Growth_mort

Carmen

October 31, 2019

Next step: Consider adding CEIN to the initialization even though there isn't any in the American River footprint

This script runs simulations with the most up-to-date functions

Define parameters

```r
fire <- "AMRC"
year <- "2016"
years <- 20
iterations <- 1000
max_shrub_ht_cm <- 250
max_shrub_ht_years <- 15
n_seedlings <- 100
length_m <- 40
height_m <- 40
lambda <- 4
shrub_clumpiness <- 7
```

Load functions

Create shrub patch

```r
source("functions/shrubclump.R")
```

Initialization function

```r
source("functions/initialize.R")
```

Height growth functions

```r
source("functions/abcogrowth.R")
source("functions/pipogrowth.R")
```

Diameter growth functions

```r
source("functions/abcodia.R")
source("functions/pipodia.R")
```

Mortality functions

```r
source("functions/abcomort.R")
source("functions/pipomort.R")
```

Shrub growth functions

```
source("functions/abco_shrubgrowth.R")
source("functions/pipo_shrubgrowth.R")
```

Simulation function

```
source("functions/sim.R")
```

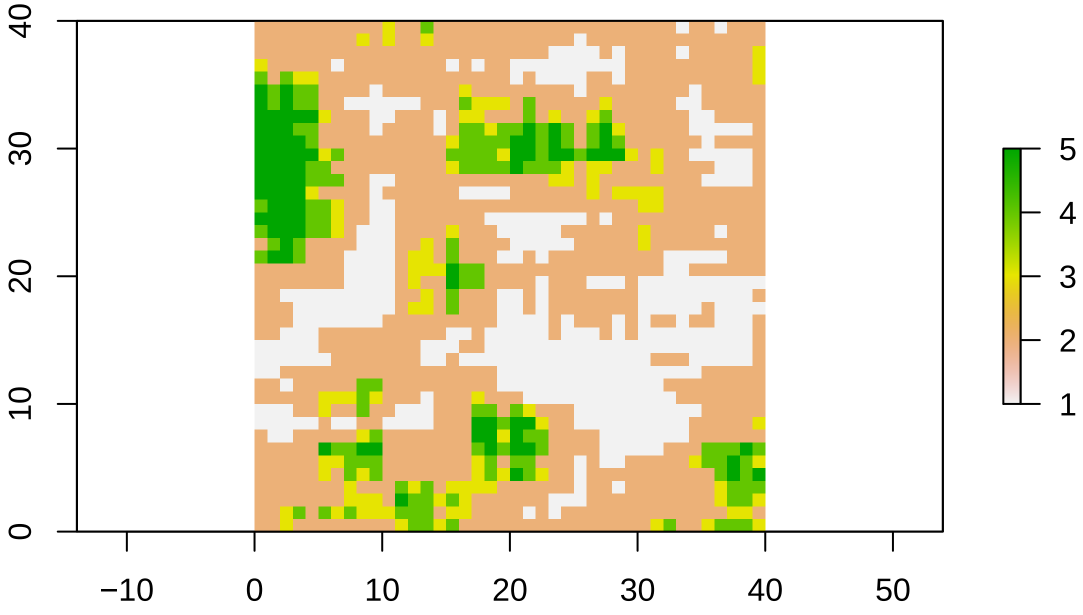
Iteration function

```
source("functions/iterate.R")
```

Initialize

Create clumped shrub pattern

```
shrubclump()
plot(r)
```



Randomly select seedlings from data and place them on the shrub patch

```
initialize()
```

