Code: Sampling bias exaggerates a textbook example of a trophic cascade

E.M. Brice, E.J. Larsen, D.R. MacNulty

10/18/2021

The following code and data can be used to replicate the results for “Sampling bias exaggerates a textbook example of a trophic cascade.”

R Code

Install necessary packages

```
## Install required packages (if not already installed)
install.packages("tidyverse") # for data manipulation
install.packages("gridExtra") # for plotting multi-panel figures
install.packages("lme4") # for mixed models
install.packages("lmerTest") # for mixed models (generates p-values)
install.packages("AICcmodavg") # for model comparison
install.packages("margins") # for average marginal effects and predictions
install.packages("emmeans") # for comparison of categorical effects

## Load required packages
library(tidyverse)
library(gridExtra)
library(lme4)
library(lmerTest)
library(AICcmodavg)
library(margins)
library(emmeans)
```

Read and format data

The dataset “aspen.df.all” has 18,792 records, including 18,623 records of individual young aspen (plants $\geq$ 1 year-old & $\leq$ 600 cm) and 169 records of plots with no young aspen (“zero plots”). Records of individual young aspen (N = 18,623) were used in the majority of analyses (Fig. 2-5a,b), including generalized linear mixed models (GLMMs) that tested how the effect of year on browsing, height, and recruitment of stems differed by sampling method. This dataset was also used to model the effect of stem height on browsing to estimate the preferred browse height (PBH) and browse escape height (BEH). The full dataset that includes plots with no young aspen was only used to calculate the percentage of plots and stands each year with median heights greater than 200 (Fig. 5c) or 300 cm (Fig. 5d).

The dataframe has the following 6 columns:

1) Plot: individual identifier for each of 113 plots distributed randomly across the study area. Each plot was a 1×20 m belt transect located randomly within an aspen stand
2) Year: year in which aspen was sampled

3) Tree: individual identifier for each stem within a plot
4) Browse: denotes the browsing status (browsed = 1,unbrowsed = 0) of the leader (tallest) stem. A leader was 'browsed' if its growth from the previous growing season had been eaten
5) Height: height (cm) of the leader stem of each individual aspen
6) Type: sampling method. Every young aspen within a plot is a "random" stem, and each of the five tallest young aspen within the stand is a "5T" stem.

The following code reads and formats the aspen data for the subsequent analysis. Specifically, it:

- Reads the full dataset and renames the first column in case of special characters (sometimes occurs on PCs)
- Subsets the full dataset to only include plots with young aspen (i.e., removes "zero plots")
- Makes year into continuous and categorical variables
- Codes sampling type as 1 for 5T stems and 0 for random stems

```
## Set working directory
setwd()

## Read aspen data
aspen.df.all <- read.csv("Aspen_Data.csv")

# Rename the first column in case there are special characters
colnames(aspen.df.all)[1] <- "Plot"

## Get rid of zero plots
aspen.df <- aspen.df.all %>%
  filter(Tree != 0)

# Make a continuous year effect
aspen.df$Year.c <- aspen.df$Year - 2006

# Make year a factor
aspen.df$Year.f <- as.factor(aspen.df$Year)

## Code type as 5T = 1, Random = 0
aspen.df <- aspen.df %>%
  mutate(Type2 = ifelse(Type == "5T",1,0))
```

Figure 2: Browse & Height Distributions

The following code produces probability densities of browse occurrence and height of the five tallest young aspen and randomly sampled young aspen in northern Yellowstone National Park during the first and last years of the study (2007, 2017), as shown in Figure 2.

Specifically, the code does the following:

- Calculates the percent of browsed stems in each plot for both random and 5T stems in 2007 and 2017
- Categorizes random stems into the 6 discrete browsing values that the 5T can take (e.g., 0, 20, 40, 60, 80, 100%)
- Calculates the density of each browsing category by type and year
- Plots the browse distributions for 2007 and 2017
- Plots the height distributions for 2007 and 2017

```
## Browse Plots ####
## Calculate % browse per plot and categorize random stems into
## same 6 % options as five tallest (0, 20, 40, 60, 80, 100%)
browse.dist.df <- aspen.df %>%
```

```r
  filter(Year == 2007 | Year == 2017) %>%
  group_by(Type,Plot,Year) %>%
  summarize(Browse = mean(Browse,na.rm=T)*100) %>%
  mutate(BrowseCat = cut(Browse, breaks = c(-1,10,30,50,70,90,101),
                        labels = c(0,20,40,60,80,100), right = F)) %>%
  mutate(BrowseCat = as.numeric(as.character(BrowseCat)))

## To store figure panels:
fig2.list <- list()

### 1) 2007 ####
# Calculate the density of each browse category by type
a <- browse.dist.df %>%
  filter(Year == 2007) %>%
  filter(Type == "Random") %>%
  ggplot(aes(x=BrowseCat,group=Type))+
  geom_histogram(aes(y=..density..,fill=Type),bins=6,color="black")

b <- browse.dist.df %>%
  filter(Year == 2007) %>%
  filter(Type == "5T") %>%
  ggplot(aes(x=BrowseCat,group=Type))+
  geom_histogram(aes(y=..density..,fill=Type),bins=6,color="black")

# Build the initial plot and extract density information
a.plot <- ggplot_build(a)
a.y <- a.plot[["data"]][[1]][["y"]]
b.plot <- ggplot_build(b)
b.y <- b.plot[["data"]][[1]][["y"]]

# Dataframe for the figures
c <- data.frame(Type=factor(c(rep("Random",6),
                              rep("Five Tallest",6)),
                            levels = c("Random","Five Tallest")),
                Browse=c(rep(seq(0,100,20),2)),
                Density=c(a.y,b.y))

fig2.list$a <- ggplot(data= c,
       aes(x=Browse,y = Density,group=Type))+
  geom_bar(aes(fill=Type),stat="identity",position="identity",color="black",
           alpha=0.3)+
  scale_fill_manual(values=c("darkblue","red"),name="",
                    labels=c("Random stems","Five tallest stems"))+
  ylab("Probability density")+
  xlab("")+
  ggtitle("(a) 2007")+
  scale_x_continuous(breaks=seq(0,100,20),labels=seq(0,100,20))+
  scale_y_continuous(breaks=seq(0,0.03,0.01),labels=scales::number_format(accuracy = 0.01),
                     limits =c(0,0.032), expand = c(0,0)) +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        panel.background = element_blank(), axis.line = element_line(colour = "black"),
        legend.title = element_blank(),legend.key.size = unit(0.5, "cm"),
        legend.justification = c("left","top"),legend.position = c(0.01,1.05),
```

```r
        legend.text = element_text(size=10),
        axis.title.x = element_text(size=11),
        axis.title.y = element_text(size=11),
        plot.title = element_text(size=11, face = "bold"),
        axis.text = element_text(size=10, color="black"),
        plot.margin=unit(c(c(0.5, 0.2, 0, 0.5)), units="line"))

### 2) 2017 ####
a <- browse.dist.df %>%
  filter(Year == 2017) %>%
  filter(Type == "Random") %>%
  ggplot(aes(x=BrowseCat,group=Type))+
  geom_histogram(aes(y=..density..,fill=Type),bins=6,color="black")

b <- browse.dist.df %>%
  filter(Year == 2017) %>%
  filter(Type == "5T") %>%
  ggplot(aes(x=BrowseCat,group=Type))+
  geom_histogram(aes(y=..density..,fill=Type),bins=6,color="black")

a.plot <- ggplot_build(a)
a.y <- a.plot[["data"]][[1]][["y"]]
b.plot <- ggplot_build(b)
b.y <- b.plot[["data"]][[1]][["y"]]


c <- data.frame(Type=factor(c(rep("Random",6),
                              rep("Five Tallest",6)),
                            levels = c("Random","Five Tallest")),
                Browse=c(rep(seq(0,100,20),2)),
                Density=c(a.y,b.y))

fig2.list$c <- ggplot(data= c,
       aes(x=Browse,y = Density,group=Type))+
  geom_bar(aes(fill=Type),stat="identity",position="identity",color="black",
           alpha=0.3)+
  scale_fill_manual(values=c("darkblue","red"),name="",
                    labels=c("Random stems","Five tallest stems"))+
  ylab("Probability density")+
  xlab("% stems browsed")+
  ggtitle("(c) 2017")+
  guides(fill = FALSE)+
  scale_x_continuous(breaks=seq(0,100,20),labels=seq(0,100,20))+
  scale_y_continuous(breaks=seq(0,0.03,0.01),labels=scales::number_format(accuracy = 0.01),
                     limits =c(0,0.032), expand = c(0,0)) +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        panel.background = element_blank(), axis.line = element_line(colour = "black"),
        legend.title = element_blank(),legend.key.size = unit(0.5, "cm"),
        legend.justification = c("left","top"),legend.position = c(0.01,1.05),
        legend.text = element_text(size=10),
        axis.title.x = element_text(size=11),
        axis.title.y = element_text(size=11),
        plot.title = element_text(size=11, face = "bold"),
```

```r
        axis.text = element_text(size=10, color="black"),
        plot.margin=unit(c(c(0, 0.2, 0.5, 0.5)), units="line"))

### Height Plots ####
## 1) 2007  ####
fig2.list$b <- aspen.df %>%
  filter(Year == 2007) %>%
  ggplot(aes(x=Height,group=Type))+
  geom_histogram(aes(y=..density..,fill=Type),binwidth=10,color="black",alpha=0.35,
                 position="identity")+
  scale_fill_manual(values=c("red","darkblue"),labels = c("Five tallest stems",
                                                          "Random stems"))+
  ylab("")+
  xlab("")+
  ggtitle("(b) 2007")+
  guides(fill = guide_legend(reverse = TRUE,label.position = "left"))+
  scale_x_continuous(breaks=seq(0,400,100),labels=seq(0,400,100),limits = c(0,405))+
  scale_y_continuous(breaks=seq(0,0.015,0.005),labels=scales::number_format(accuracy = 0.001),
                     limits =c(0,0.016), expand = c(0,0)) +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        panel.background = element_blank(), axis.line = element_line(colour = "black"),
        legend.title = element_blank(),legend.key.size = unit(0.5, "cm"),
        legend.justification = c("right","top"),legend.position = c(1,1.05),
        legend.text = element_text(size=10),
        axis.title.x = element_text(size=11),
        axis.title.y = element_text(size=11),
        plot.title = element_text(size=11, face = "bold"),
        axis.text = element_text(size=10, color="black"),
        plot.margin=unit(c(c(0.5, 0.5, 0, 0)), units="line"))

## 2) 2017  ####
fig2.list$d <- aspen.df %>%
  filter(Year == 2017) %>%
  ggplot(aes(x=Height,group=Type))+
  geom_histogram(aes(y=..density..,fill=Type),binwidth=10,color="black",alpha=0.35,
                 position="identity")+
  scale_fill_manual(values=c("red","darkblue"),labels = c("Five tallest stems",
                                                          "Random stems"))+
  ylab("")+
  xlab("Stem height (cm)")+
  ggtitle("(d) 2017")+
  guides(fill = FALSE)+
  scale_x_continuous(breaks=seq(0,400,100),labels=seq(0,400,100),limits = c(0,405))+
  scale_y_continuous(breaks=seq(0,0.015,0.005),labels=scales::number_format(accuracy = 0.001),
                     limits =c(0,0.016), expand = c(0,0)) +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        panel.background = element_blank(), axis.line = element_line(colour = "black"),
        axis.title.y = element_text(size=11),
        plot.title = element_text(size=11, face = "bold"),
        axis.text = element_text(size=10, color="black"),
        plot.margin=unit(c(c(0, 0.5, 0.5, 0)), units="line"))

## Plot it
```

```r
grid.arrange(fig2.list$a,fig2.list$b,fig2.list$c,fig2.list$d,ncol=2)

# Can save figure with:
# tiff("Fig2.tiff",units = "in",width=6.8,height=5.5, res = 150)
# grid.arrange(fig2.list$a,fig2.list$b,fig2.list$c,fig2.list$d,ncol=2)
# dev.off()
```

