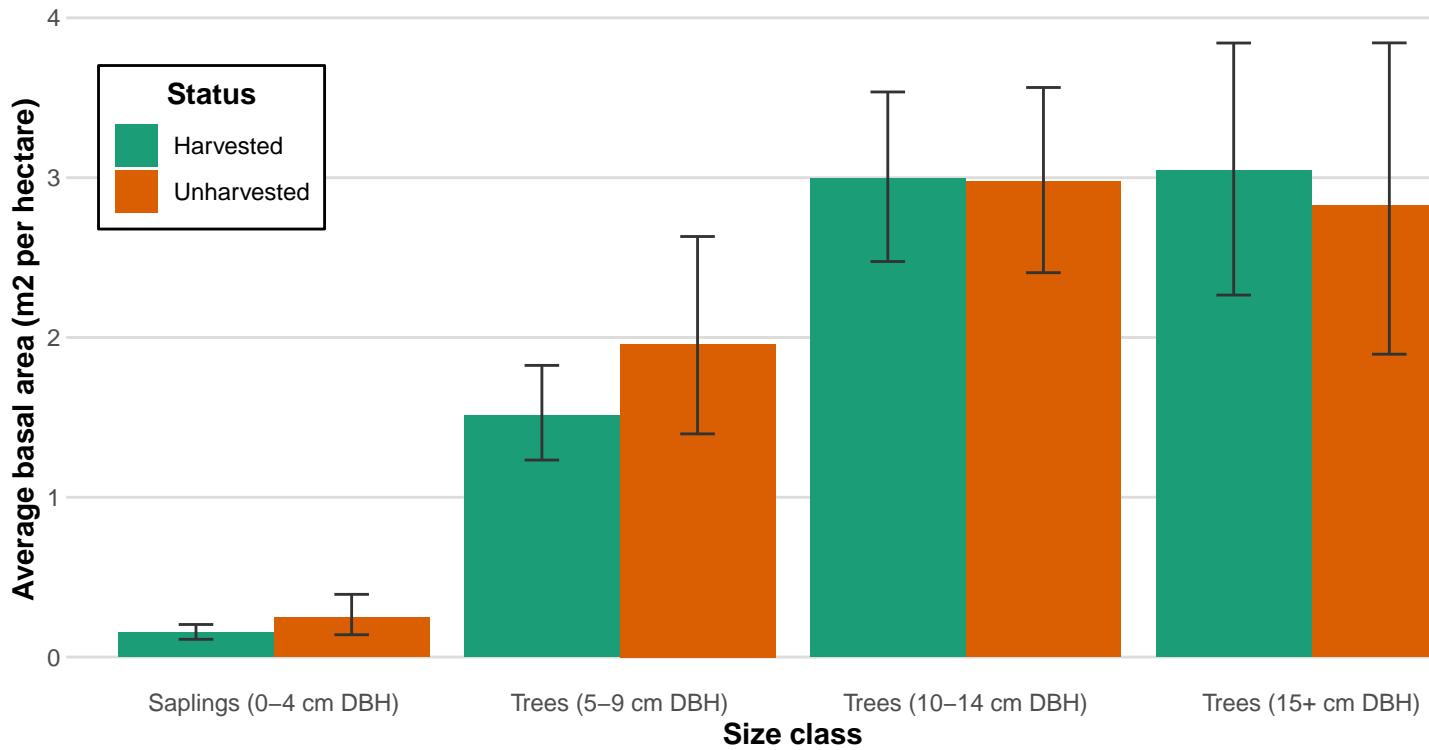


Bar plot

```

ba_sizeclass_fig2 <- ba_mean_data %>%
  ggplot(aes(x = size_class,
             fill = fct_relevel(harvested, "Harvested", "Unharvested"))) +
  geom_col(aes(y = mean),
            position = "dodge") +
  geom_errorbar(aes(x = size_class, ymin = lower, ymax = upper), col = "gray20",
                position = position_dodge(width = 0.9), width = 0.2, show.legend = FALSE) +
  plot_theme +
  theme(legend.position = c(0.1, 0.8)) +
  scale_fill_brewer(palette = "Dark2") +
  labs(x = "Size class", y = "Average basal area (m2 per hectare)") +
  guides(fill=guide_legend(title = "Status"))
ggsave("figures/basal_area_sizeclass.png", height = 4, width = 8)
ba_sizeclass_fig2

```



Distribution of basal area by size class

Organize data

```
ba_dist_data <- data_plots %>%
  select(harvested, ba_saplings, starts_with("ba_trees")) %>%
  pivot_longer(cols = starts_with("ba"),
               names_prefix = "ba_",
               names_to = "size_class",
               values_to = "ba") %>%
  mutate(size_class = as.factor(size_class)) %>%
  mutate(harvested = fct_recode(harvested,
                                "Harvested" = "yes",
                                "Unharvested" = "no")) %>%
  mutate(size_class = fct_recode(size_class,
                                 "Saplings (0-4 cm DBH)" = "saplings",
                                 "Trees (5-9 cm DBH)" = "trees05to09",
                                 "Trees (10-14 cm DBH)" = "trees10to14",
                                 "Trees (15+ cm DBH)" = "trees15plus"))
```

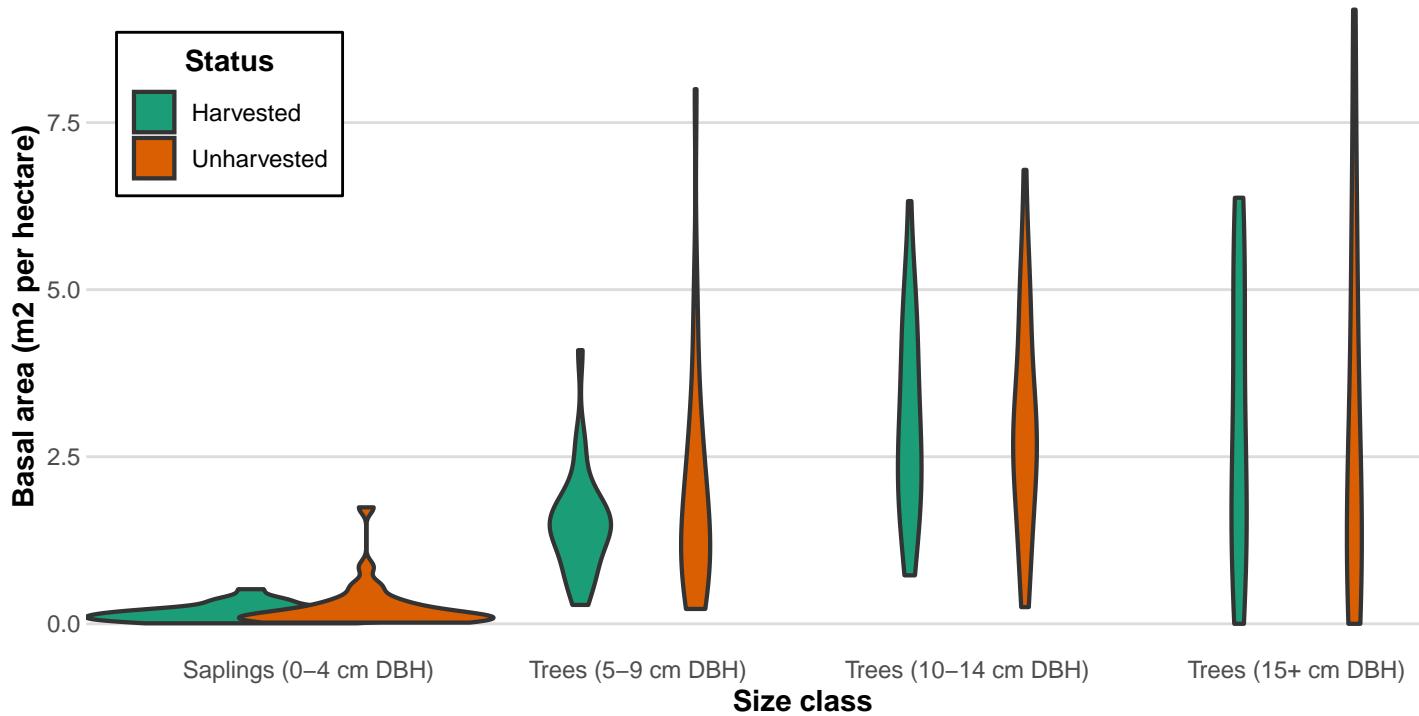
Violin plot

```
ba_dist_data %>%
  ggplot() +
  geom_violin(aes(x = size_class, y = ba,
```

```

      fill = fct_relevel(harvested, "Harvested", "Unharvested"),
      scale = "area", width = 2, position = position_dodge(width = 0.7),
      lwd = 0.75, kernel = "gaussian", adjust = 1.25) +
plot_theme +
theme(legend.position = c(0.1, 0.8)) +
scale_fill_brewer(palette = "Dark2") +
labs(x = "Size class", y = "Basal area (m2 per hectare)") +
guides(fill = guide_legend(title = "Status"))

