Ant Mound eCognition Analysis

R Pienaar

2023-12-12

load packages

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.2
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.3.1
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v lubridate 1.9.2     v tibble    3.2.1
## v purrr     1.0.1     v tidyr     1.3.0
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(dplyr)
library(tidyr)
library(ggpubr)
library(patchwork)
```

```
## Warning: package 'patchwork' was built under R version 4.3.2
```

```
library(ggh4x)
```

```
## Warning: package 'ggh4x' was built under R version 4.3.2
```

```
library(Rmisc)
```

```
## Warning: package 'Rmisc' was built under R version 4.3.1
```

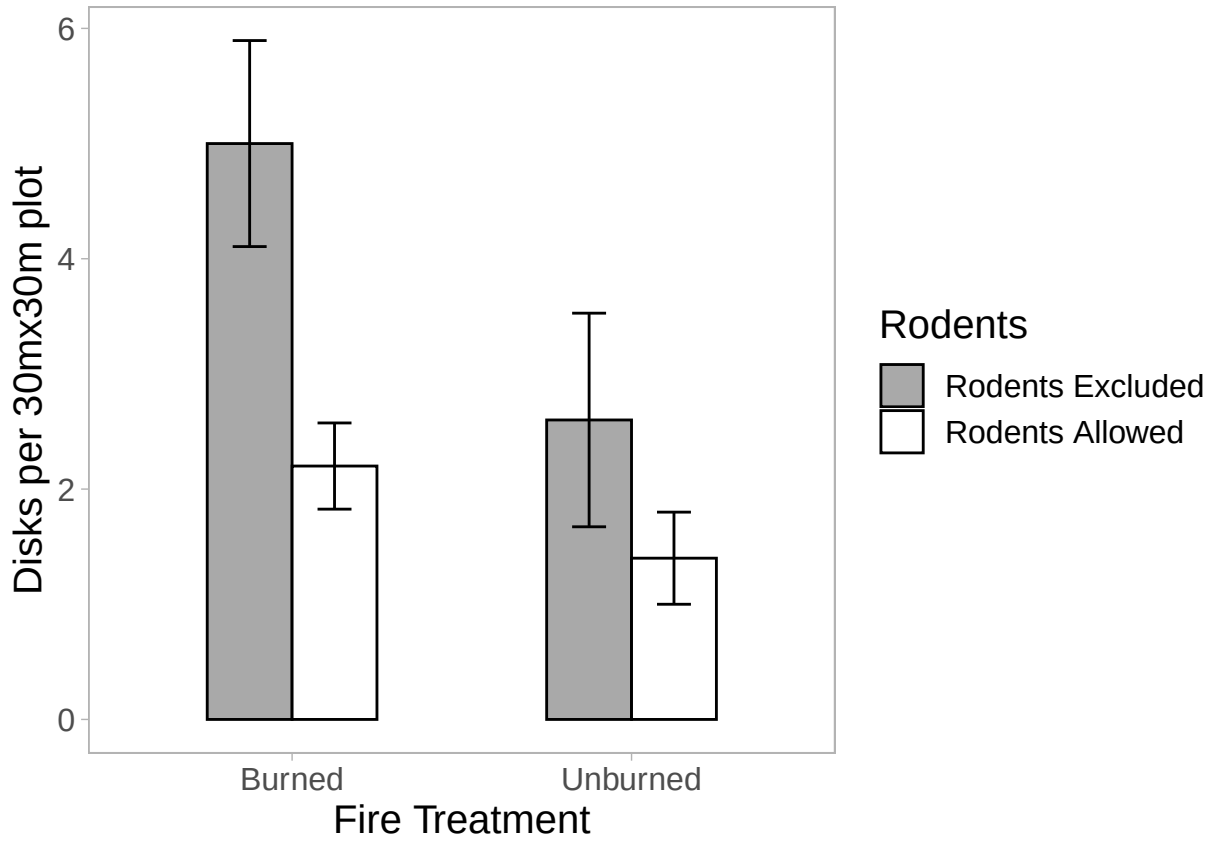
```
## Loading required package: lattice
## Loading required package: plyr
## ------------------------------------------------------------------------------
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
## ------------------------------------------------------------------------------
##
## Attaching package: 'plyr'
##
## The following object is masked from 'package:ggpubr':
##
```

```
##     mutate
##
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
##
## The following object is masked from 'package:purrr':
##
##     compact
```

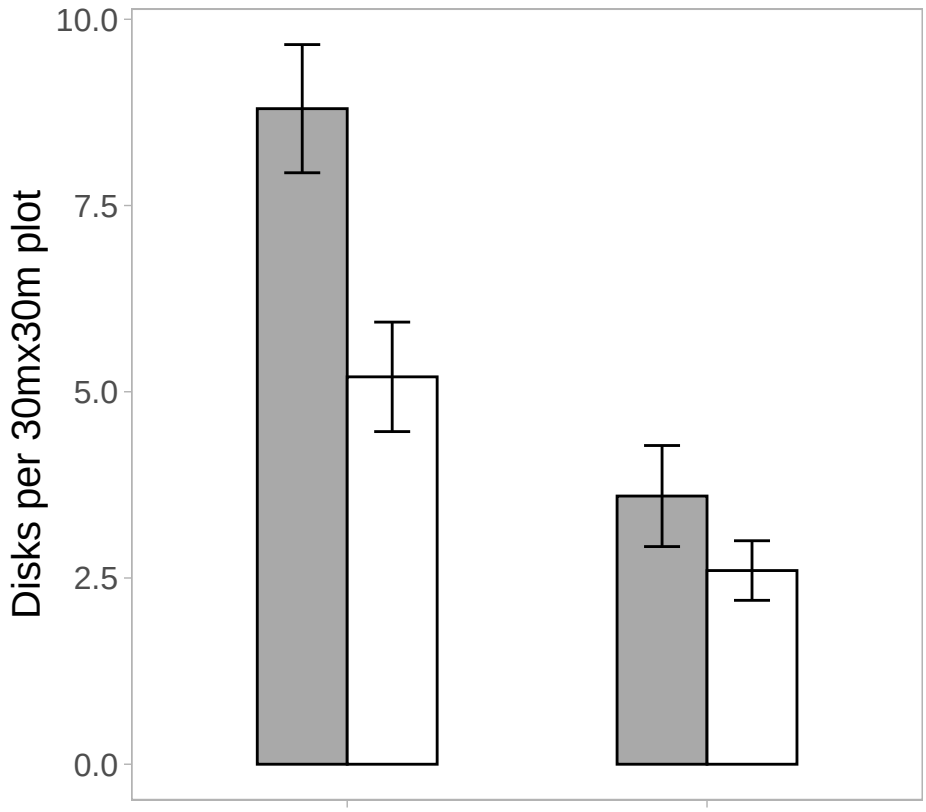
```
d21 <- read.csv("2021_mound_per_plot.csv")

summary_df21 <- d21 %>%
  group_by(Burned, Rodents, Block) %>%
  dplyr::summarise(Count = n_distinct(OBJECTID))
```