Figure 3: Browse and Height Models

We used GLMMs to test how the effect of year on browsing and stem height differed by sampling method. We treated the stem as the unit of analysis and used GLMMs with a Bernoulli distribution and a logit link to separately analyze the probability a stem was browsed (1 = browsed; 0 = not browsed), and GLMMs with a gamma distribution and a log link to analyze stem height (cm), which took only non-negative values that were strongly right-skewed. Year was an integer that ranged from 1 (2007) to 11 (2017) and sampling method was a dummy variable (1 = 5T sampling; 0 = random sampling). All GLMMs included a random intercept for stand identity and a random slope for year. Models were fit with unstructured covariance and the Laplace approximation, which are the default settings of `lme4`. We used likelihood ratio tests to compare models with and without an interaction between year and sampling type.

We estimated average marginal effects (AMEs) from GLMMs using the `margins` package to quantify and compare annual changes of 5T and random stems. AMEs describe the average effect of changes in explanatory variables on the change in a response variable and are useful for interpreting generalized linear models. We used the `emmeans` package to calculate z-scores to test how the AMEs of year on browsing and stem height differed between 5T and random stems.

Finally, we calculated population-averaged fitted values from best-fit GLMMs by deriving marginal expectations of the responses averaged over the random effects but conditional on the observed variables (again using the `margins` package). We refit year as a categorical factor and plotted the associated fitted values to illustrate the distribution of the underlying data after controlling for stand-level heterogeneity, and to assess the negative correlation between mean annual browsing and mean annual stem height. Models with categorical year included a random intercept for plot and did not include a random slope for year.

The code below does the following:

- Models browse as a function of the interaction between sampling method and continuous year
- Models browse as a function of sampling method and continuous year without an interaction
- Conducts a likelihood ratio test between the two models
- Models browse as a function of the interaction between sampling method and categorical year
- Calculates the average marginal effects (AMEs) of annual changes in browse probability depending on sampling method
- Conducts a z-test to compare AMEs of year on browse probability based on sampling method
- Calculates the marginal predictions of browse probabiltiy for each year and sampling method for both models (year as a continuous vs. categorical predictor)
- Repeats these steps for the height analysis

```r
#----------------------- * * * * PREP * * * * ----------------------------

# Optimizer to use to ensure convergence
br.control <- glmerControl(optimizer="bobyqa",optCtrl=list(maxfun=2e5))

## If model still fails to converge, switch to Nelder-Mead optimizer (PBH models)
br.control.2 <- glmerControl(optimizer="Nelder_Mead",optCtrl=list(maxfun=2e5))

mod.list <- list()

#---------------------- *  *  *  *  * BROWSE *  *  *  *  * ----------------
```

```r
mod.list$Browse <- list()

# Continuous Year --------------------------------------------------------
mod.list$Browse$Con <- glmer(Browse ~ Type2*Year.c +
                               (Year.c|Plot),
                             data=aspen.df, family = binomial,
                             control = br.control)
summary(mod.list$Browse$Con)

## 95% Confidence interval for interaction term (B +/- 1.96*SE)
summary(mod.list$Browse$Con)$coef[4,1] - 1.96*summary(mod.list$Browse$Con)$coef[4,2]
summary(mod.list$Browse$Con)$coef[4,1] + 1.96*summary(mod.list$Browse$Con)$coef[4,2]


## No interaction
mod.list$Browse$Con.2 <- glmer(Browse ~ Type2+Year.c +
                                 (Year.c|Plot),
                               data=aspen.df, family = binomial,
                               control = br.control)

# Conduct likelihood ratio test
anova(mod.list$Browse$Con.2,mod.list$Browse$Con)


# Categorical Year -------------------------------------------------------
# Model takes ~ 30 seconds to run
mod.list$Browse$Cat <- glmer(Browse ~ Type2*Year.f + (1|Plot),
                             data=aspen.df, family = binomial,
                             control = br.control)
summary(mod.list$Browse$Cat)



# Average Marginal Effects -----------------------------------------------
ame.list <- list()
ame.list$Browse <- list()

# Dataframe of all combinations of plot + year to predict on
new.df.all <- expand.grid(Year.c = 1:11,
                          Plot = unique(aspen.df$Plot),
                          Type2 = c(0,1)) %>%
  mutate(Year = Year.c + 2006,
         Year.f = as.factor(Year),
         Type = ifelse(Type2 == 0, "Random","5T")) %>%
  mutate(Year.f = ifelse(Year == 2015,NA,paste(Year.f)))


# To get the AME of browse over time by the two types
summary(margins(model = mod.list$Browse$Con,
                data =new.df.all[new.df.all$Type2==1,],variables = "Year.c"))

summary(margins(model = mod.list$Browse$Con,
                data =new.df.all[new.df.all$Type2==0,],variables = "Year.c"))
```

```r
# Z-test to compare AMEs of year on browse based on type
emtrends(mod.list$Browse$Con, pairwise ~ Type2, var = "Year.c")


# Marginal Predictions ----------------------------------------------------
# Continuous
ame.list$Browse$Pred.con <- prediction(mod.list$Browse$Con, type = "response",
                                       data = new.df.all)

# Calculate average margin by year + type, including 95% CI
ame.list$Browse$Pred.con <- ame.list$Browse$Pred.con %>%
  group_by(Type2, Year.c) %>%
  summarize(Mean = mean(fitted),
            N = n(),
            SD = sd(fitted),
            SE = SD/sqrt(N),
            Lower = Mean - 1.96*SE,
            Upper = Mean + 1.96*SE) %>%
  select(Year.c,Type2,Mean,Lower,Upper)


# Categorical
ame.list$Browse$Pred.cat <- prediction(mod.list$Browse$Cat, type = "response",
                                       data = new.df.all)

# Calculate average margin by year + type, including 95% CI
ame.list$Browse$Pred.cat <- ame.list$Browse$Pred.cat %>%
  group_by(Type2, Year.f) %>%
  summarize(Mean = mean(fitted),
            N = n(),
            SD = sd(fitted),
            SE = SD/sqrt(N),
            Lower = Mean - 1.96*SE,
            Upper = Mean + 1.96*SE) %>%
  select(Year.f,Type2,Mean,Lower,Upper)


#---------------------- *   *   *   *   * HEIGHT *   *   *   *   * ----------------
mod.list$Height <- list()

# Continuous Year ---------------------------------------------------------
mod.list$Height$Con <- glmer(Height ~ Type2*Year.c +
                               (Year.c|Plot),
                             data=aspen.df, family = Gamma(link = "log"),
                             control = br.control)
summary(mod.list$Height$Con)

## 95% Confidence interval for interaction term (B +/- 1.96*SE)
summary(mod.list$Height$Con)$coef[4,1] - 1.96*summary(mod.list$Height$Con)$coef[4,2]
summary(mod.list$Height$Con)$coef[4,1] + 1.96*summary(mod.list$Height$Con)$coef[4,2]
```

```r
## No interaction
mod.list$Height$Con.2 <- glmer(Height ~ Type2+Year.c +
                                 (Year.c|Plot),
                               data=aspen.df, family = Gamma(link = "log"),
                               control = br.control)

anova(mod.list$Height$Con.2,mod.list$Height$Con)


# Categorical Year --------------------------------------------------------
# Model takes ~ 1 minute to run
mod.list$Height$Cat <- glmer(Height ~ Type2*Year.f + (1|Plot),
                             data=aspen.df, family = Gamma(link = "log"),
                             control = br.control)
summary(mod.list$Height$Cat)


# Average Marginal Effects ------------------------------------------------
ame.list$Height <- list()

# To get the AME of height over time by the two types
summary(margins(model = mod.list$Height$Con,
                data =new.df.all[new.df.all$Type2==1,],variables = "Year.c"))

summary(margins(model = mod.list$Height$Con,
                data =new.df.all[new.df.all$Type2==0,],variables = "Year.c"))


# Z-test to compare AMEs of year on height based on type
emtrends(mod.list$Height$Con, pairwise ~ Type2, var = "Year.c")


# Marginal Predictions ----------------------------------------------------
# Continuous
ame.list$Height$Pred.con <- prediction(mod.list$Height$Con, type = "response",
                                       data = new.df.all)


# Calculate average margin by year + type, including 95% CI
ame.list$Height$Pred.con <- ame.list$Height$Pred.con %>%
  group_by(Type2, Year.c) %>%
  summarize(Mean = mean(fitted),
            N = n(),
            SD = sd(fitted),
            SE = SD/sqrt(N),
            Lower = Mean - 1.96*SE,
            Upper = Mean + 1.96*SE) %>%
  select(Year.c,Type2,Mean,Lower,Upper)


# Categorical
ame.list$Height$Pred.cat <- prediction(mod.list$Height$Cat, type = "response",
                                       data = new.df.all)
```

```r
# Calculate average margin by year + type, including 95% CI
ame.list$Height$Pred.cat <- ame.list$Height$Pred.cat %>%
  group_by(Type2, Year.f) %>%
  summarize(Mean = mean(fitted),
            N = n(),
            SD = sd(fitted),
            SE = SD/sqrt(N),
            Lower = Mean - 1.96*SE,
            Upper = Mean + 1.96*SE) %>%
  select(Year.f,Type2,Mean,Lower,Upper)
```