```

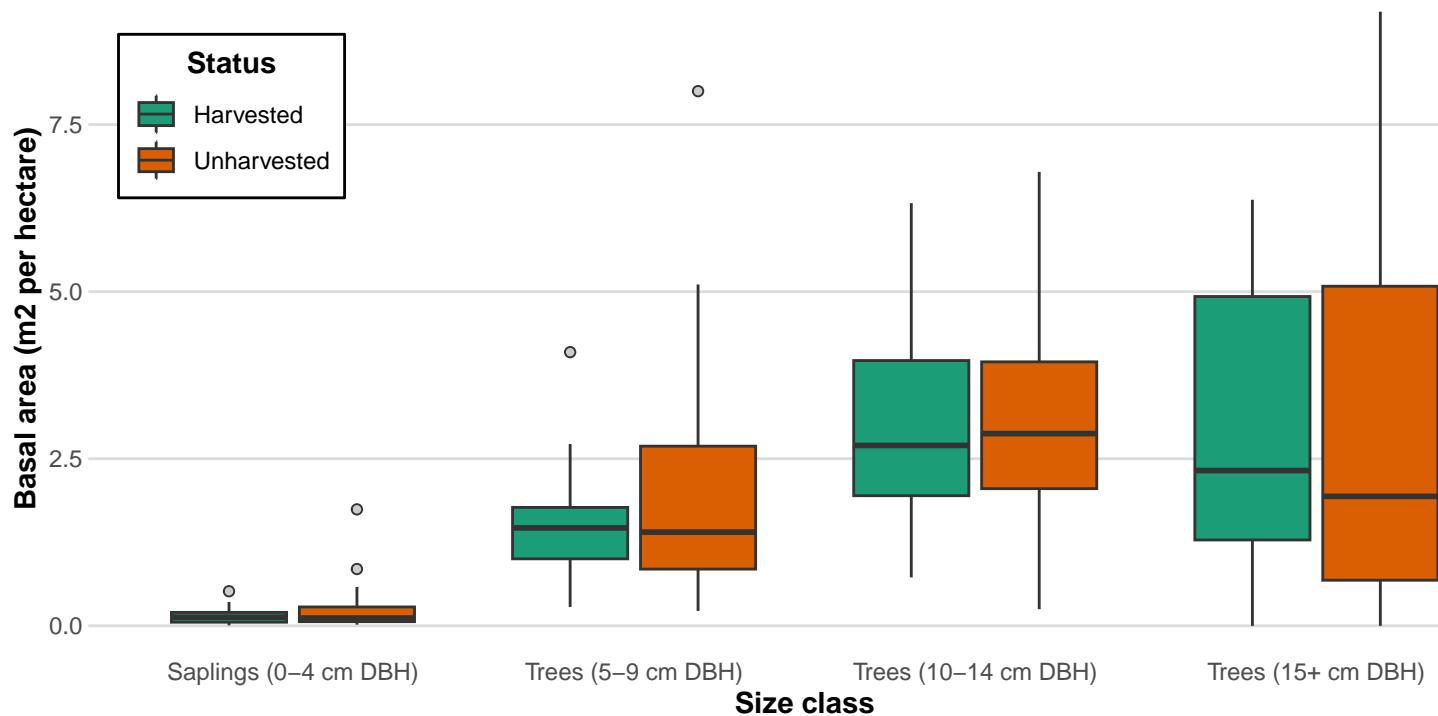


Box plot

```

ba_sizeclass_fig3 <- ba_dist_data %>%
ggplot() +
geom_boxplot(aes(x = size_class, y = ba,
fill = fct_relevel(harvested, "Harvested", "Unharvested")),
outlier.shape = 21, outlier.color = "gray20",
outlier.fill = "gray80") +
plot_theme +
theme(legend.position = c(0.1, 0.8)) +
scale_fill_brewer(palette = "Dark2") +
labs(x = "Size class", y = "Basal area (m2 per hectare)") +
guides(fill = guide_legend(title = "Status"))
ggsave("figures/not_in_manuscript/basal_area_sizeclass_boxplot.png", height = 4, width = 8)
ba_sizeclass_fig3

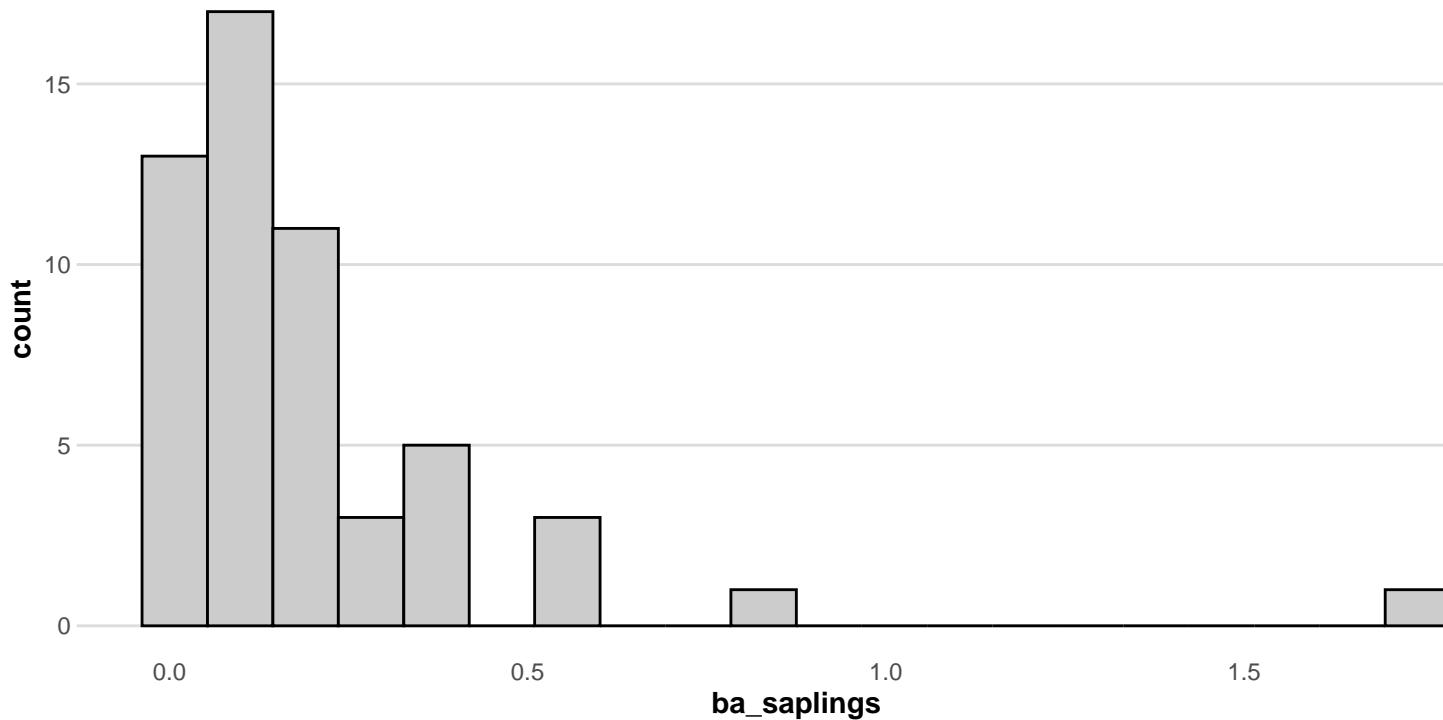
```



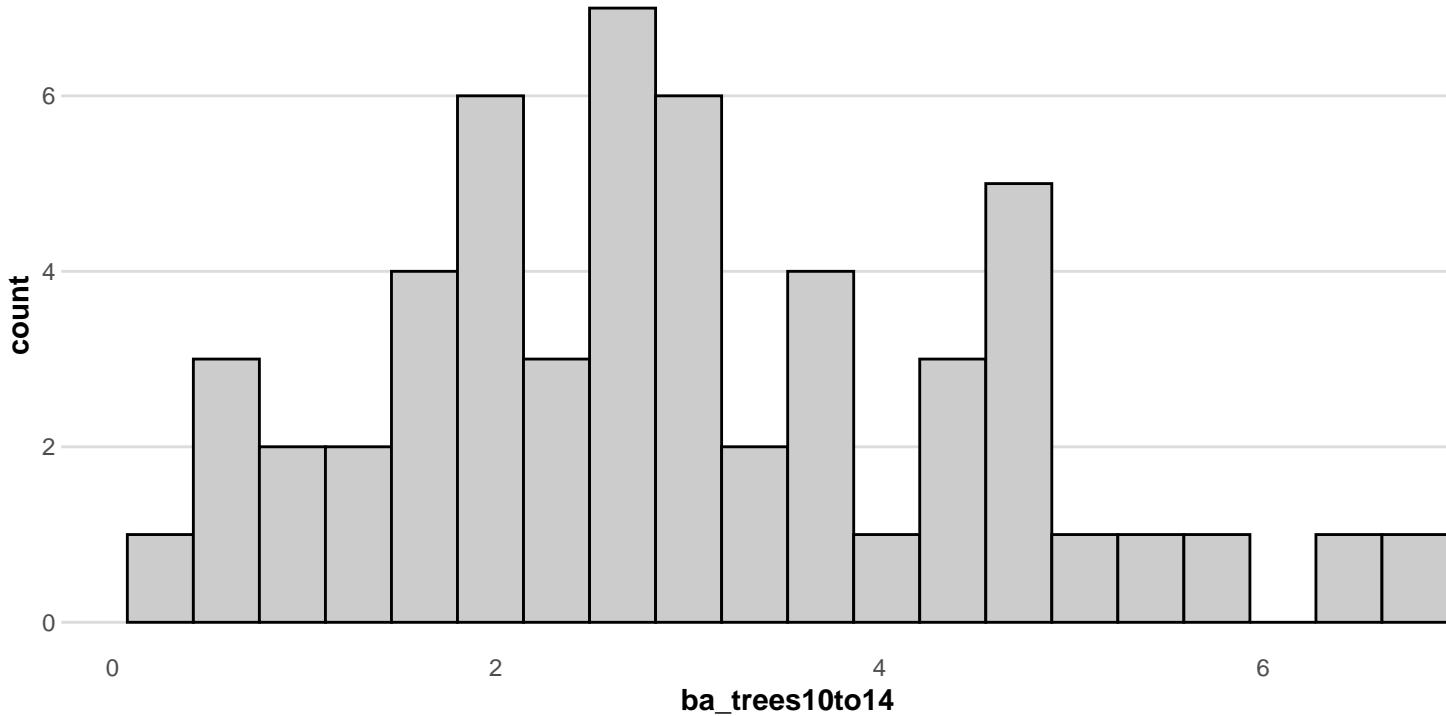
Basal area by size class for different covariates

Histograms, harvesting status

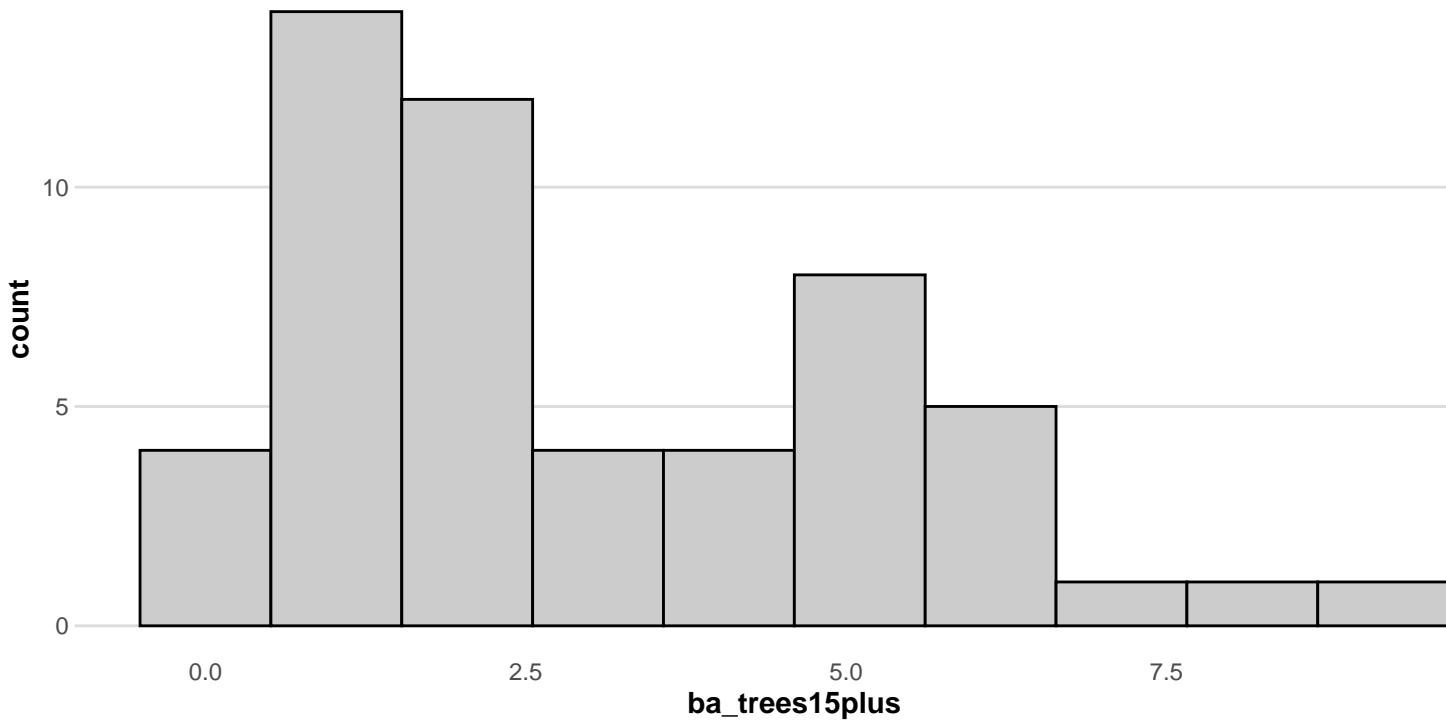
```
data_plots %>%
  ggplot() + geom_histogram(aes(x = ba_saplings), bins = 20,
                            col = "black", fill = "gray80") + plot_theme
```



```
data_plots %>%
  ggplot() + geom_histogram(aes(x = ba_trees10to14), bins = 20,
                            col = "black", fill = "gray80") + plot_theme
```



```
data_plots %>%
  ggplot() + geom_histogram(aes(x = ba_trees15plus), bins = 10,
                            col = "black", fill = "gray80") + plot_theme
```