```
## `summarise()` has grouped output by 'Burned', 'Rodents'. You can override using
## the `.groups` argument.
```

```
all_combinations <- expand(d21, Burned, Rodents, Block, ID = NULL)

summary_df21 <- dplyr::left_join(all_combinations, summary_df21, by = c("Burned", "Rodents", "Block"))

summary_df21[is.na(summary_df21)]<- 0

d21plot <- ggplot(data = summary_df21, aes(x = Burned, y = Count, fill = Rodents))+
  stat_summary(geom = "bar", fun = mean, position = "dodge", color = "black", width = 0.5) +
  stat_summary(geom = "errorbar", fun.data = mean_se, width = 0.2, position = position_dodge(0.5))+
  ylab("Disks per 30mx30m plot")+
  theme(axis.text.y = element_text(color = "black"))+
  theme(axis.text.x = element_text(color = "black"))+
  xlab("Fire Treatment")+
  theme_light()+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        panel.background = element_rect(fill="transparent"))+
  theme(plot.title = element_text(hjust = 0.5), text = element_text(size=15))+
  scale_fill_manual(values = c("dark grey","white"),labels=c('Rodents Excluded', 'Rodents Allowed'))+
  scale_x_discrete(labels= c("Burned", "Unburned"), )+
  theme(legend.position = "right", aspect.ratio = 1)

d21plot
```



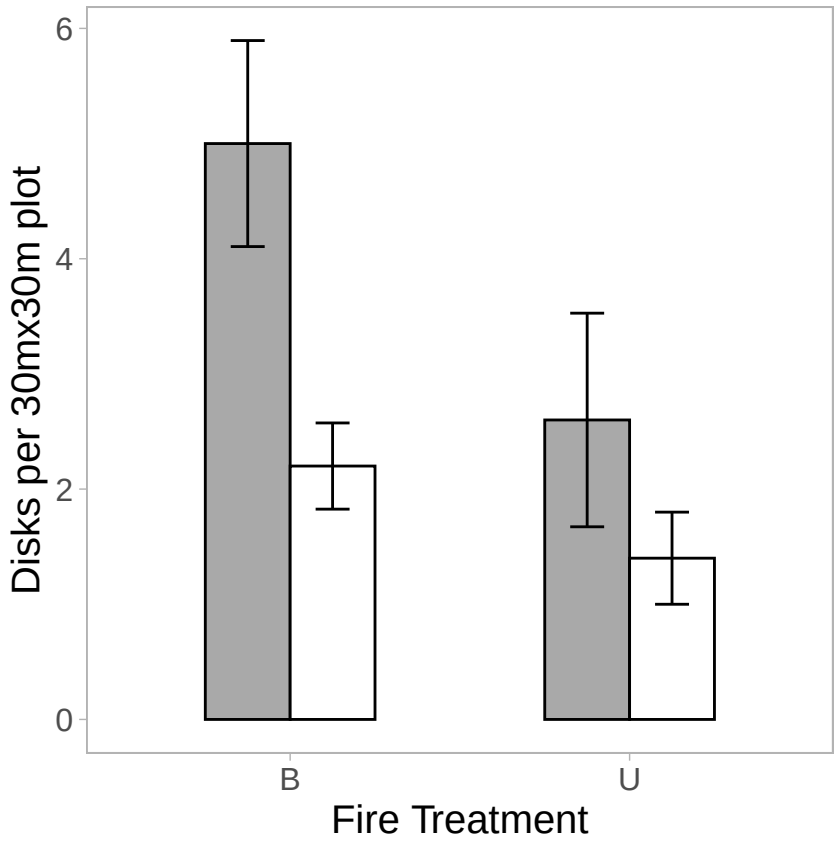

```
d21_m <- read.csv("Density_manual.csv")



d21_m_plot <- ggplot(data = d21_m, aes(x = Burned, y = Count, fill = Rodents))+
  stat_summary(geom = "bar", fun = mean, position = "dodge", color = "black", width = 0.5) +
  stat_summary(geom = "errorbar", fun.data = mean_se, width = 0.2, position = position_dodge(0.5))+
  ylab("Disks per 30mx30m plot")+
  xlab(NULL)+
  theme(axis.text.y = element_text(color = "black"))+
  theme(axis.text.x = element_text(color = "black"))+
  theme_light()+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        panel.background = element_rect(fill="transparent"))+
  theme(plot.title = element_text(hjust = 0.5), text = element_text(size=15))+
  scale_fill_manual(values = c("dark grey","white"),labels=c('Rodents Excluded', 'Rodents Allowed'))+
  scale_x_discrete(labels= c("", ""), )+
  theme(legend.position = "none", aspect.ratio = 1)

d21_m_plot
```




```
dall_d <- read.csv("year_comp.csv")

d21_d <- dall_d %>%
  group_by(Burned, Rodents, Block) %>%
  dplyr::summarise(Count = n_distinct(OID_))
```

```
## `summarise()` has grouped output by 'Burned', 'Rodents'. You can override using
## the `.groups` argument.
```