We regressed the categorical predictions for each type against each other with the following code, which:

- Creates a dataframe of the fitted values from the models of browse and height as functions of categorical year and sampling method
- Subsets the data into just 5T stems
- Subsets the data into just random stems
- Calculates Pearson's correlation of browse and height for the two sampling methods

```r
browse.height <- list()

browse.height$DF <- ame.list$Height$Pred.cat %>%
  rename(Height_mu = Mean,
         Height_low = Lower,
         Height_upp = Upper) %>%
  ungroup()%>%
  mutate(Browse_mu = ame.list$Browse$Pred.cat$Mean,
         Browse_low = ame.list$Browse$Pred.cat$Lower,
         Browse_upp = ame.list$Browse$Pred.cat$Upper)


browse.height$DF.5T <- browse.height$DF %>%
  filter(Type2 == 1)

browse.height$DF.rand <- browse.height$DF %>%
  filter(Type2 == 0)

## Correlations

cor.test(browse.height$DF.5T$Browse_mu,browse.height$DF.5T$Height_mu)

cor.test(browse.height$DF.rand$Browse_mu,browse.height$DF.rand$Height_mu)
```

The following code generates Figure 3.

- Create a dataframe of the fitted values of browse probability from models with continuous and categorical year
- Create a dataframe of the fitted values of stem height from models with continuous and categorical year
- Create a dataframe of the fitted values of browse probability and stem height from the models with categorical year
- Plot each panel of Fig. 3
- Combine panels into one figure

```r
#---------------------- *  *  *  *  * FIGURE 3 *  *  *  *  * ----------------
# Data ---------------------------------------------------------------
```

```r
fig3.list <- list()
fig3.list$DF <- list()

fig3.list$DF$a <- data.frame(Year = ame.list$Browse$Pred.con$Year.c+2006,
                             Type = ame.list$Browse$Pred.con$Type2,
                             ContBrowse = ame.list$Browse$Pred.con$Mean,
                             ContLower = ame.list$Browse$Pred.con$Lower,
                             ContUpper = ame.list$Browse$Pred.con$Upper,
                             CatBrowse = c(ame.list$Browse$Pred.cat$Mean[1:8],
                                           NA,
                                           ame.list$Browse$Pred.cat$Mean[9:10],
                                           ame.list$Browse$Pred.cat$Mean[12:19],
                                           NA,
                                           ame.list$Browse$Pred.cat$Mean[20:21]),
                             CatLower = c(ame.list$Browse$Pred.cat$Lower[1:8],
                                          NA,
                                          ame.list$Browse$Pred.cat$Lower[9:10],
                                          ame.list$Browse$Pred.cat$Lower[12:19],
                                          NA,
                                          ame.list$Browse$Pred.cat$Lower[20:21]),
                             CatUpper = c(ame.list$Browse$Pred.cat$Upper[1:8],
                                          NA,
                                          ame.list$Browse$Pred.cat$Upper[9:10],
                                          ame.list$Browse$Pred.cat$Upper[12:19],
                                          NA,
                                          ame.list$Browse$Pred.cat$Upper[20:21])) %>%
  mutate(Type = ifelse(Type == 0, "Random","5T"))

fig3.list$DF$b <- data.frame(Year = ame.list$Height$Pred.con$Year.c+2006,
                             Type = ame.list$Height$Pred.con$Type2,
                             ContHeight = ame.list$Height$Pred.con$Mean,
                             ContLower = ame.list$Height$Pred.con$Lower,
                             ContUpper = ame.list$Height$Pred.con$Upper,
                             CatHeight = c(ame.list$Height$Pred.cat$Mean[1:8],
                                           NA,
                                           ame.list$Height$Pred.cat$Mean[9:10],
                                           ame.list$Height$Pred.cat$Mean[12:19],
                                           NA,
                                           ame.list$Height$Pred.cat$Mean[20:21]),
                             CatLower = c(ame.list$Height$Pred.cat$Lower[1:8],
                                          NA,
                                          ame.list$Height$Pred.cat$Lower[9:10],
                                          ame.list$Height$Pred.cat$Lower[12:19],
                                          NA,
                                          ame.list$Height$Pred.cat$Lower[20:21]),
                             CatUpper = c(ame.list$Height$Pred.cat$Upper[1:8],
                                          NA,
                                          ame.list$Height$Pred.cat$Upper[9:10],
                                          ame.list$Height$Pred.cat$Upper[12:19],
                                          NA,
                                          ame.list$Height$Pred.cat$Upper[20:21])) %>%
  mutate(Type = ifelse(Type == 0, "Random","5T"))
```

```r
fig3.list$DF$c <- browse.height$DF %>%
  rename(Height = Height_mu,
          HtLower= Height_low,
         HtUpper= Height_upp,
         Browse = Browse_mu,
         BrLower= Browse_low,
         BrUpper= Browse_upp) %>%
  mutate(Type = ifelse(Type2 == 0, "Random","5T")) %>%
  filter(!is.na(Year.f))



fig3.list$Plots <- list()

# Panel a ----------------------------------------------------------------
fig3.list$Plots$a <- ggplot()+
  geom_ribbon(data=fig3.list$DF$a,
              aes(x=Year,ymin=ContLower, ymax = ContUpper,
                  group = Type, fill=Type),alpha = 0.3,show.legend = F)+
  geom_line(data=fig3.list$DF$a[which(fig3.list$DF$a$Year<2015),],
            aes(x=Year,y=ContBrowse,group=Type,color=Type),size=0.7)+
  geom_line(data=fig3.list$DF$a[which(fig3.list$DF$a$Year>2013 &fig3.list$DF$a$Year<2017),],
            aes(x=Year,y=ContBrowse,group=Type,color=Type),lty="dashed",size=0.7,
            show.legend = F)+
  geom_line(data=fig3.list$DF$a[which(fig3.list$DF$a$Year>2015),],
            aes(x=Year,y=ContBrowse,group=Type,color=Type),size=0.7)+
  geom_errorbar(data=fig3.list$DF$a,
                aes(x=Year,ymin=CatLower,ymax=CatUpper),width=0)+
  geom_point(data=fig3.list$DF$a,aes(x=Year,y=CatBrowse,fill=Type),color="black",
             size=2,shape=21)+
  scale_fill_manual(name="",values=c("#CC2529","#396AB1"),labels=c("Five tallest stems","Random stems")
  scale_color_manual(name="",values=c("#CC2529","#396AB1"),labels=c("Five tallest stems","Random stems"
  scale_x_continuous(breaks=seq(2007,2017,2),labels=seq(2007,2017,2),
                     limits = c(2007,2017))+
  scale_y_continuous(breaks=seq(0,1,0.2),labels=scales::number_format(accuracy = 0.1),
                     limits = c(0,1.0),expand = c(0, 0))+
  xlab("Year")+
  ylab("Browse probability")+
  ggtitle("(a)")+
  guides(fill=guide_legend(reverse = TRUE),color=guide_legend(reverse = TRUE))+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        panel.background = element_blank(), axis.line = element_line(colour = "black"),
        legend.key=element_blank(),legend.position = c(0.01,0.01),
        legend.key.size = unit(0.3, "cm"),legend.title = element_blank(),
        legend.text = element_text(size=10),
        legend.justification = c("left", "bottom"),
        plot.title = element_text(size=11,face="bold"),
        axis.title = element_text(size=11),
        axis.title.x=element_text(margin = margin(t = 5, r = 0, b = 0, l = 0)),
        axis.title.y=element_text(margin = margin(t = 0, r = 5, b = 0, l = 0)),
        axis.text.x = element_text(size=10, color="black"),
        axis.text.y=element_text(color="black",size=10),
        plot.margin=unit(c(c(0.5, 0.5, 0, 0.5)), units="line"))
```

```r
# Panel b -----------------------------------------------------------
fig3.list$Plots$b <- ggplot()+
  geom_ribbon(data=fig3.list$DF$b,
              aes(x=Year,ymin=ContLower, ymax = ContUpper,
                  group = Type, fill=Type),alpha = 0.3,show.legend = F)+
  geom_line(data=fig3.list$DF$b[which(fig3.list$DF$b$Year<2015),],
            aes(x=Year,y=ContHeight,group=Type,color=Type),size=0.7)+
  geom_line(data=fig3.list$DF$b[which(fig3.list$DF$b$Year>2013 &fig3.list$DF$b$Year<2017),],
            aes(x=Year,y=ContHeight,group=Type,color=Type),lty="dashed",size=0.7,
            show.legend = F)+
  geom_line(data=fig3.list$DF$b[which(fig3.list$DF$b$Year>2015),],
            aes(x=Year,y=ContHeight,group=Type,color=Type),size=0.7)+
  geom_errorbar(data=fig3.list$DF$b,aes(x=Year,ymin=CatLower,ymax=CatUpper),width=0)+
  geom_point(data=fig3.list$DF$b,aes(x=Year,y=CatHeight,fill=Type),color="black",size=2,shape=21)+
  scale_fill_manual(name="",values=c("#CC2529","#396AB1"),labels=c("Five tallest stems","Random stems")
  scale_color_manual(name="",values=c("#CC2529","#396AB1"),labels=c("Five tallest stems","Random stems"
  scale_x_continuous(breaks=seq(2007,2017,2),labels=seq(2007,2017,2),
                     limits = c(2007,2017))+
  scale_y_continuous(breaks=seq(0,400,100),labels=seq(0,400,100),
                     limits = c(0,405),expand = c(0, 0))+
  xlab("Year")+
  ylab("Stem height (cm)")+
  ggtitle("(b)")+
  guides(fill=guide_legend(reverse = TRUE),color=guide_legend(reverse = TRUE))+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        panel.background = element_blank(), axis.line = element_line(colour = "black"),
        legend.key=element_blank(),legend.position = c(0.01,1.05),
        legend.justification = c("left", "top"),
        legend.key.size = unit(0.3, "cm"),legend.title = element_blank(),
        legend.text = element_text(size=10),
        plot.title = element_text(size=11,face="bold"),
        axis.title = element_text(size=11),
        axis.title.x=element_text(margin = margin(t = 5, r = 0, b = 0, l = 0)),
        axis.title.y=element_text(margin = margin(t = 0, r = 5, b = 0, l = 0)),
        axis.text.x = element_text(size=10, color="black"),
        axis.text.y=element_text(color="black",size=10),
        plot.margin=unit(c(c(0, 0.5, 0, 0.3)), units="line"))


# Panel c -----------------------------------------------------------
fig3.list$Plots$c <- ggplot(data=fig3.list$DF$c)+
  geom_errorbar(aes(x=Browse,ymin=HtLower,ymax=HtUpper),width=0)+
  geom_errorbarh(aes(x=Browse,y=Height,xmin=BrLower,xmax=BrUpper),width=0)+
  geom_point(aes(x=Browse,y=Height,fill=Type),color="black",size=2,shape=21)+
  geom_smooth(aes(x=Browse,y=Height,group=Type),method="lm",se=F,color="black",size=0.3)+
  scale_fill_manual(name="",values=c("#CC2529","#396AB1"),labels=c("Five tallest stems","Random stems")
  scale_y_continuous(breaks=seq(0,400,100),labels=seq(0,400,100),
                     limits = c(0,405),expand = c(0, 0))+
  scale_x_continuous(breaks=seq(0.3,0.9,0.1),labels=scales::number_format(accuracy = 0.1),
                     limits = c(0.24,0.92),expand=c(0,0))+
  xlab("Browse probability")+
  ylab("Stem height (cm)")+
  ggtitle("(c)")+
```

```r
  guides(fill=guide_legend(reverse = TRUE,label.position = "left"))+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        panel.background = element_blank(), axis.line = element_line(colour = "black"),
        legend.key=element_blank(),legend.position = c(1,1.05),
        legend.justification = c("right", "top"),
        legend.key.size = unit(0.3, "cm"),legend.title = element_blank(),
        legend.text = element_text(size=10),
        plot.title = element_text(size=11,face="bold"),
        axis.title = element_text(size=11),
        axis.title.x=element_text(margin = margin(t = 5, r = 0, b = 0, l = 0)),
        axis.title.y=element_text(margin = margin(t = 0, r = 5, b = 0, l = 0)),
        axis.text.x = element_text(size=10, color="black"),
        axis.text.y=element_text(color="black",size=10),
        plot.margin=unit(c(c(0, 0.5, 0.2, 0.3)), units="line"))


# Print Plot ------------------------------------------------------------------
grid.arrange(fig3.list$Plots$a,fig3.list$Plots$b,fig3.list$Plots$c,ncol=1)

# Can also save plot with:
# tiff("Fig3.tiff",units = "in",width=3.2, height=8.5, res = 150)
# grid.arrange(fig3.list$Plots$a,fig3.list$Plots$b,fig3.list$Plots$c,ncol=1)
# dev.off()
```