Five-number summaries, harvesting status

```
summary(data_plots$ba_saplings)

##      Min. 1st Qu. Median     Mean 3rd Qu.    Max.
## 0.0075  0.0575  0.1200  0.2029  0.2169  1.7425

data_plots %>% filter(harvested == "yes") %>% pull(ba_saplings) %>% summary()

##      Min. 1st Qu. Median     Mean 3rd Qu.    Max.
## 0.00750 0.05437 0.12875 0.15471 0.20125 0.51750

data_plots %>% filter(harvested == "no") %>% pull(ba_saplings) %>% summary()

##      Min. 1st Qu. Median     Mean 3rd Qu.    Max.
## 0.01812 0.06250 0.11625 0.24770 0.28281 1.74250

summary(data_plots$ba_trees05to09)

##      Min. 1st Qu. Median     Mean 3rd Qu.    Max.
## 0.2225  0.9025  1.4650  1.7430  1.9406  8.0000

data_plots %>% filter(harvested == "yes") %>% pull(ba_trees05to09) %>% summary()

##      Min. 1st Qu. Median     Mean 3rd Qu.    Max.
## 0.2825  1.0025  1.4650  1.5106  1.7719  4.0950
```

```
data_plots %>% filter(harvested == "no") %>% pull(ba_trees05to09) %>% summary()
```

```
##      Min. 1st Qu. Median     Mean 3rd Qu.     Max.  
## 0.2225  0.8488 1.4025 1.9588 2.6881  8.0000
```

```
summary(data_plots$ba_trees10to14)
```

```
##      Min. 1st Qu. Median     Mean 3rd Qu.     Max.  
## 0.250   1.989  2.751  2.986  3.969  6.793
```

```
data_plots %>% filter(harvested == "yes") %>% pull(ba_trees10to14) %>% summary()
```

```
##      Min. 1st Qu. Median     Mean 3rd Qu.     Max.  
## 0.725   1.948  2.699  2.996  3.969  6.325
```

```
data_plots %>% filter(harvested == "no") %>% pull(ba_trees10to14) %>% summary()
```

```
##      Min. 1st Qu. Median     Mean 3rd Qu.     Max.  
## 0.250   2.053  2.876  2.977  3.950  6.793
```

```
summary(data_plots$ba_trees15plus)
```

```
##      Min. 1st Qu. Median     Mean 3rd Qu.     Max.  
## 0.0000  0.7225 2.2287 2.9315 5.0275  9.1900
```

```
data_plots %>% filter(harvested == "yes") %>% pull(ba_trees15plus) %>% summary()
```

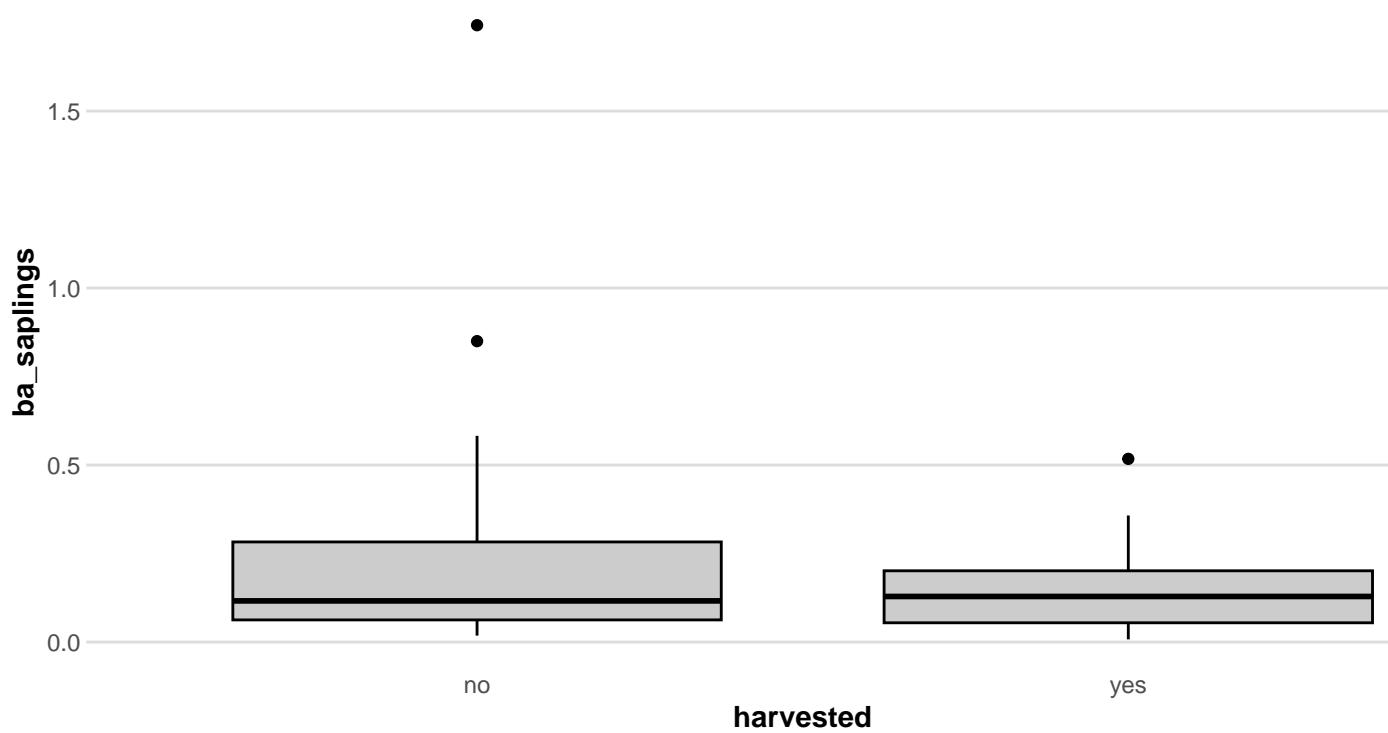
```
##      Min. 1st Qu. Median     Mean 3rd Qu.     Max.  
## 0.000   1.285  2.324  3.043  4.928  6.375
```

```
data_plots %>% filter(harvested == "no") %>% pull(ba_trees15plus) %>% summary()
```

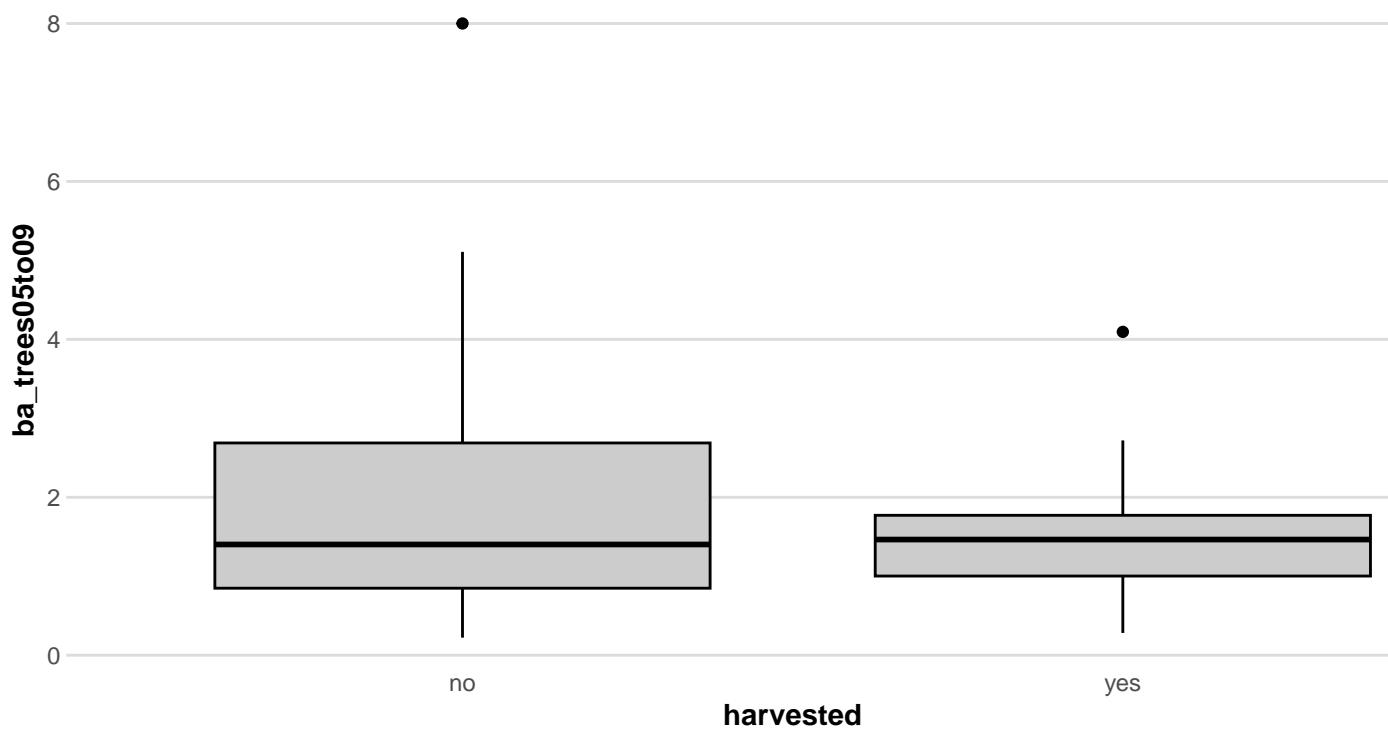
```
##      Min. 1st Qu. Median     Mean 3rd Qu.     Max.  
## 0.0000  0.6825 1.9375 2.8279 5.0812  9.1900
```