```
d21_d <- dplyr::left_join(all_combinations, d21_d, by = c("Burned", "Rodents", "Block"))

d21_d[is.na(d21_d)]<- 0

d21_d_plot <- ggplot(data = d21_d, aes(x = Burned, y = Count, fill = Rodents))+
  stat_summary(geom = "bar", fun = mean, position = "dodge", color = "black", width = 0.5) +
  stat_summary(geom = "errorbar", fun.data = mean_se, width = 0.2, position = position_dodge(0.5))+
  ylab("Disks per 30mx30m plot")+
  xlab("Fire Treatment")+
  theme_light()+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        panel.background = element_rect(fill="transparent"))+
  theme(plot.title = element_text(hjust = 0.5), text = element_text(size=15))+
  scale_fill_manual(values = c("dark grey","white"))+
  theme(legend.position = "none", aspect.ratio = 1)

d21_d_plot
```



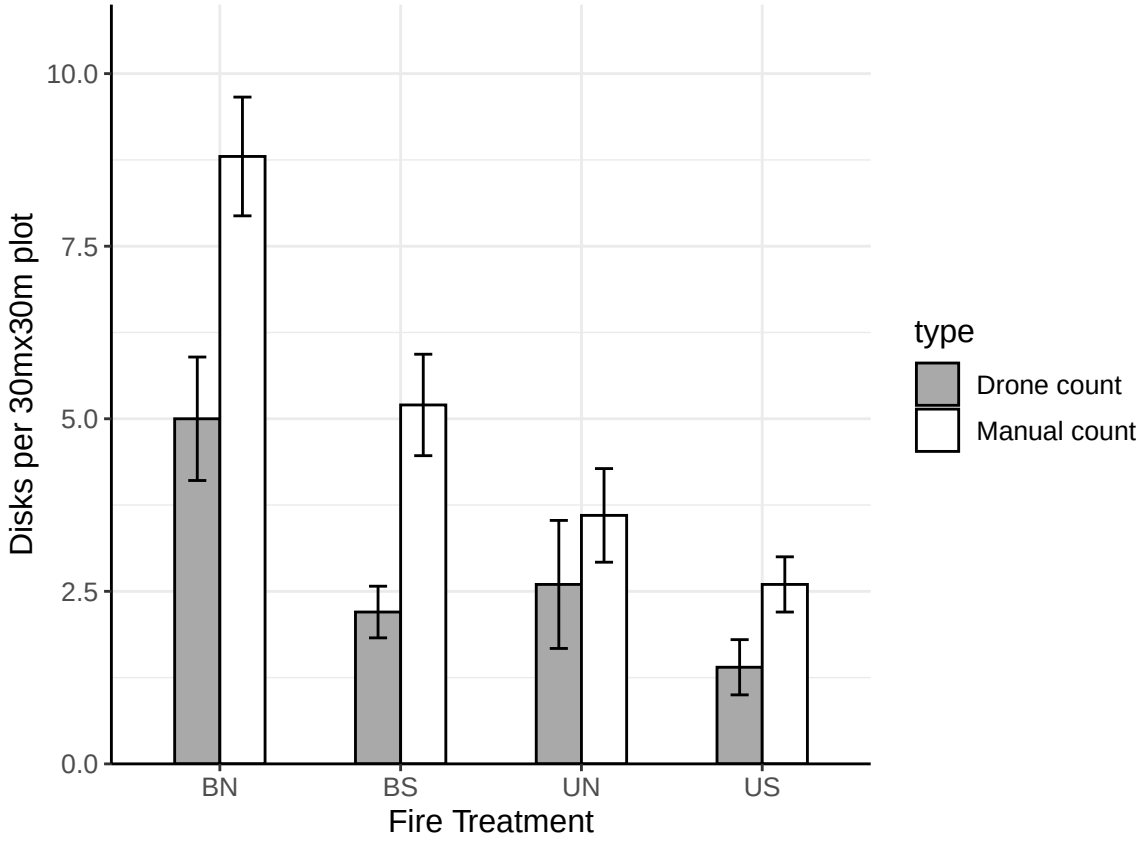

```
d21_d
```

```
## # A tibble: 20 x 4
##    Burned Rodents Block Count
##    <chr>  <chr>   <int> <int>
##  1 B      N           1     7
##  2 B      N           2     6
##  3 B      N           3     4
##  4 B      N           4     6
##  5 B      N           5     2
##  6 B      S           1     2
##  7 B      S           2     1
##  8 B      S           3     3
##  9 B      S           4     3
## 10 B      S           5     2
## 11 U      N           1     4
## 12 U      N           2     5
## 13 U      N           3     0
## 14 U      N           4     3
## 15 U      N           5     1
## 16 U      S           1     1
## 17 U      S           2     1
## 18 U      S           3     3
## 19 U      S           4     1
## 20 U      S           5     1
```

```
d21_m
```

```
##    Block Plot Burned Rodents Count     Area
## 1      1   BN      B       N     8 9503.318
## 2      2   BN      B       N     9 8607.964
## 3      3   BN      B       N     6 6565.929
## 4      4   BN      B       N    10 9000.663
## 5      5   BN      B       N    11 6418.274
## 6      1   BS      B       S     5 5089.380
## 7      2   BS      B       S     8 5664.292
## 8      3   BS      B       S     5 6776.415
## 9      4   BS      B       S     4 5215.044
## 10     5   BS      B       S     4 3647.389
## 11     1   UN      U       N     6 6352.300
## 12     2   UN      U       N     4 6053.849
## 13     3   UN      U       N     2 3047.345
## 14     4   UN      U       N     3 3078.761
## 15     5   UN      U       N     3 2733.186
## 16     1   US      U       S     3 4156.327
## 17     2   US      U       S     1 1884.956
## 18     3   US      U       S     3 2846.283
## 19     4   US      U       S     3 3424.336
## 20     5   US      U       S     3 5403.539
```

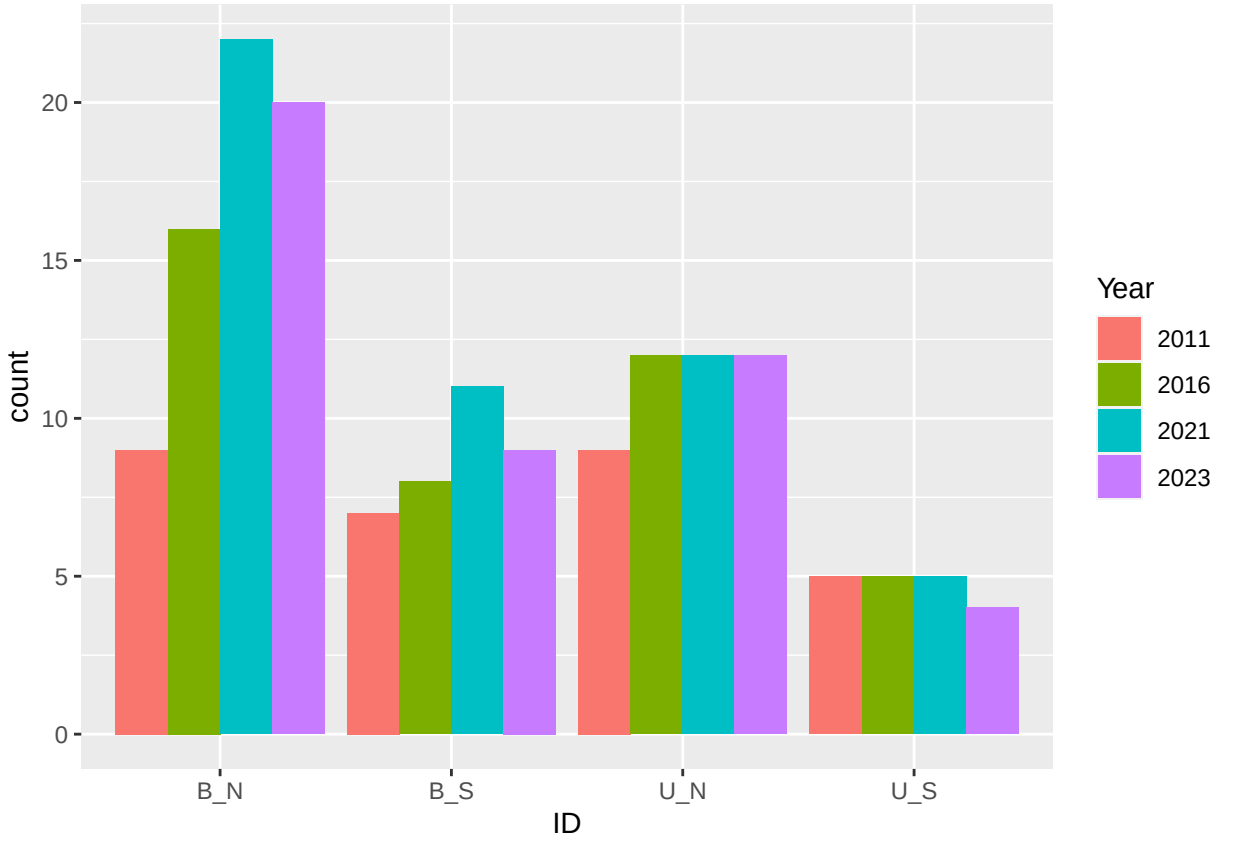
```
d21_d$Block <- as.factor(d21_d$Block)
d21_m$Block <- as.factor(d21_m$Block)