Figure 4: Preferred Browse Height/Browse Escape Height

We modeled the effect of stem height on browsing to estimate PBH and BEH. We estimated separate GLMMs for 5T stems (N = 4,265) and random stems (N = 14,358), and included crossed random intercepts for stand identity and year. We used piecewise linear splines to identify the PBH, which we defined as the height threshold beyond which browsing probability decreased with further height increase. We constructed variables containing a linear spline for stem height so that the estimated coefficients measure the slopes of the segments before and after the threshold. We compared models with a single height threshold placed from 10-200 cm (first at 10 cm and then 1 cm intervals), a model with no height threshold, and an intercept-only model.

We evaluated competing models using marginal likelihoods and information-theoretic statistics (AICc), and used AMEs of the best models to estimate how browsing probability changed with increasing stem height. We also used the best models to identify the BEH.

The code comparing each height threshold takes several minutes for each sampling method. However, there is also code at the end of each model comparison section that just runs the best model for each sampling method.

The code is broken into the following steps:

- Subset the data into just 5T stems and just random stems
- Functions to create two height variables for the height spline at a user-specified knot
- Code for the 5T:
  - Intercept-only model
  - Model with no height threshold
  - Loop to run models with height thresholds from 10-cm to 200-cm and compare their AICc values
  - Run the best model (knot at 132 cm)
- Code for the random stems:
  - Intercept-only model
  - Model with no height threshold
  - Loop to run models with height thresholds from 10-cm to 200-cm and compare their AICc values