Box plots, harvesting status

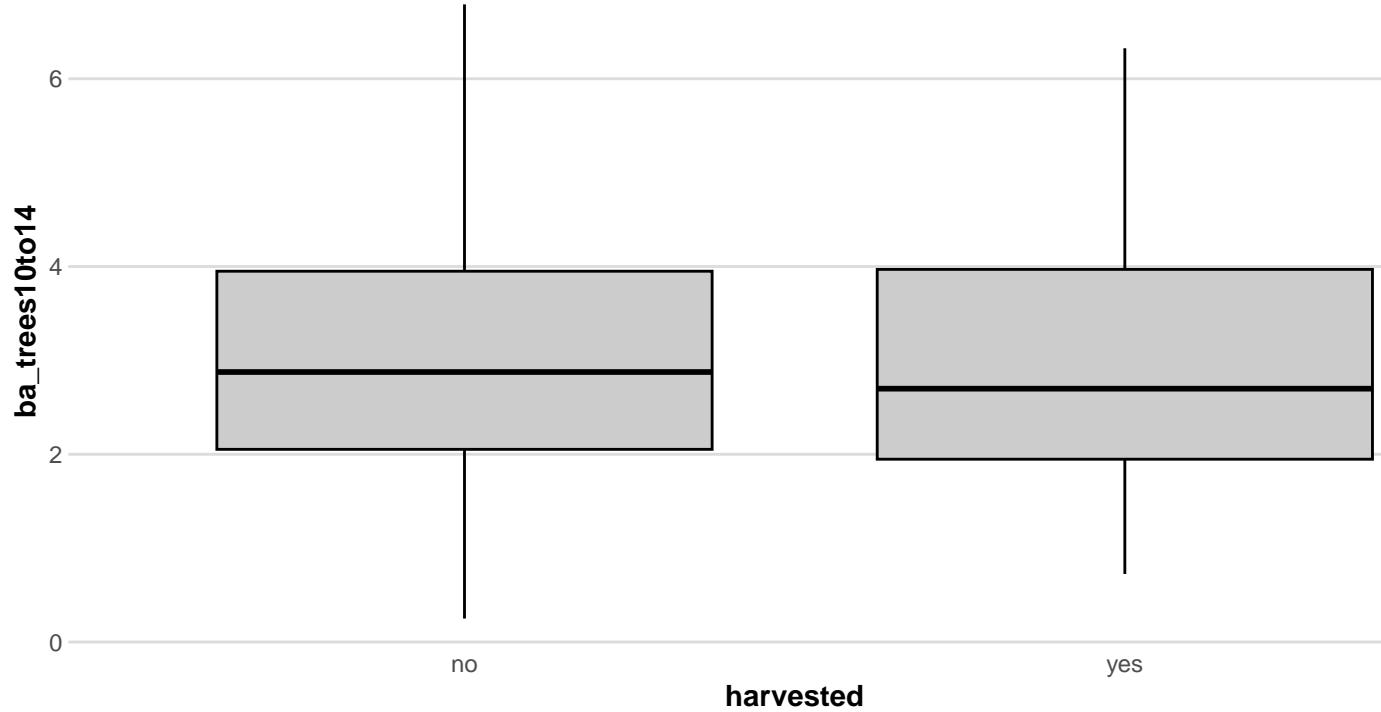
```
data_plots %>% ggplot() +  
  geom_boxplot(aes(x = harvested, y = ba_saplings),  
               col = "black", fill = "gray80") + plot_theme
```



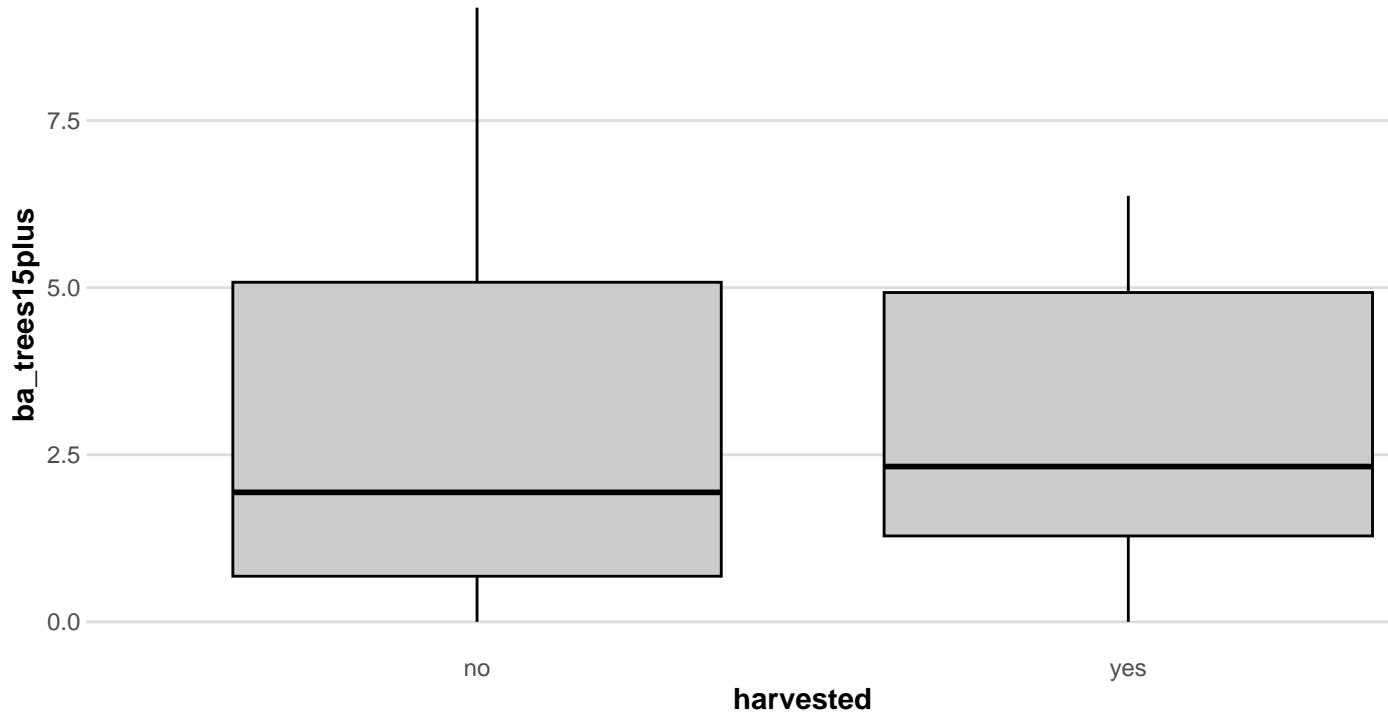
```
data_plots %>% ggplot() +
  geom_boxplot(aes(x = harvested, y = ba_trees05to09),
               col = "black", fill = "gray80") + plot_theme
```



```
data_plots %>% ggplot() +
  geom_boxplot(aes(x = harvested, y = ba_trees10to14),
    col = "black", fill = "gray80") + plot_theme
```

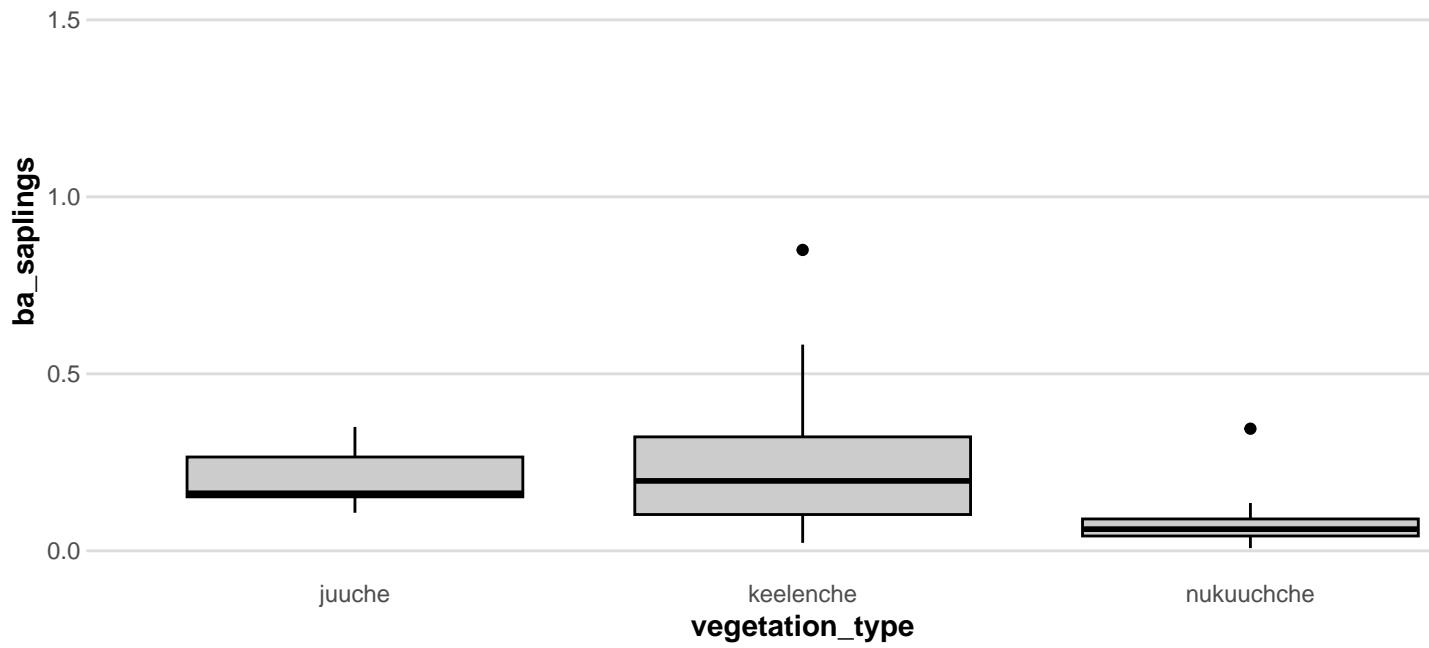


```
data_plots %>% ggplot() +
  geom_boxplot(aes(x = harvested, y = ba_trees15plus),
    col = "black", fill = "gray80") + plot_theme
```