colnames(d21_m)[colnames(d21_m) == "Count"]<- "Count_m"

d_both <- cbind(d21_d, d21_m$Count_m)

colnames(d_both)[colnames(d_both) == "d21_m$Count_m"]<- "Count_m"

d_both %>% unite("ID", Burned, Rodents, remove = FALSE) %>%
  pivot_longer(cols = c(Count, Count_m), names_to = "type", values_to = "Count") %>%
ggplot(aes(x= ID, y = Count, fill = type))+
  #geom_bar(stat = "identity", position = position_dodge())+
  stat_summary(geom = "bar", fun = mean, position = "dodge", color = "black", width = 0.5) +
  stat_summary(geom = "errorbar", fun.data = mean_se, width = 0.2, position = position_dodge(0.5))+
  ylab("Disks per 30mx30m plot")+
  xlab("Fire Treatment")+
 theme_bw()+
  theme(panel.background = element_rect(fill="transparent"), panel.border = element_blank())+
  theme(plot.title = element_text(hjust = 0.5), text = element_text(size=12))+
  theme(legend.position = "right", aspect.ratio = 1)+
  theme(axis.line = element_line(color = 'black'))+
  scale_x_discrete(labels= c("BN", "BS", "UN", "US"), )+
  scale_fill_manual(values = c("dark grey","white"),labels=c('Drone count', 'Manual count'))+
  theme(legend.position = "right", aspect.ratio = 1)+
    scale_y_continuous(expand = c(0, 0), limits = c(0, 11))
```




```
d21_drone <- read.csv("year_comp.csv")

head(d21_drone)
```

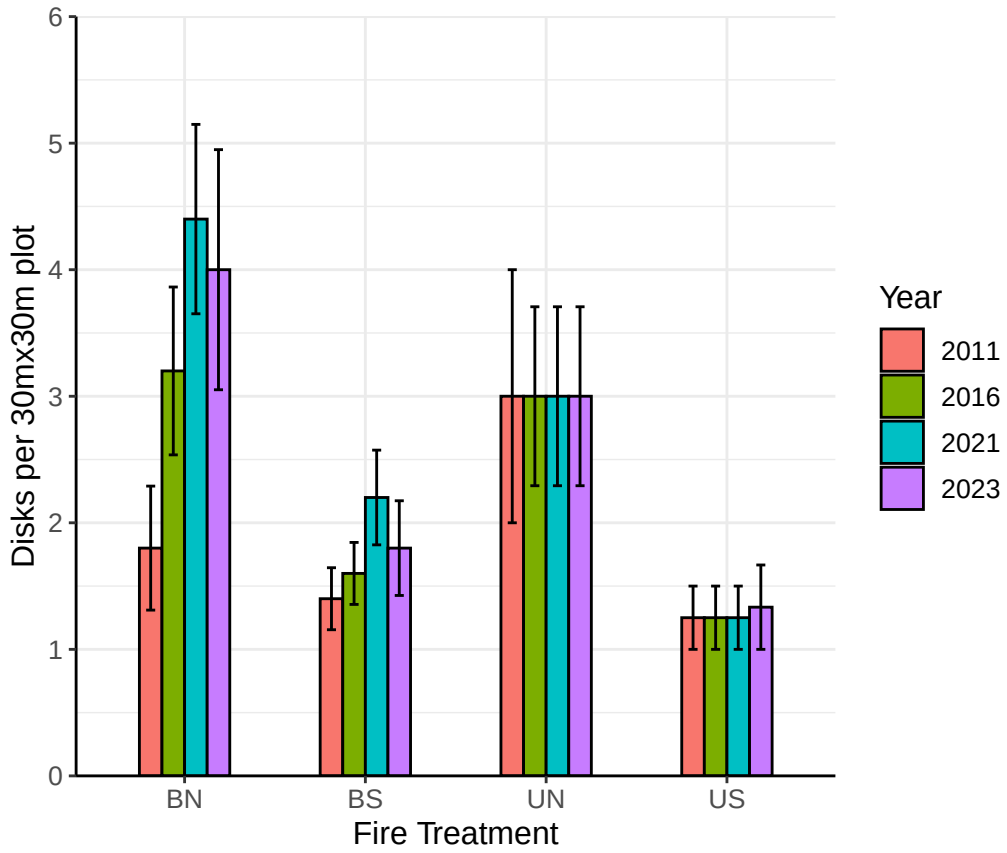
```
##   OID_ Block Plot Burned Rodents Y2011 Y2016 Y2021 Y2023
## 1   30     1   US      U       S     1     1     1     0
## 2   31     1   BN      B       N     0     1     1     1
## 3   32     1   BN      B       N     1     1     1     1
## 4   33     1   BN      B       N     0     1     1     1
## 5   36     1   BN      B       N     1     1     1     1
## 6   37     1   BN      B       N     1     1     1     1
```