    - Run the best model (knot at 122 cm)
- Calculate average marginal effects for each of the best models
- Generate fitted values for the best models

```
#---------------------- *  *  *  *  * PBH *  *  *  *  * ----------------
# Prep -------------------------------------------------------------------
pbh.list <- list()

pbh.list$DF.5T <- aspen.df %>%
  filter(Type == "5T")

pbh.list$DF.rand <- aspen.df %>%
  filter(Type == "Random")

# Breakpoint Functions
b1 <- function(x, bp) ifelse(x <= bp, x, bp)
b2 <- function(x, bp) ifelse(x < bp, 0, x - bp)


#----------------------* * 5T * * ----------------
## NOTE: Using glmerControl with Nelder-Mead to ensure convergence
# Linear Model -----------------------------------------------------------
pbh.list$Models <- list()

pbh.list$Models$InterceptOnly <- glmer(Browse ~ (1|Year.f) + (1|Plot),
                                   family = binomial,
                                   data = pbh.list$DF.5T,
                                   control = br.control.2)

pbh.list$Models$Linear.5T <- glmer(Browse ~ Height + (1|Year.f) + (1|Plot),
                                   family = binomial,
                                   data = pbh.list$DF.5T,
                                   control = br.control.2)

# Loop -------------------------------------------------------------------

height.bp <- sort(c(seq(10,200,10),c(111:119,121:129,131:139,141:149)))

pbh.list$Out.5T <- data.frame(BP = height.bp,
                              N = NA, Constant = 1,
                              RandInt = 2, RandSlope = NA, Covariate = NA,
                              K = NA,LL = NA, AICc = NA,
                              B1 = NA, B2 = NA)

# Takes ~ 2 minutes
for(i in 1:length(height.bp)){
  bp <- height.bp[i]
  pbh.list$DF.5T$Ht1 <- b1(pbh.list$DF.5T$Height,bp)
  pbh.list$DF.5T$Ht2 <- b2(pbh.list$DF.5T$Height,bp)

  mod <- glmer(Browse ~ Ht1 + Ht2 + (1|Year.f) + (1|Plot),
               family = binomial, data = pbh.list$DF.5T,
               control = br.control.2)
```

```r
  skip_to_next <- FALSE #To keep going after errors (if spline is dropped)
  tryCatch({
    pbh.list$Out.5T$K[i] <- AICc(mod,return.K = T)
    pbh.list$Out.5T$LL[i] <- logLik(mod)[1]
    pbh.list$Out.5T$AICc[i] <- AICc(mod)
    pbh.list$Out.5T$B1[i] <- summary(mod)$coefficients[2,1]
    pbh.list$Out.5T$B2[i] <- summary(mod)$coefficients[3,1]
    pbh.list$Out.5T$N[i] <- nobs(mod)
    pbh.list$Out.5T$Covariate[i] <- length(fixef(mod)) - 1
    pbh.list$Out.5T$RandSlope[i] <- pbh.list$Out.5T$K[i]- 1 - pbh.list$Out.5T$RandInt[i]- pbh.list$Out.5

    print(paste("Knot",i))
  }
  , error = function(e) { skip_to_next <<- TRUE})
  if(skip_to_next) { next }
}


# Check AICc
pbh.list$Out.5T %>%
  mutate(Delta = AICc - min(AICc)) %>%
  filter(Delta < 2)

## Best is 130 and then 140

## Add to the intercept-only and linear models to broad list
pbh.list$Out.5T <- rbind(data.frame(BP = c("Intercept","Linear"),
                                    N = c(nobs(pbh.list$Models$InterceptOnly),
                                          nobs(pbh.list$Models$Linear.5T)),
                                    Constant = 1,
                                    RandInt = 2, RandSlope = 0,
                                    Covariate = c(length(
                                      fixef(pbh.list$Models$InterceptOnly)) - 1,
                                                  length(fixef(
                                                    pbh.list$Models$Linear.5T)) -
                                        1),
                                    K = c(AICc(pbh.list$Models$InterceptOnly,
                                               return.K = T),
                                          AICc(pbh.list$Models$Linear.5T,
                                               return.K = T)),
                                    LL = c(logLik(pbh.list$Models$InterceptOnly)[1],
                                           logLik(pbh.list$Models$Linear.5T)[1]),
                                    AICc = c(AICc(pbh.list$Models$InterceptOnly),
                                             AICc(pbh.list$Models$Linear.5T)),
                                    B1 = NA, B2 = NA),
                         pbh.list$Out.5T) %>%
  mutate(Delta = AICc - min(AICc))


# Best Model ---------------------------------------------------------------
# Save the model with knot at 132
pbh.list$DF.5T$Ht1 <- b1(pbh.list$DF.5T$Height,132)
pbh.list$DF.5T$Ht2 <- b2(pbh.list$DF.5T$Height,132)
```

```r
pbh.list$Models$Best.5T <- glmer(Browse ~ Ht1 + Ht2 + (1|Year.f) + (1|Plot),
                                  family = binomial, data = pbh.list$DF.5T,
                                  control = br.control.2)
summary(pbh.list$Models$Best.5T)


#----------------------* * Random *  *----------------

# Linear Model ------------------------------------------------------------

pbh.list$Models$Intercept.rand <- glmer(Browse ~ (1|Year.f) + (1|Plot),
                                        family = binomial,
                                        data = pbh.list$DF.rand,
                                        control = br.control.2)

pbh.list$Models$Linear.rand <- glmer(Browse ~ Height + (1|Year.f) + (1|Plot),
                                     family = binomial,
                                     data = pbh.list$DF.rand,
                                     control = br.control.2)


#  Loop -------------------------------------------------------------------

height.bp <- sort(c(seq(10,200,10),c(111:119,121:129,131:139,141:149)))

pbh.list$Out.rand <- data.frame(BP = height.bp,
                               N = NA, Constant = 1,
                               RandInt = 2, RandSlope = NA, Covariate = NA,
                               K = NA,LL = NA, AICc = NA,
                               B1 = NA, B2 = NA)

# Takes ~ 5 min
for(i in 1:length(height.bp)){
  bp <- height.bp[i]
  pbh.list$DF.rand$Ht1 <- b1(pbh.list$DF.rand$Height,bp)
  pbh.list$DF.rand$Ht2 <- b2(pbh.list$DF.rand$Height,bp)

  mod <- glmer(Browse ~ Ht1 + Ht2 + (1|Year.f) + (1|Plot),
               family = binomial, data = pbh.list$DF.rand,
               control = br.control.2)

  skip_to_next <- FALSE #To keep going after errors (if spline is dropped)
  tryCatch({
    pbh.list$Out.rand$K[i] <- AICc(mod,return.K = T)
    pbh.list$Out.rand$LL[i] <- logLik(mod)[1]
    pbh.list$Out.rand$AICc[i] <- AICc(mod)
    pbh.list$Out.rand$B1[i] <- summary(mod)$coefficients[2,1]
    pbh.list$Out.rand$B2[i] <- summary(mod)$coefficients[3,1]
    pbh.list$Out.rand$N[i] <- nobs(mod)
    pbh.list$Out.rand$Covariate[i] <- length(fixef(mod)) - 1
    pbh.list$Out.rand$RandSlope[i] <- pbh.list$Out.rand$K[i]- 1 - pbh.list$Out.rand$RandInt[i]- pbh.lis

    print(paste("Knot",i))
```

```r
  }
  , error = function(e) { skip_to_next <<- TRUE})
  if(skip_to_next) { next }
}


# Check AICc
pbh.list$Out.rand %>%
  mutate(Delta = AICc - min(AICc)) %>%
  filter(Delta < 2)

## Add to the intercept/linear list
pbh.list$Out.rand <- rbind(data.frame(BP = c("Intercept","Linear"),
                                    N = c(nobs(pbh.list$Models$Intercept.rand),
                                          nobs(pbh.list$Models$Linear.rand)),
                                    Constant = 1,
                                    RandInt = 2, RandSlope = 0,
                                    Covariate = c(length(fixef(
                                      pbh.list$Models$Intercept.rand)) - 1,
                                                  length(fixef(
                                                    pbh.list$Models$Linear.rand)) -
                                        1),
                                    K = c(AICc(pbh.list$Models$Intercept.rand,
                                               return.K = T),
                                          AICc(pbh.list$Models$Linear.rand,
                                               return.K = T)),
                                    LL = c(logLik(
                                      pbh.list$Models$Intercept.rand)[1],
                                           logLik(pbh.list$Models$Linear.rand)[1]),
                                    AICc = c(AICc(pbh.list$Models$Intercept.rand),
                                             AICc(pbh.list$Models$Linear.rand)),
                                    B1 = NA, B2 = NA),
                         pbh.list$Out.rand) %>%
  mutate(Delta = AICc - min(AICc))



# Best Model --------------------------------------------------------------
# Save the model with knot at 122
pbh.list$DF.rand$Ht1 <- b1(pbh.list$DF.rand$Height,122)
pbh.list$DF.rand$Ht2 <- b2(pbh.list$DF.rand$Height,122)

pbh.list$Models$Best.rand <- glmer(Browse ~ Ht1 + Ht2 + (1|Year.f) + (1|Plot),
                                   family = binomial, data = pbh.list$DF.rand,
                                   control = br.control.2)
summary(pbh.list$Models$Best.rand)


#----------------------* * AMEs *  *----------------
# Average marginal effects
pbh.list$AME <- list()

### RANDOM
```

```r
pbh.list$AME$Random <- margins(model = pbh.list$Models$Best.rand)

summary(pbh.list$AME$Random)

# To get change for 10-cm, multiply by 10 (current change is for 1-cm)

# 95% CI for height 1 AME
summary(pbh.list$AME$Random)[1,2] - 1.96*summary(pbh.list$AME$Random)[1,3]
summary(pbh.list$AME$Random)[1,2] + 1.96*summary(pbh.list$AME$Random)[1,3]

# 95% CI for height 2 AME
summary(pbh.list$AME$Random)[2,2] - 1.96*summary(pbh.list$AME$Random)[2,3]
summary(pbh.list$AME$Random)[2,2] + 1.96*summary(pbh.list$AME$Random)[2,3]


### 5T
pbh.list$AME$Five <- margins(model = pbh.list$Models$Best.5T)
summary(pbh.list$AME$Five)

# 95% CI for height 1 AME
summary(pbh.list$AME$Five)[1,2] - 1.96*summary(pbh.list$AME$Five)[1,3]
summary(pbh.list$AME$Five)[1,2] + 1.96*summary(pbh.list$AME$Five)[1,3]

# 95% CI for height 2 AME
summary(pbh.list$AME$Five)[2,2] - 1.96*summary(pbh.list$AME$Five)[2,3]
summary(pbh.list$AME$Five)[2,2] + 1.96*summary(pbh.list$AME$Five)[2,3]


# Predictions -------------------------------------------------------------
fig4.list <- list()
fig4.list$DF <- list()

ht1 <- c(b1(seq(0,400,1),122),b1(seq(0,400,1),132))
ht2 <- c(b2(seq(0,400,1),122),b2(seq(0,400,1),132))

fig4.list$DF$a <- data.frame(Type = c(rep("Random",401),rep("5T",401)),
                             Height = seq(0,400,1),
                             Ht1 = ht1, Ht2 = ht2,
                             Year.f = 0, Plot = 0,
                             Browse = NA, Lower = NA, Upper = NA)

fig4.list$DF$a$Browse <- plogis(c(predict(pbh.list$Models$Best.rand,
                                        newdata = fig4.list$DF$a[1:401,],
                                        re.form = NA),
                                  predict(pbh.list$Models$Best.5T,
                                          newdata = fig4.list$DF$a[402:802,],
                                          re.form = NA)))

ci1 <- confint(pbh.list$Models$Best.rand,method = "Wald")
ci2 <- confint(pbh.list$Models$Best.5T,method = "Wald")

fig4.list$DF$a$Lower <- c(plogis(ci1[3,1] + ci1[4,1]*fig4.list$DF$a$Ht1[1:401]
                               + ci1[5,1]*fig4.list$DF$a$Ht2[1:401]),
```

```r
                          plogis(ci2[3,1] + ci2[4,1]*fig4.list$DF$a$Ht1[402:802]
                                 + ci2[5,1]*fig4.list$DF$a$Ht2[402:802]))

fig4.list$DF$a$Upper <- c(plogis(ci1[3,2] + ci1[4,2]*fig4.list$DF$a$Ht1[1:401]
                                 + ci1[5,2]*fig4.list$DF$a$Ht2[1:401]),
                          plogis(ci2[3,2] + ci2[4,2]*fig4.list$DF$a$Ht1[402:802]
                                 + ci2[5,2]*fig4.list$DF$a$Ht2[402:802]))
```