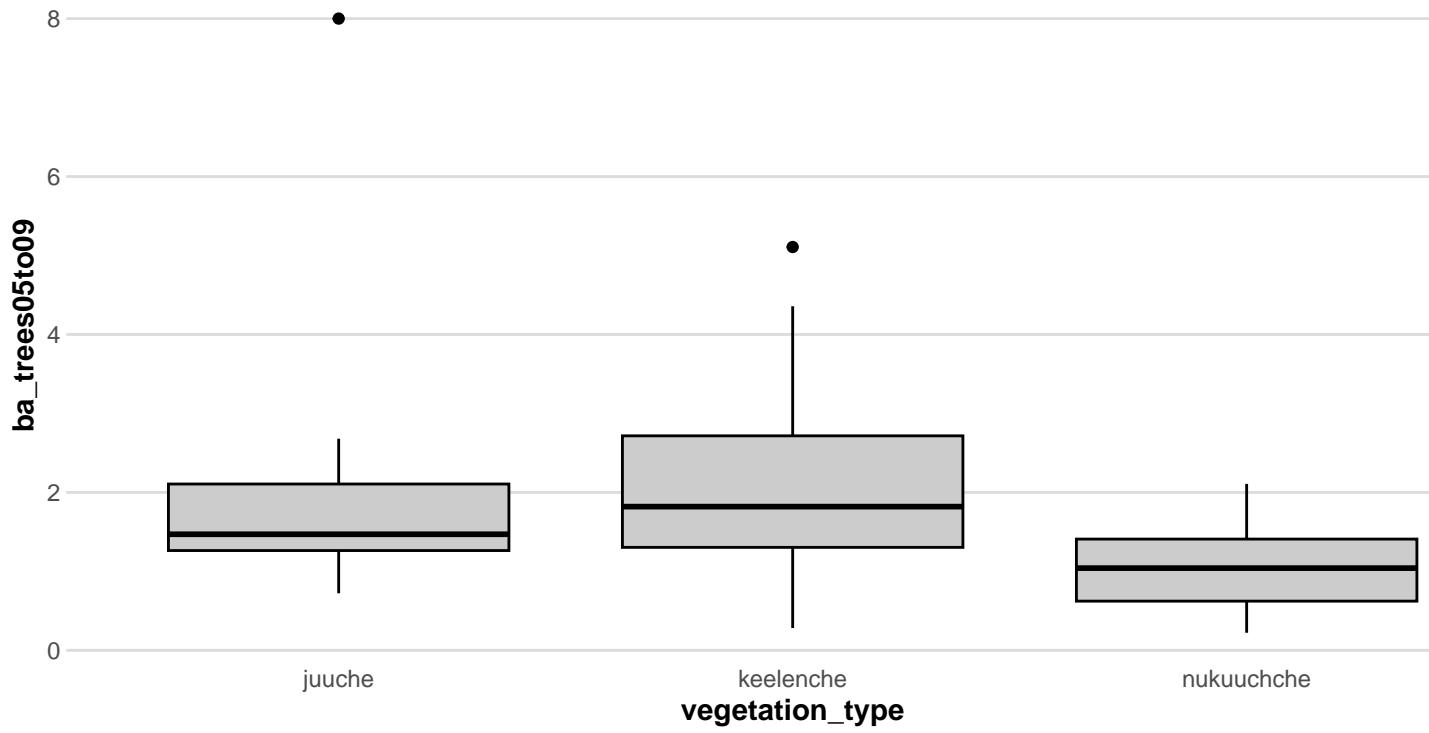


Box plots, vegetation type

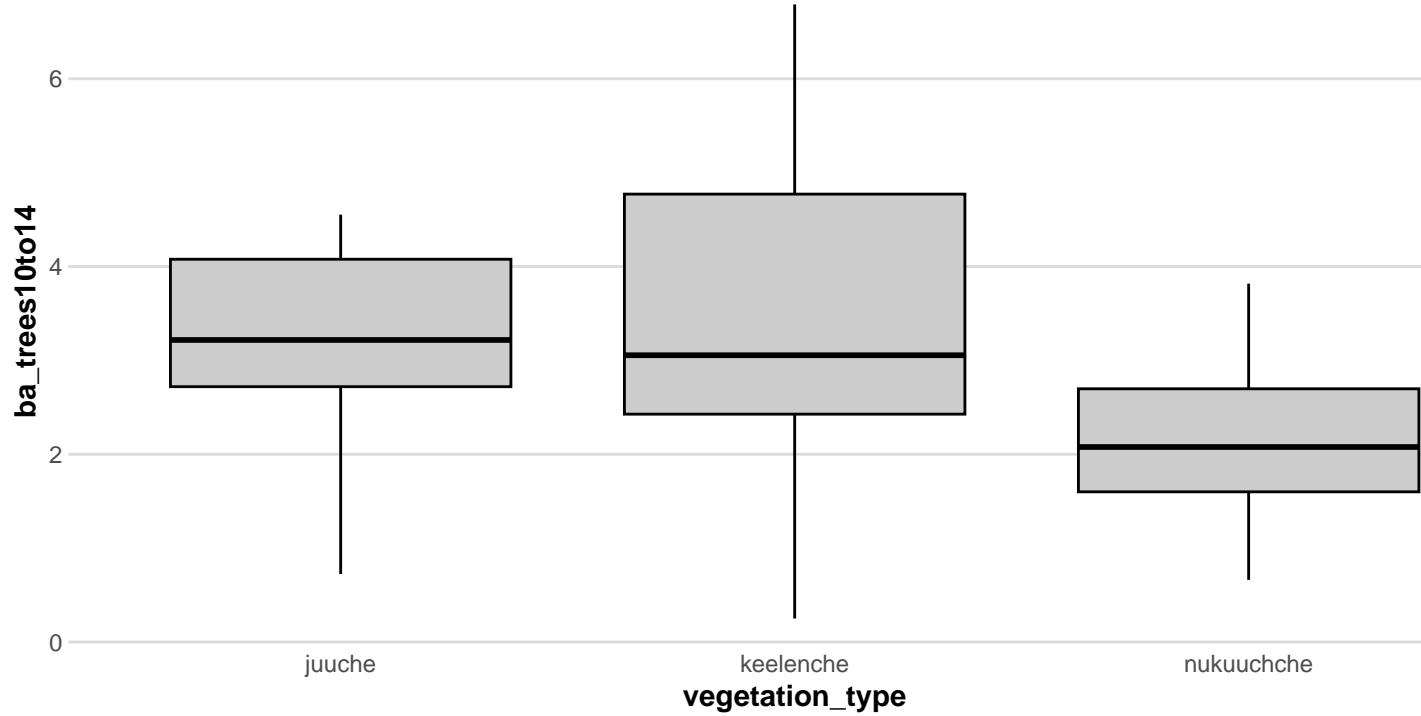
```
data_plots %>% ggplot() +  
  geom_boxplot(aes(x = vegetation_type, y = ba_saplings),  
               col = "black", fill = "gray80") + plot_theme
```



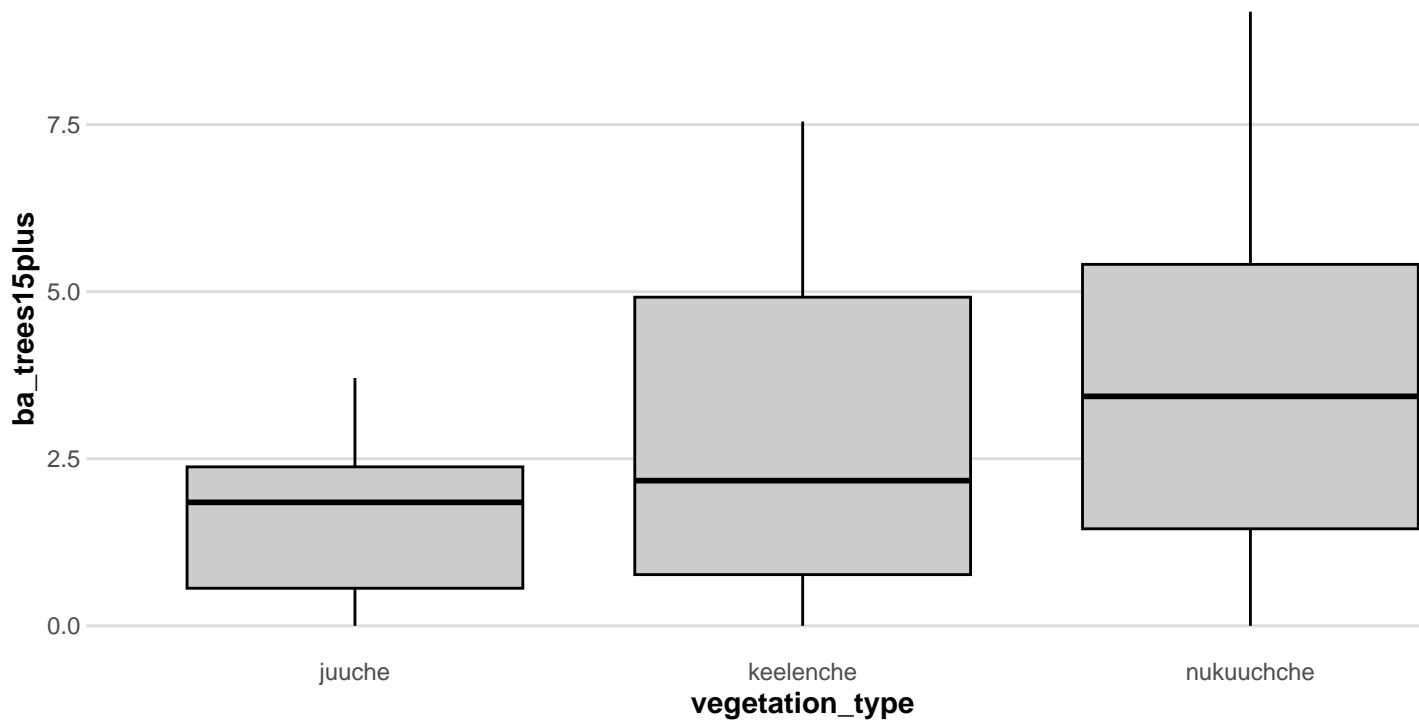
```
data_plots %>% ggplot() +
  geom_boxplot(aes(x = vegetation_type, y = ba_trees05to09),
               col = "black", fill = "gray80") + plot_theme
```



```
data_plots %>% ggplot() +
  geom_boxplot(aes(x = vegetation_type, y = ba_trees10to14),
               col = "black", fill = "gray80") + plot_theme
```