```
colnames(d21_drone) <- c("OID_", "Block", "Plot", "Burned", "Rodents",
    "2011", "2016",  "2021", "2023")


d21_drone <- d21_drone %>% unite("ID", Burned, Rodents, remove = FALSE) %>%
  pivot_longer(cols = c("2011", "2016", "2021", "2023"), names_to = "Year",
               values_to = "Count")

#d21_drone$Year <- as.numeric(d21_drone$Year)

ggplot(data= d21_drone[d21_drone$Count==1,], aes(x=ID, fill = Year))+
    geom_bar(position = position_dodge())
```




```
d21_drone[d21_drone$Count==1,] |>
  group_by(Plot, Year, Block) |>
  dplyr::summarise(mean_count = mean(n())) |>
ggplot( aes(x=Plot, y = mean_count, fill = Year))+
  #geom_col(position = position_dodge())+
  stat_summary(geom = "bar", fun = mean, position = "dodge", color = "black", width = 0.5) +
  stat_summary(geom = "errorbar", fun.data = mean_se, width = 0.2, position = position_dodge(0.5))+
  ylab("Disks per 30mx30m plot")+
  xlab("Fire Treatment")+
 theme_bw()+
  theme(panel.background = element_rect(fill="transparent"), panel.border = element_blank())+
  theme(plot.title = element_text(hjust = 0.5), text = element_text(size=12))+
  theme(legend.position = "right", aspect.ratio = 1)+
  theme(axis.line = element_line(color = 'black'))+
  scale_x_discrete(labels= c("BN", "BS", "UN", "US"), )+
  theme(legend.position = "right", aspect.ratio = 1)+
    scale_y_continuous(expand = c(0, 0), limits = c(0, 6))+
  scale_fill_discrete(labels=c('2011', '2016', "2021", "2023"))
```

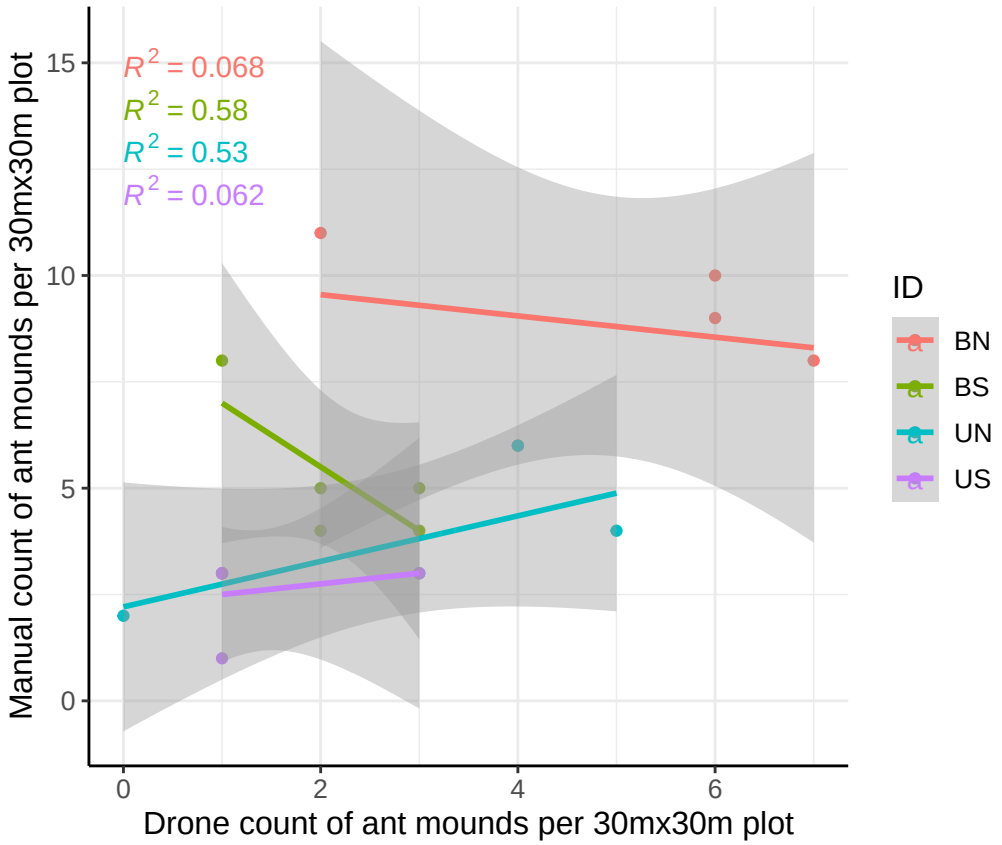
```
## `summarise()` has grouped output by 'Plot', 'Year'. You can override using the
## `.groups` argument.
```



Plot the density of the manually collected against the density of the drone collected