We also plotted the % of stems taller than 132 cm (5T) or 122 cm (random) each year. The code below calculates these values, and can be used to reproduce figure 4.

```r
#---------------------- *  *  *  *  * FIGURE 4 *  *  *  *  * ----------------
## See end of PBH analysis for data formulation for panel a

fig4.list$DF$b <- aspen.df %>%
  mutate(Recruit = ifelse(Height > 132 & Type == "5T",1,
                          ifelse(Height <= 132 & Type == "5T",0,
                                 ifelse(Height > 122 & Type == "Random",1,0)))) %>%
  group_by(Year,Type) %>%
  summarize(N = n(),
            Recruit = sum(Recruit),
            Percent = (Recruit/N)*100)

fig4.list$Plots <- list()

# Panel A: Browse v Height -------------------------------------------------
fig4.list$Plots$a <- ggplot(data=fig4.list$DF$a)+
  geom_ribbon(aes(x=Height,ymin=Lower,ymax=Upper,fill=Type),alpha=0.4,show.legend = F)+
  geom_line(aes(x=Height,y=Browse,color=Type),size=.8)+
  scale_y_continuous(breaks=seq(0,1,0.2),labels=scales::number_format(accuracy = 0.1),
                     limits = c(0,1.0),expand = c(0, 0))+
  scale_fill_manual(name="",values=c("#CC2529","#396AB1"),labels=c("Five tallest stems","Random stems"))
  scale_color_manual(name="",values=c("#CC2529","#396AB1"),labels=c("Five tallest stems","Random stems"
  xlab("Stem height (cm)")+
  ylab("Browse probability")+
  ggtitle("(a)")+
  scale_x_continuous(breaks=seq(0,400,50),labels=seq(0,400,50),limits=c(0,420),
                     expand=c(0,0))+
  guides(color=guide_legend(reverse = TRUE,label.position = "left"))+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        panel.background = element_blank(), axis.line = element_line(colour = "black"),
        legend.key=element_blank(),legend.position = c(1,1.05),
        legend.justification = c("right", "top"),
        legend.key.size = unit(0.3, "cm"),legend.title = element_blank(),
        legend.text = element_text(size=9),
        plot.title = element_text(size=11,face="bold"),
        axis.title = element_text(size=11),
        axis.title.x=element_text(margin = margin(t = 5, r = 0, b = 0, l = 0)),
        axis.title.y=element_text(margin = margin(t = 0, r = 5, b = 0, l = 0)),
        axis.text.x = element_text(size=10,color="black"),
        axis.text.y=element_text(color="black",size=10),
        plot.margin=unit(c(c(0.5, 0.5, 0.5, 0.5)), units="line"))
```

```
# Panel B: Threshold ----------------------------------------------------------
fig4.list$Plots$b <- ggplot(data=fig4.list$DF$b)+
  geom_bar(position=position_dodge(-.9),stat="identity",aes(x=as.factor(Year),y=Percent,fill=Type),colo
           "black",alpha=0.7)+
  scale_fill_manual(name="",values=c("#CC2529","#396AB1"),labels=c("Five tallest stems","Random stems")
  xlab("Year")+
  ylab("% stems > preferred browse height")+
  ggtitle("(b)")+
  scale_y_continuous(breaks=seq(0,80,20),labels=seq(0,80,20),limits = c(0,85),
                     expand = c(0, 0))+
  scale_x_discrete(labels=c("'07","'08","'09","'10","'11","'12","'13","'14","'16","'17"))+
  guides(fill=guide_legend(reverse = TRUE))+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        panel.background = element_blank(), axis.line = element_line(colour = "black"),
        legend.key=element_blank(),legend.position = c(0.01,1.05),
        legend.justification = c("left", "top"),
        legend.key.size = unit(0.3, "cm"),legend.title = element_blank(),
        legend.text = element_text(size=9),
        plot.title = element_text(size=11,face="bold"),
        axis.title = element_text(size=11),
        axis.title.x=element_text(margin = margin(t = 5, r = 0, b = 0, l = 0)),
        axis.title.y=element_text(margin = margin(t = 0, r = 5, b = 0, l = 0)),
        axis.text.x = element_text(size=10, color="black"),
        axis.text.y=element_text(color="black",size=10),
        plot.margin=unit(c(c(0.5, 0.5, 0.5, 0.5)), units="line"))



# Print Plot ------------------------------------------------------------------
grid.arrange(fig4.list$Plots$a,fig4.list$Plots$b,ncol=2)

# Can save plot with:
# tiff("Fig4.tiff",units = "in",width=6.8, height=3, res = 150)
# grid.arrange(fig4.list$Plots$a,fig4.list$Plots$b,ncol=2)
# dev.off()
```

Figure 5: Recruitment

Finally, we used GLMMs to test how the effect of year on recruitment of stems differed by sampling method. We treated the stem as the unit of analysis and used GLMMs with a Bernoulli distribution and a logit link to analyze the probability a stem recruited (1 = recruited; 0 = not recruited). A stem 'recruited' if it exceeded the commonly assumed BEH of 200 cm or our estimated BEH of 300 cm (see section on PBH/BEH). Year was an integer that ranged from 1 (2007) to 11 (2017) and sampling method was a dummy variable (1 = 5T sampling; 0 = random sampling). We included a random intercept for plot, but had too few observations to also fit a random coefficient for year. Models were fit with unstructured covariance and the Laplace approximation. We used likelihood ratio tests to compare models with and without an interaction between year and sampling type.

We estimated average marginal effects (AMEs) from GLMMs using the `margins` package to quantify and compare annual changes of 5T and random stems. We used the `emmeans` package to calculate z-scores to test how the AMEs of year on recruitment differed between 5T and random stems.

We calculated population-averaged fitted values from best-fit GLMMs by deriving marginal expectations of the responses averaged over the random effects but conditional on the observed variables (again using the `margins` package). We refit year as a categorical factor and plotted the associated fitted values to illustrate the distribution of the underlying data after controlling for stand-level heterogeneity. Models with categorical year included a random intercept for plot and did not include a random slope for year.

Following previous studies, we also estimated recruitment at the stand level as the percentage of sampled stands with stems taller than the presumed reach of elk. We calculated this separately for 5T and random stems as the annual percentage of sampled stands in which the median stem height exceeded 200 cm (Fig. 5c) or 300 cm (Fig. 5d). Consistent with previous studies, recruitment estimates from 5T stems excluded stands that produced no young aspen.

The code below does the following:

- Creates 2 recruitment columns in the data: one for stem heights > 200 cm, and one for stem heights > 300 cm
- For recruitment measured as >200 cm:
  - Models recruitment as a function of the interaction between sampling method and continuous year
  - Models recruitment as a function of sampling method and continuous year without an interaction
  - Conducts a likelihood ratio test between the two models
  - Models recruitment as a function of the interaction between sampling method and categorical year
  - Calculates the average marginal effects (AMEs) of annual changes in recruitment probability depending on sampling method
  - Conducts a z-test to compare AMEs of year on recruitment probability based on sampling method
  - Calculates the marginal predictions of recruitment probability for each year and sampling method for both models (year as a continuous vs. categorical predictor)
- For recruitment measured as >300 cm:
  - Repeats steps taken for recruitment measured as > 200 cm

```
#---------------------- *  *  *  *  * RECRUIT 200 *  *  *  *  * ----------------
# Data -----------------------------------------------------------------
# Make recruitment variable
aspen.df <- aspen.df %>%
  mutate(Recruit.200 = ifelse(Height > 200, 1, 0),
         Recruit.300 = ifelse(Height > 300, 1, 0))

mod.list$R.200 <- list()

# Continuous Year ------------------------------------------------------
mod.list$R.200$Con <- glmer(Recruit.200 ~ Type2*Year.c +
                              (1|Plot),
                            data=aspen.df, family = binomial,
                            control = br.control)
summary(mod.list$R.200$Con)

## 95% Confidence interval for interaction term (B +/- 1.96*SE)
summary(mod.list$R.200$Con)$coef[4,1] - 1.96*summary(mod.list$R.200$Con)$coef[4,2]
summary(mod.list$R.200$Con)$coef[4,1] + 1.96*summary(mod.list$R.200$Con)$coef[4,2]

## No interaction
mod.list$R.200$Con.2 <- glmer(Recruit.200 ~ Type2+Year.c +
                                (1|Plot),
                              data=aspen.df, family = binomial,
                              control = br.control)

# Likelihood ratio test
anova(mod.list$R.200$Con.2,mod.list$R.200$Con)
```

```r
# Categorical Year ---------------------------------------------------------
# Model takes ~ 1 minute to run
mod.list$R.200$Cat <- glmer(Recruit.200 ~ Type2*Year.f + (1|Plot),
                            data=aspen.df, family = binomial,
                            control = br.control)
summary(mod.list$R.200$Cat)

# Average Marginal Effects -------------------------------------------------
ame.list$R.200 <- list()

summary(margins(model = mod.list$R.200$Con,
                data =new.df.all[new.df.all$Type2==1,],variables = "Year.c"))

summary(margins(model = mod.list$R.200$Con,
                data =new.df.all[new.df.all$Type2==0,],variables = "Year.c"))


# Marginal Predictions -----------------------------------------------------
# Continuous

ame.list$R.200$Pred.con <- prediction(mod.list$R.200$Con, type = "response",
                                      data = new.df.all)

# Get average margin by type and year
ame.list$R.200$Pred.con <- ame.list$R.200$Pred.con %>%
  group_by(Type2, Year.c) %>%
  summarize(Mean = mean(fitted),
            N = n(),
            SD = sd(fitted),
            SE = SD/sqrt(N),
            Lower = Mean - 1.96*SE,
            Upper = Mean + 1.96*SE) %>%
  select(Year.c,Type2,Mean,Lower,Upper)


# Categorical
ame.list$R.200$Pred.cat <- prediction(mod.list$R.200$Cat, type = "response",
                                      data = new.df.all)

# Get average margin by type and year
ame.list$R.200$Pred.cat <- ame.list$R.200$Pred.cat %>%
  group_by(Type2, Year.f) %>%
  summarize(Mean = mean(fitted),
            N = n(),
            SD = sd(fitted),
            SE = SD/sqrt(N),
            Lower = Mean - 1.96*SE,
            Upper = Mean + 1.96*SE) %>%
  select(Year.f,Type2,Mean,Lower,Upper)

# Z-test to compare AMEs of year on recruitment based on type
emtrends(mod.list$R.200$Con, pairwise ~ Type2, var = "Year.c")
```

```r
#---------------------- *  *  *  *  * 300 CM *  *  *  *  * ---------------
mod.list$R.300 <- list()


# Continuous Year ---------------------------------------------------------
mod.list$R.300$Con <- glmer(Recruit.300 ~ Type2*Year.c +
                              (1|Plot),
                            data=aspen.df, family = binomial,
                            control = br.control)
summary(mod.list$R.300$Con)

## 95% Confidence interval for interaction term (B +/- 1.96*SE)
summary(mod.list$R.300$Con)$coef[4,1] - 1.96*summary(mod.list$R.300$Con)$coef[4,2]
summary(mod.list$R.300$Con)$coef[4,1] + 1.96*summary(mod.list$R.300$Con)$coef[4,2]


## No interaction
mod.list$R.300$Con.2 <- glmer(Recruit.300 ~ Type2+Year.c +
                                (1|Plot),
                              data=aspen.df, family = binomial,
                              control = br.control)

anova(mod.list$R.300$Con.2,mod.list$R.300$Con)


# Categorical Year --------------------------------------------------------
# Takes 3 minutes to run
mod.list$R.300$Cat <- glmer(Recruit.300 ~ Type2*Year.f + (1|Plot),
                            data=aspen.df, family = binomial,
                            control = br.control)
summary(mod.list$R.300$Cat)

# Average Marginal Effects ------------------------------------------------
ame.list$R.300 <- list()

summary(margins(model = mod.list$R.300$Con,
                data =new.df.all[new.df.all$Type2==1,],variables = "Year.c"))

summary(margins(model = mod.list$R.300$Con,
                data =new.df.all[new.df.all$Type2==0,],variables = "Year.c"))


# Marginal Predictions ----------------------------------------------------
# Continuous

ame.list$R.300$Pred.con <- prediction(mod.list$R.300$Con, type = "response",
                                      data = new.df.all)

# Get average margin by type and year
ame.list$R.300$Pred.con <- ame.list$R.300$Pred.con %>%
  group_by(Type2, Year.c) %>%
  summarize(Mean = mean(fitted),
            N = n(),
```

```
            SD = sd(fitted),
            SE = SD/sqrt(N),
            Lower = Mean - 1.96*SE,
            Upper = Mean + 1.96*SE) %>%
  select(Year.c,Type2,Mean,Lower,Upper)


# Categorical
ame.list$R.300$Pred.cat <- prediction(mod.list$R.300$Cat, type = "response",
                                      data = new.df.all)

# Get average margin by type and year
ame.list$R.300$Pred.cat <- ame.list$R.300$Pred.cat %>%
  group_by(Type2, Year.f) %>%
  summarize(Mean = mean(fitted),
            N = n(),
            SD = sd(fitted),
            SE = SD/sqrt(N),
            Lower = Mean - 1.96*SE,
            Upper = Mean + 1.96*SE) %>%
  select(Year.f,Type2,Mean,Lower,Upper)

# Z-test to compare AMEs of year on recruitment based on type
emtrends(mod.list$R.300$Con, pairwise ~ Type2, var = "Year.c")
```