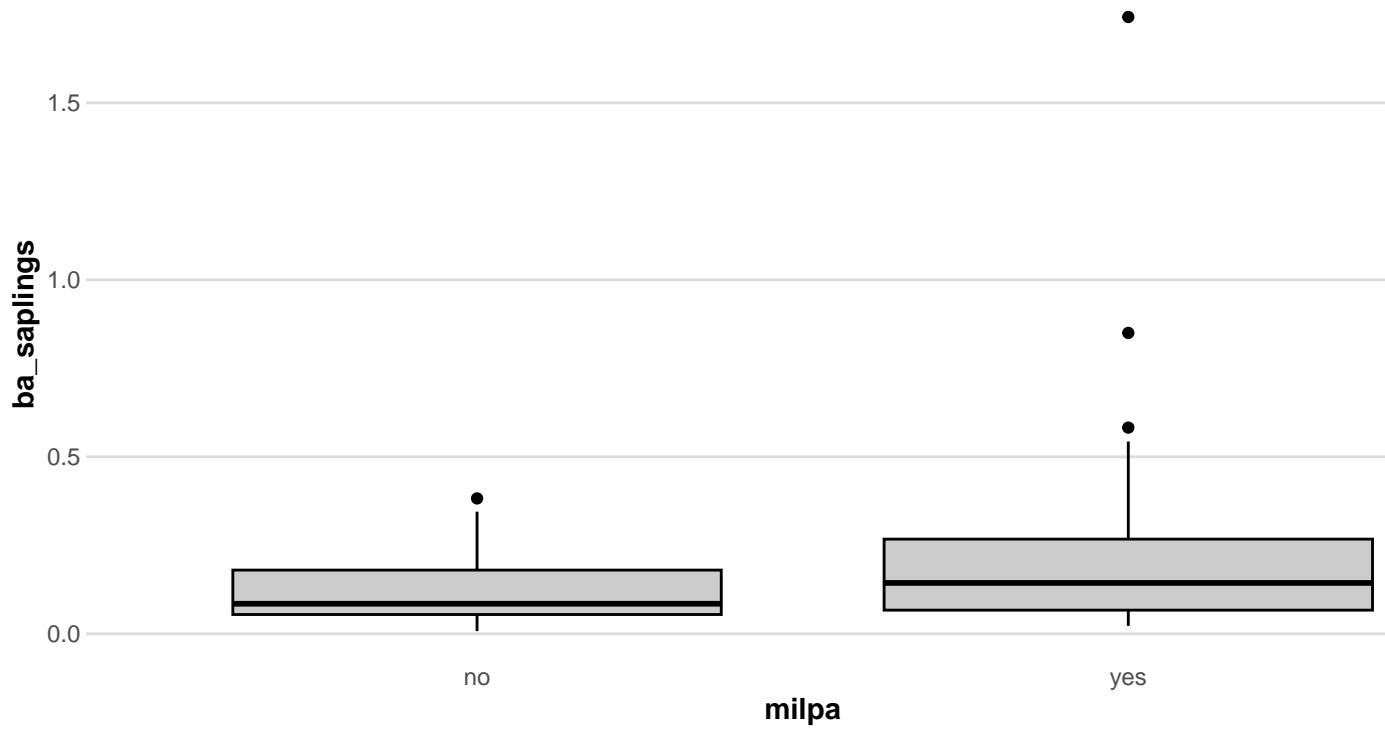


```
data_plots %>% ggplot() +
  geom_boxplot(aes(x = vegetation_type, y = ba_trees15plus),
               col = "black", fill = "gray80") + plot_theme
```

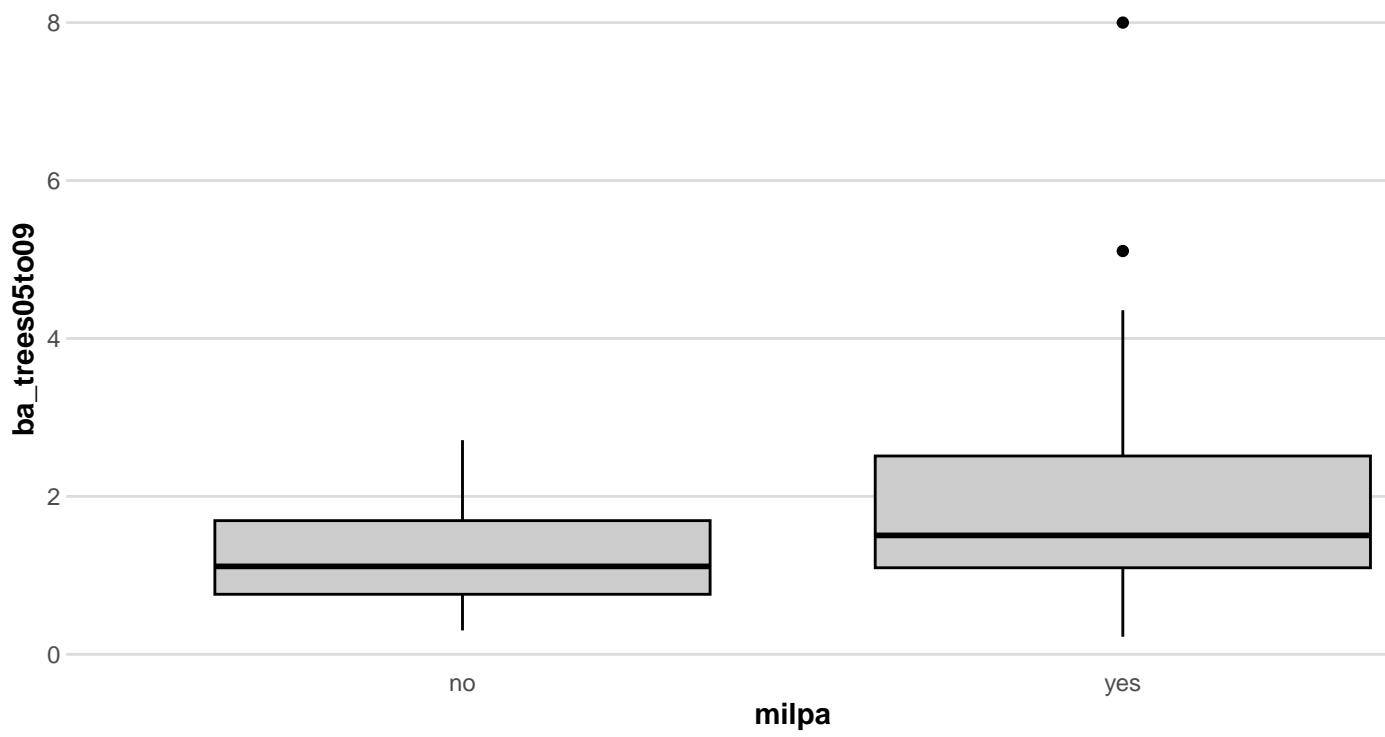


Box plots, milpa exposure

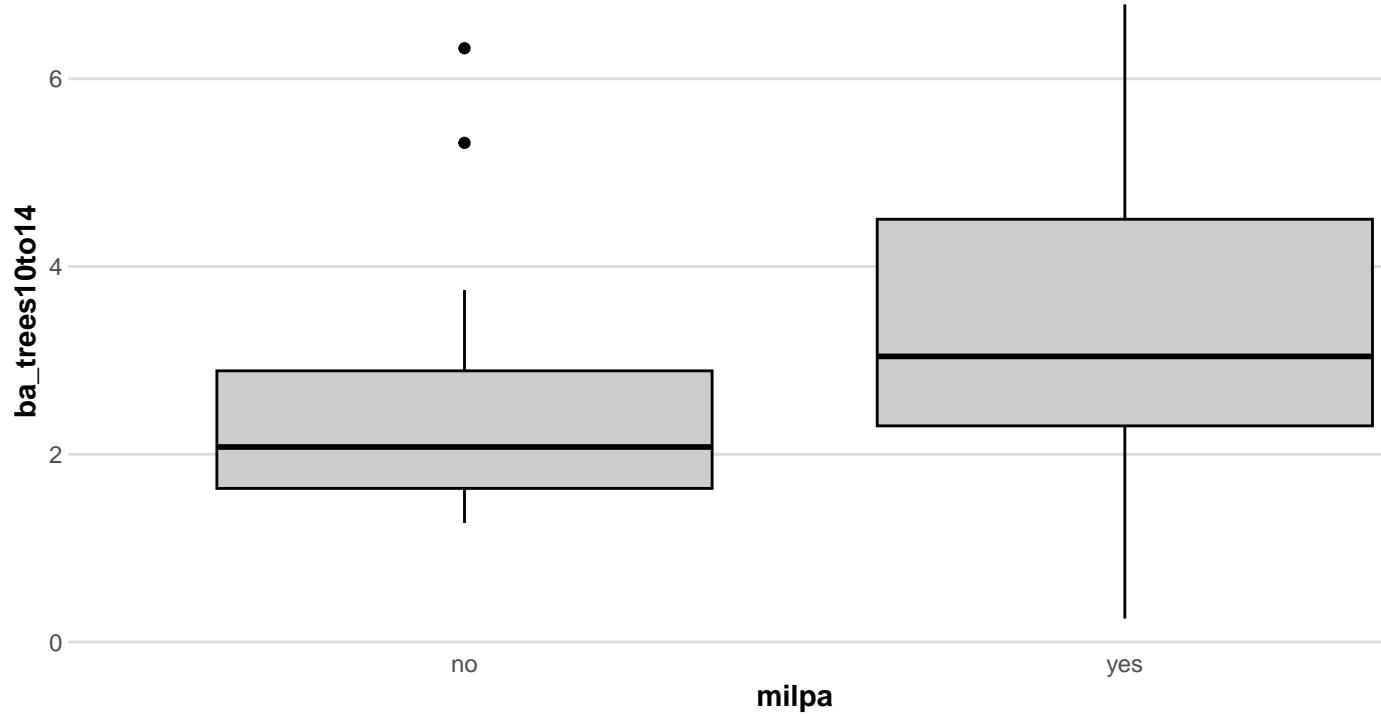
```
data_plots %>% ggplot() +
  geom_boxplot(aes(x = milpa, y = ba_saplings),
               col = "black", fill = "gray80") + plot_theme
```