```
d_both |> unite("ID", Burned, Rodents, sep = "", remove = FALSE) |>
  ggplot(aes(Count, Count_m, col = ID))+
  geom_point()+
  geom_smooth(method = "lm")+
  stat_regline_equation(aes(label = after_stat(rr.label)), label.x = 0,
                        label.y = c(15,14,13,12))+
    ylab("Manual count of ant mounds per 30mx30m plot")+
  xlab("Drone count of ant mounds per 30mx30m plot")+
  theme_bw()+
  theme(panel.background = element_rect(fill="transparent"), panel.border = element_blank())+
  theme(plot.title = element_text(hjust = 0.5), text = element_text(size=12))+
  theme(legend.position = "right", aspect.ratio = 1)+
  theme(axis.line = element_line(color = 'black'))
```

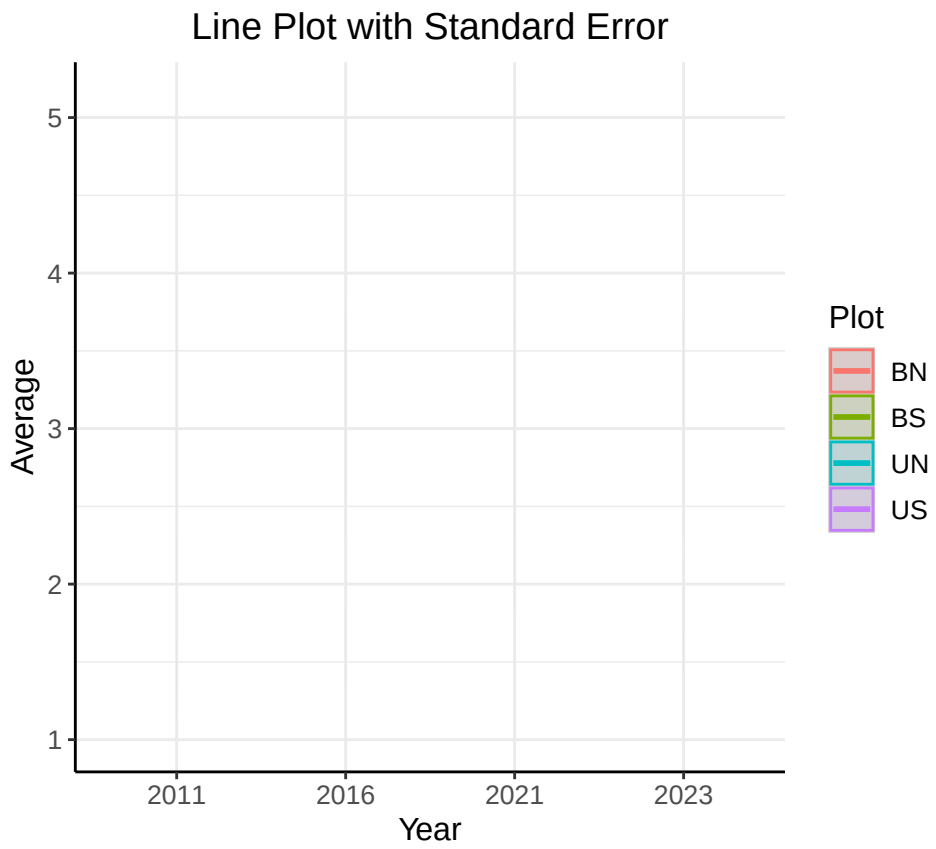
```
## `geom_smooth()` using formula = 'y ~ x'
```



Plot a change in ant mounds over years

```
d21_drone[d21_drone$Count==1,] |>
  group_by(Plot, Year, Block) |>
  dplyr::summarise(mean_count = mean(n())) |>
  group_by(Year, Plot) |>
  dplyr::summarise(ave = mean(mean_count),
                   se = sd(mean_count) / sqrt(n())) |>
    ggplot(aes(x=Year, y = ave, color = Plot))+
     geom_smooth() +
  geom_ribbon(aes(ymin = ave - se, ymax = ave + se, fill = Plot), alpha = 0.1) +
  labs(title = "Line Plot with Standard Error",
       x = "Year",
       y = "Average",
       color = "Plot") +
  theme_bw()+
  theme(panel.background = element_rect(fill="transparent"), panel.border = element_blank())+
  theme(plot.title = element_text(hjust = 0.5), text = element_text(size=12))+
  theme(legend.position = "right", aspect.ratio = 1)+
  theme(axis.line = element_line(color = 'black'))
```

```
## `summarise()` has grouped output by 'Plot', 'Year'. You can override using the
## `.groups` argument.
## `summarise()` has grouped output by 'Year'. You can override using the
## `.groups` argument.
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



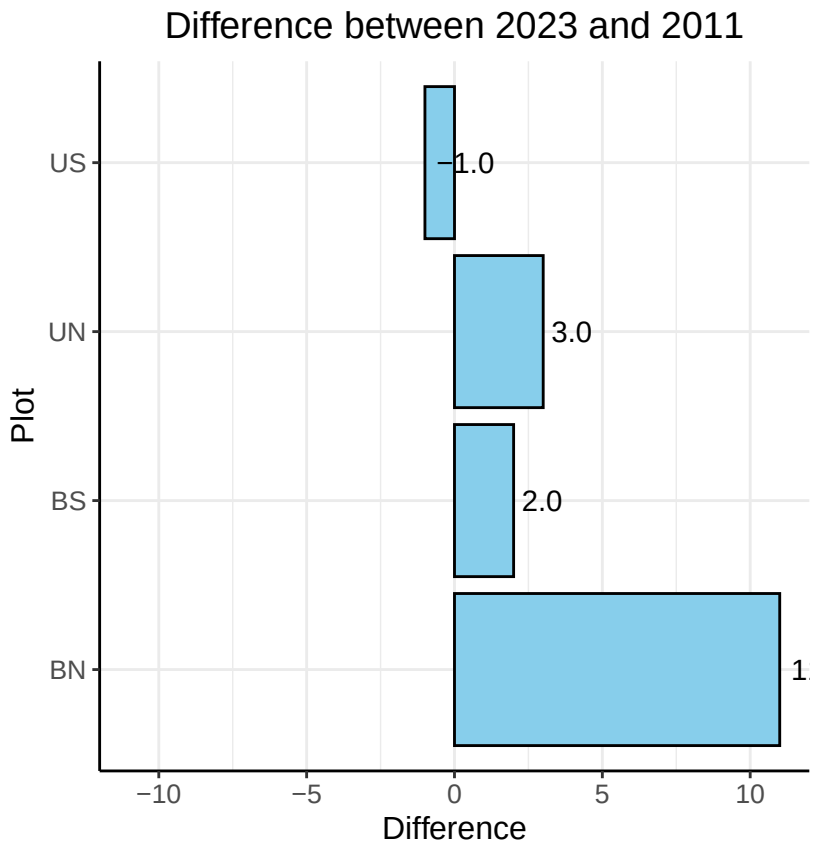
Plot Net Change

```
reshaped <- d21_drone[d21_drone$Count==1,] |>
  filter(Year %in% c(2011, 2023)) |>
  group_by(Plot, Year) |>
  dplyr::summarise(mean_count = mean(n()),
                   se = sd(mean(n())) / sqrt(n()))
```

```
## `summarise()` has grouped output by 'Plot'. You can override using the
## `.groups` argument.
```

```
reshaped <- reshaped %>%
  pivot_wider(names_from = Year, values_from = mean_count)

# Calculate the difference between the values for 2023 and 2011
reshaped <- reshaped %>%
  mutate(difference = `2023` - `2011`) %>%
  select(Plot, difference)


ggplot(reshaped, aes(x = difference, y = Plot)) +
  geom_bar(stat = "identity", fill = "skyblue", color = "black") +
  labs(title = "Difference between 2023 and 2011",
       x = "Difference",
       y = "Plot") +
  geom_text(aes(label = scales::number(difference, accuracy = 0.1)), hjust = -0.2) +
  theme_bw()+
```