The following code manipulates the data to create Figure 5.

```
#---------------------- *   *   *   *   * FIGURE 5 *   *   *   *   * ----------------
# Data --------------------------------------------------------------------

fig5.list <- list()
fig5.list$DF <- list()

fig5.list$DF$a <- data.frame(Year = ame.list$R.200$Pred.con$Year.c+2006,
                             Type = ame.list$R.200$Pred.con$Type2,
                             Cont = ame.list$R.200$Pred.con$Mean,
                             Cont.Low = ame.list$R.200$Pred.con$Lower,
                             Cont.High = ame.list$R.200$Pred.con$Upper,
                             Cat = c(ame.list$R.200$Pred.cat$Mean[1:8],
                                            NA,
                                            ame.list$R.200$Pred.cat$Mean[9:10],
                                            ame.list$R.200$Pred.cat$Mean[12:19],
                                            NA,
                                            ame.list$R.200$Pred.cat$Mean[20:21]),
                             Cat.Low = c(ame.list$R.200$Pred.cat$Lower[1:8],
                                           NA,
                                           ame.list$R.200$Pred.cat$Lower[9:10],
                                           ame.list$R.200$Pred.cat$Lower[12:19],
                                           NA,
                                           ame.list$R.200$Pred.cat$Lower[20:21]),
                             Cat.High = c(ame.list$R.200$Pred.cat$Upper[1:8],
                                           NA,
                                           ame.list$R.200$Pred.cat$Upper[9:10],
                                           ame.list$R.200$Pred.cat$Upper[12:19],
```

```
                                          NA,
                                          ame.list$R.200$Pred.cat$Upper[20:21])) %>%
  mutate(Type = ifelse(Type == 0, "Random","5T")) %>%
  mutate(Cat.Low = ifelse(Cat.Low <0,0,Cat.Low),
         Cont.Low = ifelse(Cont.Low <0,0,Cont.Low))


fig5.list$DF$b <- data.frame(Year = ame.list$R.300$Pred.con$Year.c+2006,
                             Type = ame.list$R.300$Pred.con$Type2,
                             Cont = ame.list$R.300$Pred.con$Mean,
                             Cont.Low = ame.list$R.300$Pred.con$Lower,
                             Cont.High = ame.list$R.300$Pred.con$Upper,
                             Cat = c(ame.list$R.300$Pred.cat$Mean[1:8],
                                     NA,
                                     ame.list$R.300$Pred.cat$Mean[9:10],
                                     ame.list$R.300$Pred.cat$Mean[12:19],
                                     NA,
                                     ame.list$R.300$Pred.cat$Mean[20:21]),
                             Cat.Low = c(ame.list$R.300$Pred.cat$Lower[1:8],
                                          NA,
                                          ame.list$R.300$Pred.cat$Lower[9:10],
                                          ame.list$R.300$Pred.cat$Lower[12:19],
                                          NA,
                                          ame.list$R.300$Pred.cat$Lower[20:21]),
                             Cat.High = c(ame.list$R.300$Pred.cat$Upper[1:8],
                                          NA,
                                          ame.list$R.300$Pred.cat$Upper[9:10],
                                          ame.list$R.300$Pred.cat$Upper[12:19],
                                          NA,
                                          ame.list$R.300$Pred.cat$Upper[20:21])) %>%
  mutate(Type = ifelse(Type == 0, "Random","5T"))%>%
  mutate(Cat.Low = ifelse(Cat.Low <0,0,Cat.Low),
         Cont.Low = ifelse(Cont.Low <0,0,Cont.Low))


# Use data that includes transects with no suckers:
fig5.list$DF$c <- aspen.df.all %>%
  group_by(Type,Plot,Year) %>%
  mutate(Height = ifelse(Tree ==0,0,Height))%>%
  summarize(N = n(),
            Med.Ht = median(Height)) %>%
  mutate(Percent = ifelse(Med.Ht > 200, 1, 0)) %>%
  group_by(Type,Year) %>%
  summarize(N = n(),
            Percent = (sum(Percent)/N)*100) %>%
  mutate(Type = factor(Type, levels = c("Random","5T")))

fig5.list$DF$d <- aspen.df.all %>%
  group_by(Type,Plot,Year) %>%
  mutate(Height = ifelse(Tree ==0,0,Height))%>%
  summarize(N = n(),
            Med.Ht = median(Height)) %>%
  mutate(Percent = ifelse(Med.Ht > 300, 1, 0)) %>%
```

```r
  group_by(Type,Year) %>%
  summarize(N = n(),
            Percent = (sum(Percent)/N)*100) %>%
  mutate(Type = factor(Type, levels = c("Random","5T")))

fig5.list$Plots <- list()

# Panel a ------------------------------------------------------------------
fig5.list$Plots$a <- ggplot()+
  geom_ribbon(data=fig5.list$DF$a,
              aes(x=Year,ymin=Cont.Low, ymax = Cont.High,
                  group = Type, fill=Type),alpha = 0.3,show.legend = F)+
  geom_line(data=fig5.list$DF$a[which(fig5.list$DF$a$Year<2015),],
            aes(x=Year,y=Cont,group=Type,color=Type),size=.8)+
  geom_line(data=fig5.list$DF$a[which(fig5.list$DF$a$Year>2013 &fig5.list$DF$a$Year<2017),],
            aes(x=Year,y=Cont,group=Type,color=Type),lty="dashed",size=.8,
            show.legend = F)+
  geom_line(data=fig5.list$DF$a[which(fig5.list$DF$a$Year>2015),],
            aes(x=Year,y=Cont,group=Type,color=Type),size=.8)+
  geom_errorbar(data=fig5.list$DF$a,
                aes(x=Year,ymin=Cat.Low,ymax=Cat.High),width=0)+
  geom_point(data=fig5.list$DF$a,aes(x=Year,y=Cat,fill=Type),color="black",
             size=2,shape=21)+
  scale_fill_manual(name="",values=c("#CC2529","#396AB1"),labels=c("Five tallest stems","Random stems")
  scale_color_manual(name="",values=c("#CC2529","#396AB1"),labels=c("Five tallest stems","Random stems"
  scale_x_continuous(breaks=seq(2007,2017,2),labels=seq(2007,2017,2),
                     limits = c(2007,2017))+
  scale_y_continuous(breaks=seq(0,0.75,0.25),labels=scales::number_format(accuracy = 0.01),
                     limits = c(0,0.76),expand = c(0, 0))+
  xlab("")+
  ylab("Recruitment probability")+
  ggtitle("(a)")+
  guides(fill=guide_legend(reverse = TRUE),color=guide_legend(reverse = TRUE))+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        panel.background = element_blank(), axis.line = element_line(colour = "black"),
        legend.key=element_blank(),legend.text = element_text(size=8),
        legend.justification = c("left", "top"),legend.position = c(0.005,1.05),
        legend.key.size = unit(0.3, "cm"),legend.title = element_blank(),
        plot.title = element_text(size=11,face="bold"),
        axis.title = element_text(size=11),
        axis.title.x=element_text(margin = margin(t = 5, r = 0, b = 0, l = 0)),
        axis.title.y=element_text(margin = margin(t = 0, r = 5, b = 0, l = 0)),
        axis.text.x = element_text(size=10, color="black"),
        axis.text.y=element_text(color="black",size=10),
        plot.margin=unit(c(c(0.5, 0.5, 0, 0.5)), units="line"))


# Panel b ------------------------------------------------------------------
fig5.list$Plots$b <- ggplot()+
  geom_ribbon(data=fig5.list$DF$b,
              aes(x=Year,ymin=Cont.Low, ymax = Cont.High,
                  group = Type, fill=Type),alpha = 0.3,show.legend = F)+
  geom_line(data=fig5.list$DF$b[which(fig5.list$DF$b$Year<2015),],
```

```r
            aes(x=Year,y=Cont,group=Type,color=Type),size=.8)+
  geom_line(data=fig5.list$DF$b[which(fig5.list$DF$b$Year>2013 &fig5.list$DF$b$Year<2017),],
            aes(x=Year,y=Cont,group=Type,color=Type),lty="dashed",size=.8,
            show.legend = F)+
  geom_line(data=fig5.list$DF$b[which(fig5.list$DF$b$Year>2015),],
            aes(x=Year,y=Cont,group=Type,color=Type),size=.8)+
  geom_errorbar(data=fig5.list$DF$b,
                aes(x=Year,ymin=Cat.Low,ymax=Cat.High),width=0)+
  geom_point(data=fig5.list$DF$b,aes(x=Year,y=Cat,fill=Type),color="black",
             size=2,shape=21)+
  scale_fill_manual(name="",values=c("#CC2529","#396AB1"),labels=c("Five tallest stems","Random stems")
  scale_color_manual(name="",values=c("#CC2529","#396AB1"),labels=c("Five tallest stems","Random stems"

  scale_y_continuous(breaks=seq(0,0.75,0.25),labels=scales::number_format(accuracy = 0.01),
                     limits = c(0,0.76),expand = c(0, 0))+
  xlab("")+
  ylab("")+
  ggtitle("(b)")+
  guides(fill=guide_legend(reverse = TRUE),color=guide_legend(reverse = TRUE))+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        panel.background = element_blank(), axis.line = element_line(colour = "black"),
        legend.key=element_blank(),legend.text = element_text(size=8),
        legend.justification = c("left", "top"),legend.position = c(0.005,1.05),
        legend.key.size = unit(0.3, "cm"),legend.title = element_blank(),
        plot.title = element_text(size=11,face="bold"),
        axis.title = element_text(size=11),
        axis.title.x=element_text(margin = margin(t = 5, r = 0, b = 0, l = 0)),
        axis.title.y=element_text(margin = margin(t = 0, r = 5, b = 0, l = 0)),
        axis.text.x = element_text(size=10, color="black"),
        axis.text.y=element_text(color="black",size=10),
        plot.margin=unit(c(c(0.5, 0.5, 0, 0)), units="line"))


# Panel c -----------------------------------------------------------------
fig5.list$Plots$c <- ggplot(data = fig5.list$DF$c, aes(x=as.factor(Year), y=Percent,fill=Type))+
  geom_bar(stat="identity",position="dodge",color="black",alpha=0.7)+
  scale_fill_manual(name="",values=c("#396AB1","#CC2529"),
                    labels=c("Median height of random stems >200 cm",
                             "Median height of five tallest stems >200 cm"))+
  scale_y_continuous(breaks=seq(0,70,10),labels=seq(0,70,10),limits = c(0,75),
                     expand = c(0, 0))+
  scale_x_discrete(labels=c("'07","'08","'09","'10","'11","'12","'13","'14","'16","'17"))+
  xlab("Year")+
  ylab("% aspen stands")+
  ggtitle("(c)")+
  guides(fill=guide_legend(reverse = TRUE))+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        panel.background = element_blank(), axis.line = element_line(colour = "black"),
        legend.key=element_blank(),legend.text = element_text(size=8),
        legend.justification = c("left", "top"),legend.position = c(0.005,1.05),
        legend.key.size = unit(0.3, "cm"),legend.title = element_blank(),
        plot.title = element_text(size=11,face="bold"),
        axis.title = element_text(size=11),
```

```r
        axis.title.x=element_text(margin = margin(t = 5, r = 0, b = 0, l = 0)),
        axis.title.y=element_text(margin = margin(t = 0, r = 5, b = 0, l = 0)),
        axis.text.x = element_text(size=10, color="black"),
        axis.text.y=element_text(color="black",size=10),
        plot.margin=unit(c(c(0, 0.5, 0.5, 0.5)), units="line"))

# Panel d -----------------------------------------------------------------

fig5.list$Plots$d <- ggplot(data = fig5.list$DF$d, aes(x=as.factor(Year),
                                                      y=Percent,fill=Type))+
  geom_bar(stat="identity",position="dodge",color="black",alpha=0.7)+
  scale_fill_manual(name="",values=c("#396AB1","#CC2529"),
                    labels=c("Median height of random stems >300 cm",
                             "Median height of five tallest stems >300 cm"))+
  scale_y_continuous(breaks=seq(0,70,10),labels=seq(0,70,10),limits = c(0,75),
                     expand = c(0, 0))+
  scale_x_discrete(labels=c("'07","'08","'09","'10","'11","'12","'13","'14","'16","'17"))+
  xlab("Year")+
  ylab("")+
  ggtitle("(d)")+
  guides(fill=guide_legend(reverse = TRUE))+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        panel.background = element_blank(), axis.line = element_line(colour = "black"),
        legend.key=element_blank(),legend.text = element_text(size=8),
        legend.justification = c("left", "top"),legend.position = c(0.005,1.05),
        legend.key.size = unit(0.3, "cm"),legend.title = element_blank(),
        plot.title = element_text(size=11,face="bold"),
        axis.title = element_text(size=11),
        axis.title.x=element_text(margin = margin(t = 5, r = 0, b = 0, l = 0)),
        axis.title.y=element_text(margin = margin(t = 0, r = 5, b = 0, l = 0)),
        axis.text.x = element_text(size=10, color="black"),
        axis.text.y=element_text(color="black",size=10),
        plot.margin=unit(c(c(0, 0.5, 0.5, 0)), units="line"))


# Print Plot --------------------------------------------------------------------
grid.arrange(fig5.list$Plots$a,fig5.list$Plots$b,
             fig5.list$Plots$c,fig5.list$Plots$d,ncol=2)

# Can also save the figure with:
# tiff("Fig5.tiff",units = "in",width=6.8, height=5.5, res = 150)
# grid.arrange(fig5.list$Plots$a,fig5.list$Plots$b,
#              fig5.list$Plots$c,fig5.list$Plots$d,ncol=2)
# dev.off()
```