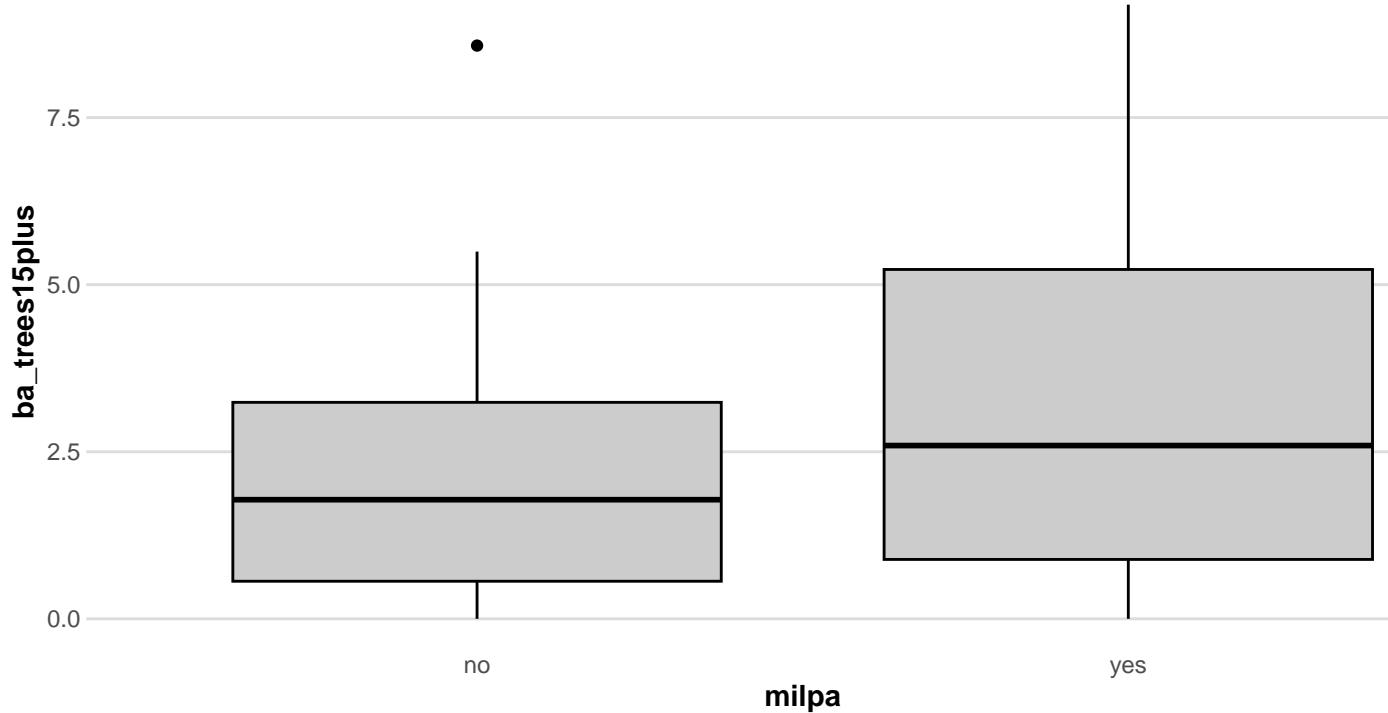
```
data_plots %>% ggplot() +
  geom_boxplot(aes(x = milpa, y = ba_trees05to09),
    col = "black", fill = "gray80") + plot_theme
```



```
data_plots %>% ggplot() +
  geom_boxplot(aes(x = milpa, y = ba_trees10to14),
    col = "black", fill = "gray80") + plot_theme
```



```
data_plots %>% ggplot() +
  geom_boxplot(aes(x = milpa, y = ba_trees15plus),
    col = "black", fill = "gray80") + plot_theme
```



Models

Seedlings

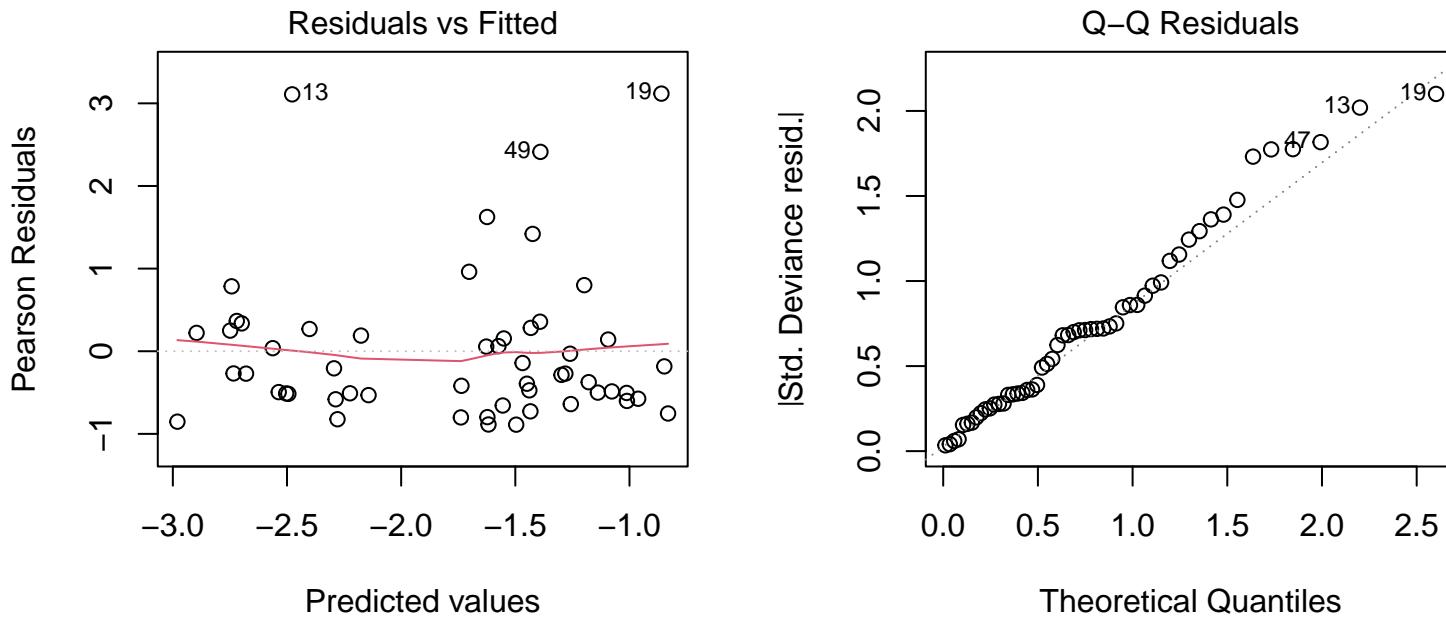
Seedlings have zero basal area by definition, so there is no need to fit a model for seedlings.

Saplings (0-4 cm DBH)

Fit gamma model and check residual plots

```
mod_saplings <- glm(ba_saplings ~ harvested + vegetation_type + milpa + latitude + longitude,
                      family = Gamma(link="log"), data = data_plots)

par(mfrow = c(1,2))
plot(mod_saplings, which = 1:2)
```



Summarize the model

```
summary(mod_saplings)
```

```
##
## Call:
## glm(formula = ba_saplings ~ harvested + vegetation_type + milpa +
##      latitude + longitude, family = Gamma(link = "log"), data = data_plots)
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)             -0.49016   0.56963 -0.860  0.39388
## harvestedyes            -0.19474   0.26132 -0.745  0.45985
## vegetation_typekeelenche -0.44423   0.43943 -1.011  0.31724
## vegetation_typenukuuchche -1.88618   0.54065 -3.489  0.00106 **
## milpayes                 -0.39218   0.37829 -1.037  0.30518
## latitude                  0.02091   0.14467  0.145  0.88569
## longitude                 0.18443   0.16741  1.102  0.27621
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.9178797)
##
## Null deviance: 56.457 on 53 degrees of freedom
## Residual deviance: 36.402 on 47 degrees of freedom
## AIC: -74.772
##
## Number of Fisher Scoring iterations: 10
```

Construct analysis of deviance table

```
Anova(mod_saplings, type = 2, test.statistic = "LR")
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: ba_saplings
##          LR Chisq Df Pr(>Chisq)
## harvested      0.5078  1   0.476080
## vegetation_type 13.1629  2   0.001386 **
## milpa          0.9552  1   0.328391
## latitude        0.0258  1   0.872281
## longitude       1.0034  1   0.316492
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

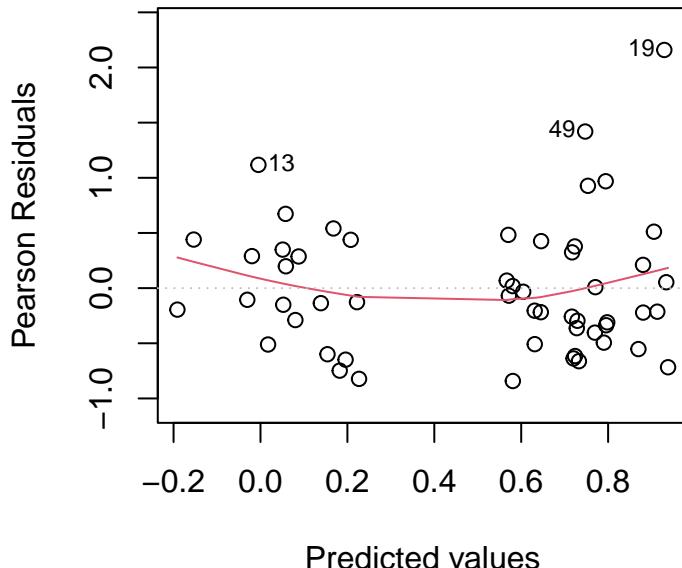
Trees (5-9 cm DBH)

Fit gamma model and check residual plots

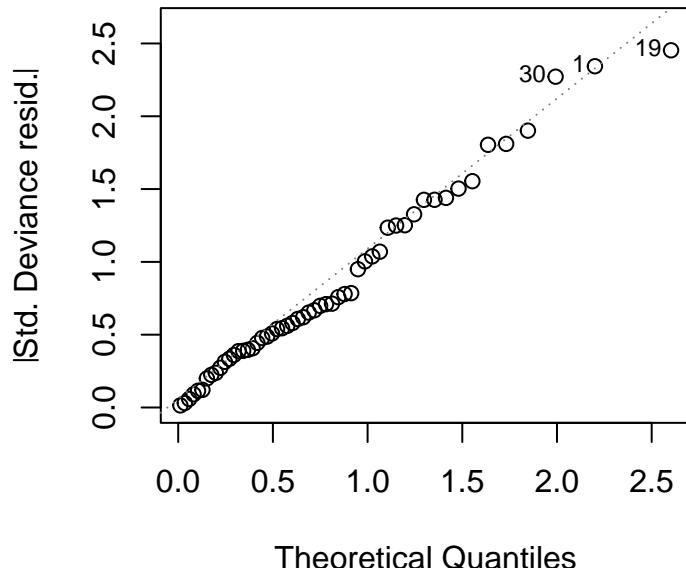
```
mod_trees05to09 <- glm(ba_trees05to09 ~ harvested + vegetation_type + milpa + latitude + longitude,
family = Gamma(link="log"), data = data_plots)
```

```
par(mfrow = c(1,2))
plot(mod_trees05to09, which = 1:2)
```

Residuals vs Fitted



Q-Q Residuals



Summarize the model

```
summary(mod_trees05to09)