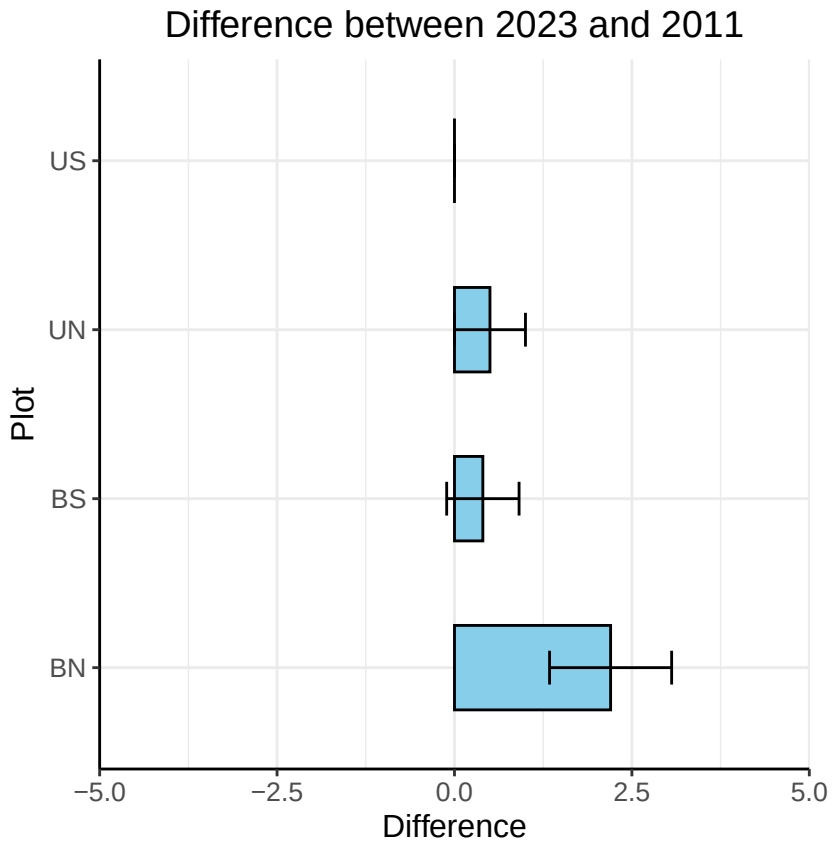
```
  theme(panel.background = element_rect(fill="transparent"), panel.border = element_blank())+
  theme(plot.title = element_text(hjust = 0.5), text = element_text(size=12))+
  theme(legend.position = "right", aspect.ratio = 1)+
  theme(axis.line = element_line(color = 'black'))+
  scale_x_continuous(expand = c(0, 0), limits = c(-12, 12))
```




```
d21_drone[d21_drone$Count==1,] |>
  group_by(Plot, Year, Block) |>
  dplyr::summarise(mean_count = mean(n())) |>
  pivot_wider(names_from = Year, values_from = mean_count) |>
  mutate(difference = `2023` - `2011`) |>
  replace_na(list(difference = 0)) |>
  select(Plot, difference) |>
  ggplot( aes(x = difference, y = Plot)) +
  stat_summary(geom = "bar", fun = mean, position = "dodge", color = "black", fill = "skyblue", width =
  stat_summary(geom = "errorbar", fun.data = mean_se, width = 0.2, position = position_dodge(0.5))+
  #geom_bar(stat = "identity", fill = "skyblue", color = "black") +
  labs(title = "Difference between 2023 and 2011",
       x = "Difference",
       y = "Plot") +
  #geom_text(aes(label = scales::number(difference, accuracy = 0.1)), hjust = -0.2) +
  theme_bw()+
  theme(panel.background = element_rect(fill="transparent"), panel.border = element_blank())+
  theme(plot.title = element_text(hjust = 0.5), text = element_text(size=12))+
  theme(legend.position = "right", aspect.ratio = 1)+
  theme(axis.line = element_line(color = 'black'))+
```

```
  scale_x_continuous(expand = c(0, 0), limits = c(-5, 5))
```

```
## `summarise()` has grouped output by 'Plot', 'Year'. You can override using the
## `.groups` argument.
```




```
#Actual stats instead of just visualizations



reshaped <- d21_drone[d21_drone$Count==1,] |>
  filter(Year %in% c(2011, 2023)) |>
  group_by(Plot, Year, Block) |>
  dplyr::summarise(mean_count = mean(n()),
                   se = sd(mean(n())) / sqrt(n()))
```

```
## `summarise()` has grouped output by 'Plot', 'Year'. You can override using the
## `.groups` argument.
```

```
reshaped[is.na(reshaped)]<- 0

reshaped <- reshaped %>%
  pivot_wider(names_from = Year, values_from = mean_count)

reshaped
```

```
## # A tibble: 18 x 5
## # Groups:   Plot [4]
```

```
##    Plot  Block    se `2011` `2023`
##    <chr> <int> <dbl>  <dbl>  <dbl>
##  1 BN        1     0      3      6
##  2 BN        2     0      1      3
##  3 BN        3     0      3      4
##  4 BN        4     0      1      6
##  5 BN        5     0      1      1
##  6 BS        1     0      2      1
##  7 BS        2     0      1      1
##  8 BS        3     0      1      3
##  9 BS        4     0      2      2
## 10 BS        5     0      1      2
## 11 UN        1     0      4      4
## 12 UN        2     0      4      4
## 13 UN        4     0      1      3
## 14 UN        5     0     NA      1
## 15 US        1     0      1     NA
## 16 US        2     0      1      1
## 17 US        3     0      2      2
## 18 US        5     0      1      1
```

```
# Calculate the difference between the values for 2023 and 2011
reshaped <- reshaped %>%
  mutate(difference = `2023` - `2011`) %>%
  select(Plot, Block, difference)


reshaped$Burned <- substr(reshaped$Plot, 1, 1)
reshaped$Rodents <- substr(reshaped$Plot, 2, 2)

diff_aov <- aov(data= reshaped, difference~ Burned + Rodents +Burned*Rodents)
summary(diff_aov)
```

```
##                Df Sum Sq Mean Sq F value Pr(>F)
## Burned          1  3.504   3.504   1.855 0.1982
## Rodents         1  7.563   7.563   4.004 0.0685 .
## Burned:Rodents  1  1.204   1.204   0.638 0.4401
## Residuals      12 22.667   1.889
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 2 observations deleted due to missingness
```