##
## Call:
## glm(formula = ba_trees05to09 ~ harvested + vegetation_type +
##      milpa + latitude + longitude, family = Gamma(link = "log"),
##      data = data_plots)
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 0.85527   0.37532   2.279   0.0273 *  
## harvestedyes -0.14624   0.17218  -0.849   0.4000    
## vegetation_typekeelenche -0.07124   0.28953  -0.246   0.8067    
## vegetation_typenukuuchche -0.75827   0.35623  -2.129   0.0386 *  
## milpayes       0.03935   0.24925   0.158   0.8752    
## latitude        0.02911   0.09532   0.305   0.7614    
## longitude       0.06558   0.11030   0.595   0.5550    
## ---            
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.398474)
##
## Null deviance: 26.127  on 53  degrees of freedom
## Residual deviance: 19.698  on 47  degrees of freedom
## AIC: 151.94
##
## Number of Fisher Scoring iterations: 7
```

Construct analysis of deviance table

```
Anova(mod_trees05to09, type = 2, test.statistic = "LR")

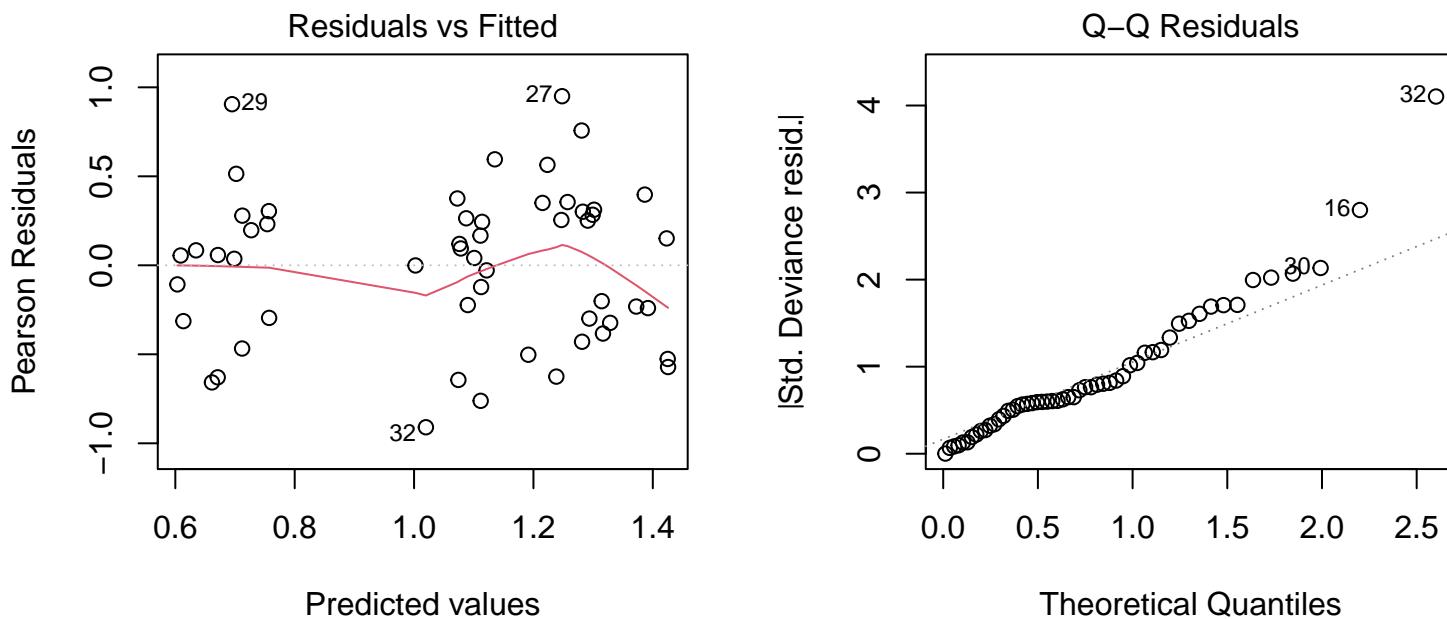
## Analysis of Deviance Table (Type II tests)
##
## Response: ba_trees05to09
##              LR Chisq Df Pr(>Chisq)    
## harvested      0.6907  1   0.40594    
## vegetation_type 7.1613  2   0.02786 *  
## milpa          0.0232  1   0.87893    
## latitude        0.1096  1   0.74062    
## longitude       0.3509  1   0.55362    
## ---            
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Trees (10-14 cm DBH)

Fit gamma model and check residual plots

```
mod_trees10to14 <- glm(ba_trees10to14 ~ harvested + vegetation_type + milpa + latitude + longitude,
                        family = Gamma(link="log"), data = data_plots)

par(mfrow = c(1,2))
plot(mod_trees10to14, which = 1:2)
```



Summarize the model

```
summary(mod_trees10to14)

##
## Call:
## glm(formula = ba_trees10to14 ~ harvested + vegetation_type +
##     milpa + latitude + longitude, family = Gamma(link = "log"),
##     data = data_plots)
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)             1.07093   0.27035  3.961 0.000252 ***
## harvestedyes            0.04629   0.12402  0.373 0.710645
## vegetation_typekeelenche 0.02043   0.20856  0.098 0.922391
## vegetation_typenkuuchche -0.26729   0.25660 -1.042 0.302909
## milpayes                0.08109   0.17954  0.452 0.653612
## latitude                 -0.04395   0.06866 -0.640 0.525250
## longitude                -0.15589   0.07945 -1.962 0.055697 .
```

```

## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.2067608)
##
## Null deviance: 17.161 on 53 degrees of freedom
## Residual deviance: 13.663 on 47 degrees of freedom
## AIC: 198.29
##
## Number of Fisher Scoring iterations: 6

```

Construct analysis of deviance table

```
Anova(mod_trees10to14, type = 2, test.statistic = "LR")
```

```

## Analysis of Deviance Table (Type II tests)
##
## Response: ba_trees10to14
##           LR Chisq Df Pr(>Chisq)
## harvested      0.1371  1    0.7111
## vegetation_type 2.3196  2    0.3135
## milpa          0.2111  1    0.6459
## latitude        0.3915  1    0.5315
## longitude       3.7470  1    0.0529 .
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Trees (15+ cm DBH)

Fit gamma model and check residual plots

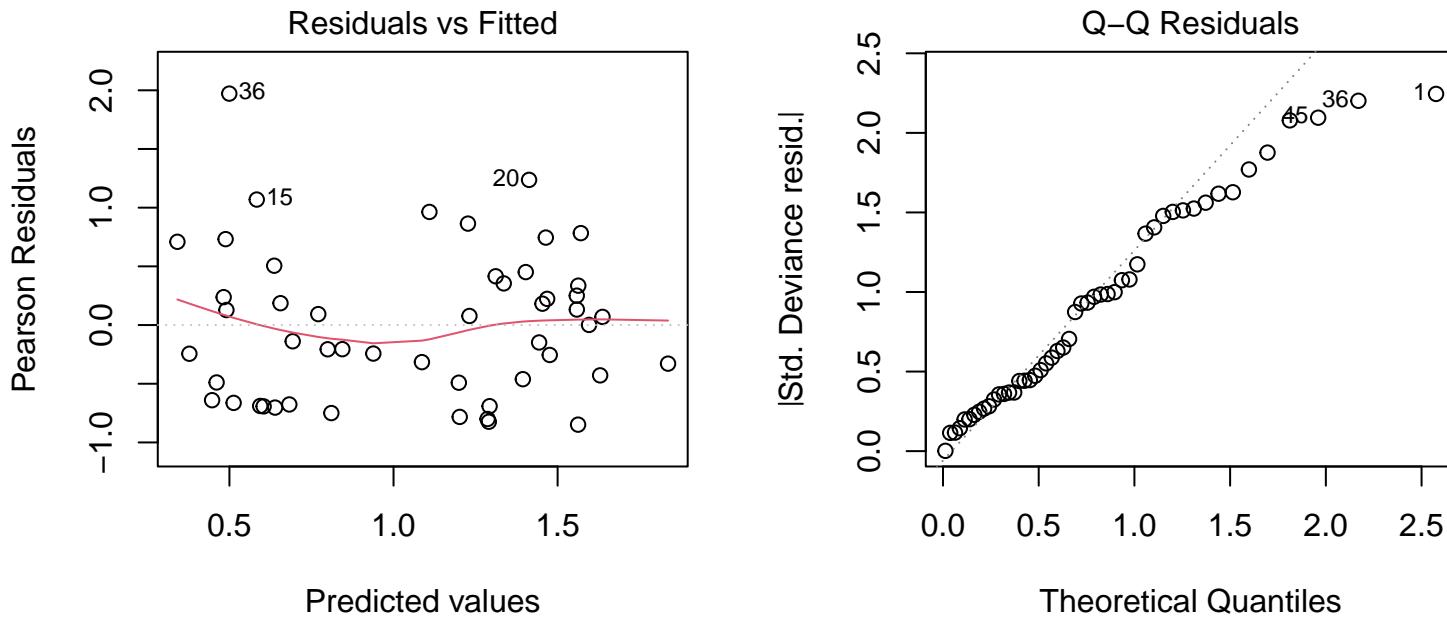
There are four plots with zero 15+ cm trees. We decide to drop these four plots; otherwise, we cannot use the gamma GLM.

```

mod_trees15plus <- glm(ba_trees15plus ~ harvested + vegetation_type + milpa + latitude + longitude,
                         family = Gamma(link="log"), data = data_plots %>% filter(stemden_trees15plus_count > 0))

par(mfrow = c(1,2))
plot(mod_trees15plus, which = 1:2)

```



Summarize the model

```
summary(mod_trees15plus)
```

```
##
## Call:
## glm(formula = ba_trees15plus ~ harvested + vegetation_type +
##      milpa + latitude + longitude, family = Gamma(link = "log"),
##      data = data_plots %>% filter(stemden_trees15plus_count >
##          0))
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)             -0.2001    0.4302  -0.465  0.64429
## harvestedyes            0.1277    0.1906   0.670  0.50658
## vegetation_typekeelenche 0.2176    0.3294   0.661  0.51240
## vegetation_typenukuuchche 1.1181    0.3986   2.805  0.00752 **
## milpayes                 0.9024    0.2867   3.148  0.00299 **
## latitude                  -0.1389   0.1062  -1.308  0.19790
## longitude                 -0.3637   0.1203  -3.023  0.00420 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.452262)
##
## Null deviance: 32.826  on 49  degrees of freedom
## Residual deviance: 23.479  on 43  degrees of freedom
## AIC: 206.41
##
## Number of Fisher Scoring iterations: 14
```

Construct analysis of deviance table

```
Anova(mod_trees15plus, type = 2, test.statistic = "LR")  
  
## Analysis of Deviance Table (Type II tests)  
##  
## Response: ba_trees15plus  
##          LR Chisq Df Pr(>Chisq)  
## harvested      0.4367  1   0.508735  
## vegetation_type 13.5138  2   0.001163 **  
## milpa          10.1466  1   0.001446 **  
## latitude        1.4942  1   0.221559  
## longitude       9.5168  1   0.002036 **  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```