Joshs Data from 2016

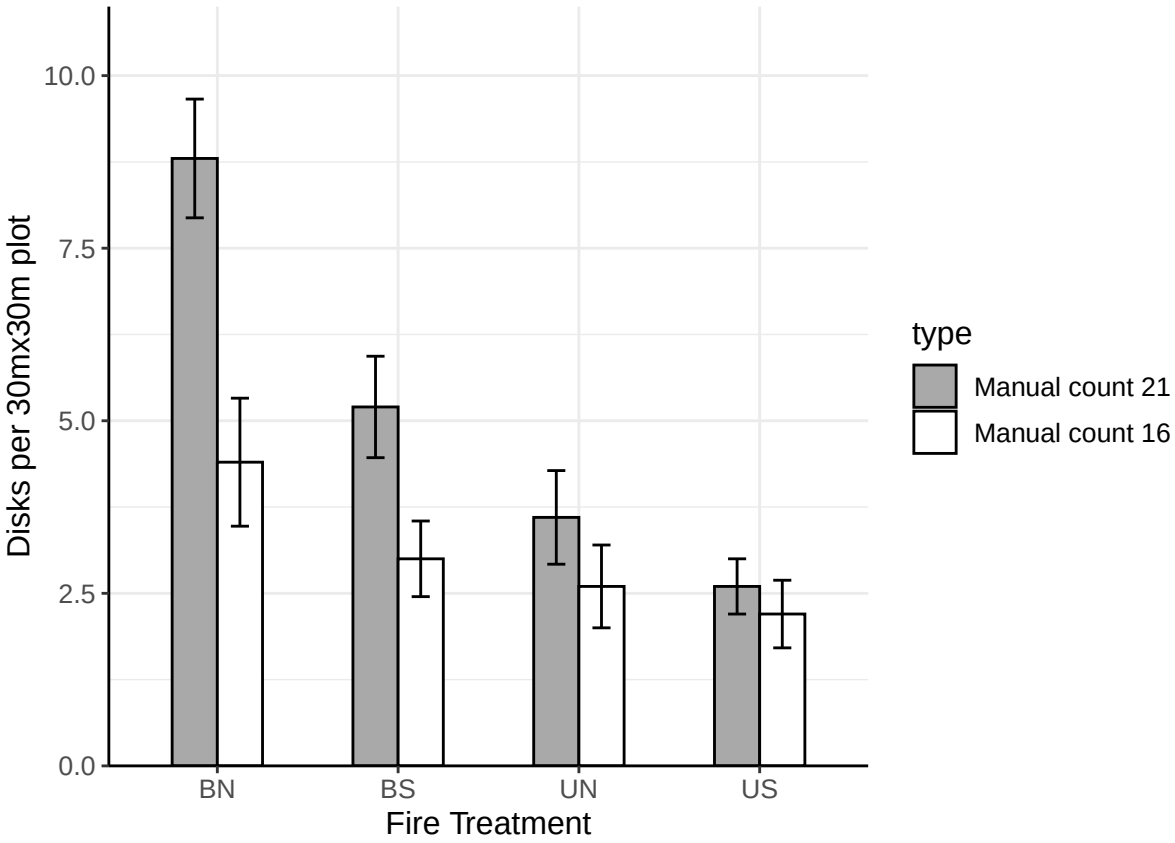
```
manual_d_16 <- read.csv("C:/Users/ryanp/Documents/Dissertation/Chapter1_Ants/Josh data/Ant mound density
head(manual_d_16)
```

```
##   block plot burn small.mammal density lndensity     sqrt ln1density
## 1     1   us    u            s       3  1.098612 1.732051  1.3862944
## 2     1   un    u            n       4  1.386294 2.000000  1.6094379
## 3     1   bs    b            s       4  1.386294 2.000000  1.6094379
## 4     1   bn    b            n       6  1.791759 2.449490  1.9459101
## 5     2   us    u            s       1  0.000000 1.000000  0.6931472
## 6     2   un    u            n       4  1.386294 2.000000  1.6094379
```

```
colnames(manual_d_16)[1]<- "Block"
colnames(manual_d_16)[3]<- "Burned"
colnames(manual_d_16)[4]<- "Rodents"
manual_d_16$Burned <- toupper(manual_d_16$Burned)
manual_d_16$Rodents <- toupper(manual_d_16$Rodents)
```

Compare 2016 and 2021 manuals to their classifications

```
d_both <- merge(d_both, manual_d_16, by = c("Block", "Burned", "Rodents"), all.x = TRUE)
d_both <- d_both[, c("Block", "Burned", "Rodents", "Count", "Count_m", "density")]

d_both %>% unite("ID", Burned, Rodents, remove = FALSE) %>%
  pivot_longer(cols = c(Count_m, density), names_to = "type", values_to = "n") %>%
ggplot(aes(x= ID, y = n, fill = type))+
  #geom_bar(stat = "identity", position = position_dodge())+
  stat_summary(geom = "bar", fun = mean, position = "dodge", color = "black", width = 0.5) +
  stat_summary(geom = "errorbar", fun.data = mean_se, width = 0.2, position = position_dodge(0.5))+
  ylab("Disks per 30mx30m plot")+
  xlab("Fire Treatment")+
 theme_bw()+
  theme(panel.background = element_rect(fill="transparent"), panel.border = element_blank())+
  theme(plot.title = element_text(hjust = 0.5), text = element_text(size=12))+
  theme(legend.position = "right", aspect.ratio = 1)+
  theme(axis.line = element_line(color = 'black'))+
  scale_x_discrete(labels= c("BN", "BS", "UN", "US"), )+
  scale_fill_manual(values = c("dark grey","white"),labels=c('Manual count 21', "Manual count 16"))+
  theme(legend.position = "right", aspect.ratio = 1)+
    scale_y_continuous(expand = c(0, 0), limits = c(0, 11))
```


10.0
7.5
5.0
2.5
0.0
Disks per 30mx30m plot
BN
BS
UN
US
Fire Treatment
type
Manual count 21
Manual count 16