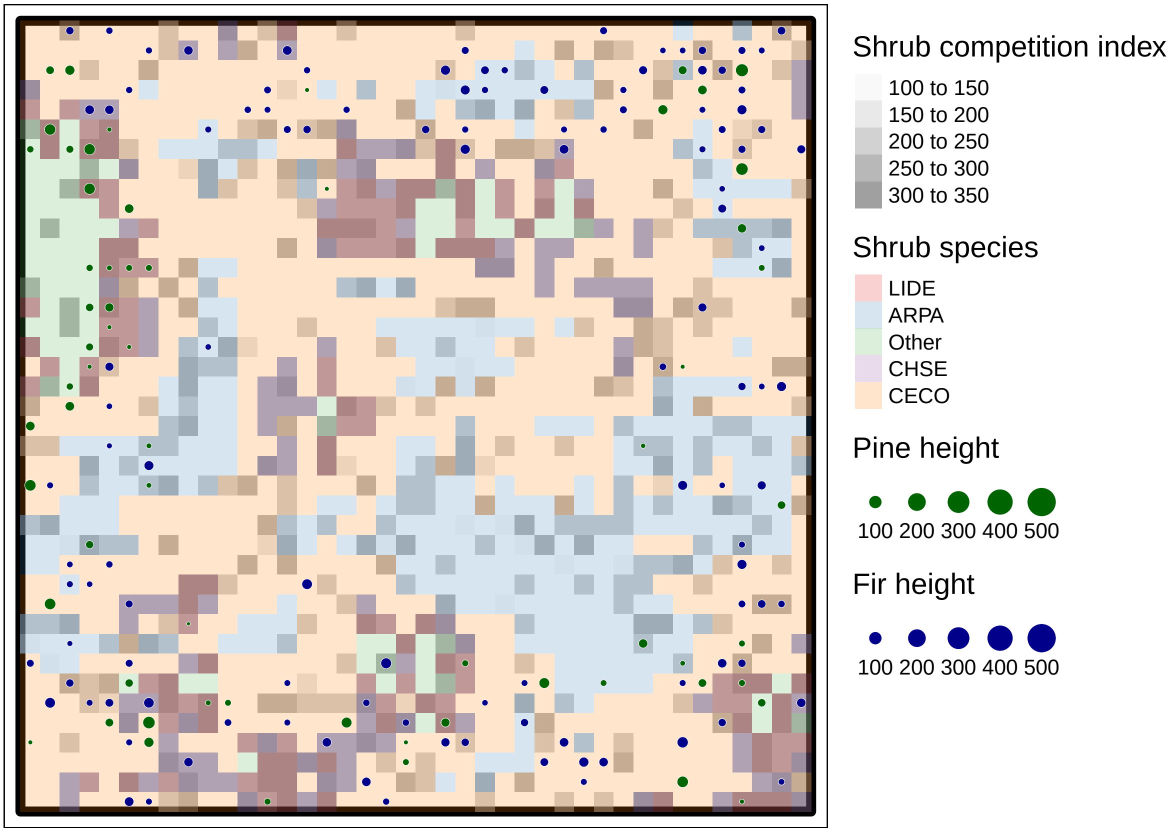
```
## Joining, by = "ID"
## Joining, by = "Sdlg"

## Warning: Column `Sdlg` joining factors with different levels, coercing to
## character vector
```

Plot patch with seedlings

```
max_shrub <- max(r@data@attributes[[1]]$sqrt_shrubarea3)
r@data@attributes[[1]]$shrub_rel <- r@data@attributes[[1]]$sqrt_shrubarea3/max_shrub
```

```
pts.sf.pipo.graph <- pts.sf.pipo %>%
  rename("Pine height" = Ht_cm1)
pts.sf.abco.graph <- pts.sf.abco %>%
  rename("Fir height" = Ht_cm1)
```

```
tm_shape(p)+
  tm_borders(col = "black", lwd= 5)+
tm_shape(r)+
  tm_raster(col = "sqrt_shrubarea3", title = "Shrub competition index", palette = "Greys", alpha = .5)+
tm_shape(r)+
  tm_raster(col = "ShrubSpp03", alpha = .2, title = "Shrub species", palette = "Set1")+
  tm_layout(asp=1:1, legend.outside = T)+
tm_shape(pts.sf.pipo.graph)+
  tm_symbols(size = "Pine height", col = "darkgreen", size.max = 500, border.col = "white", border.lwd =
tm_shape(pts.sf.abco.graph)+
  tm_symbols(size = "Fir height", col = "darkblue", size.max = 500, border.col = "white", border.lwd =
```



Simulate across years

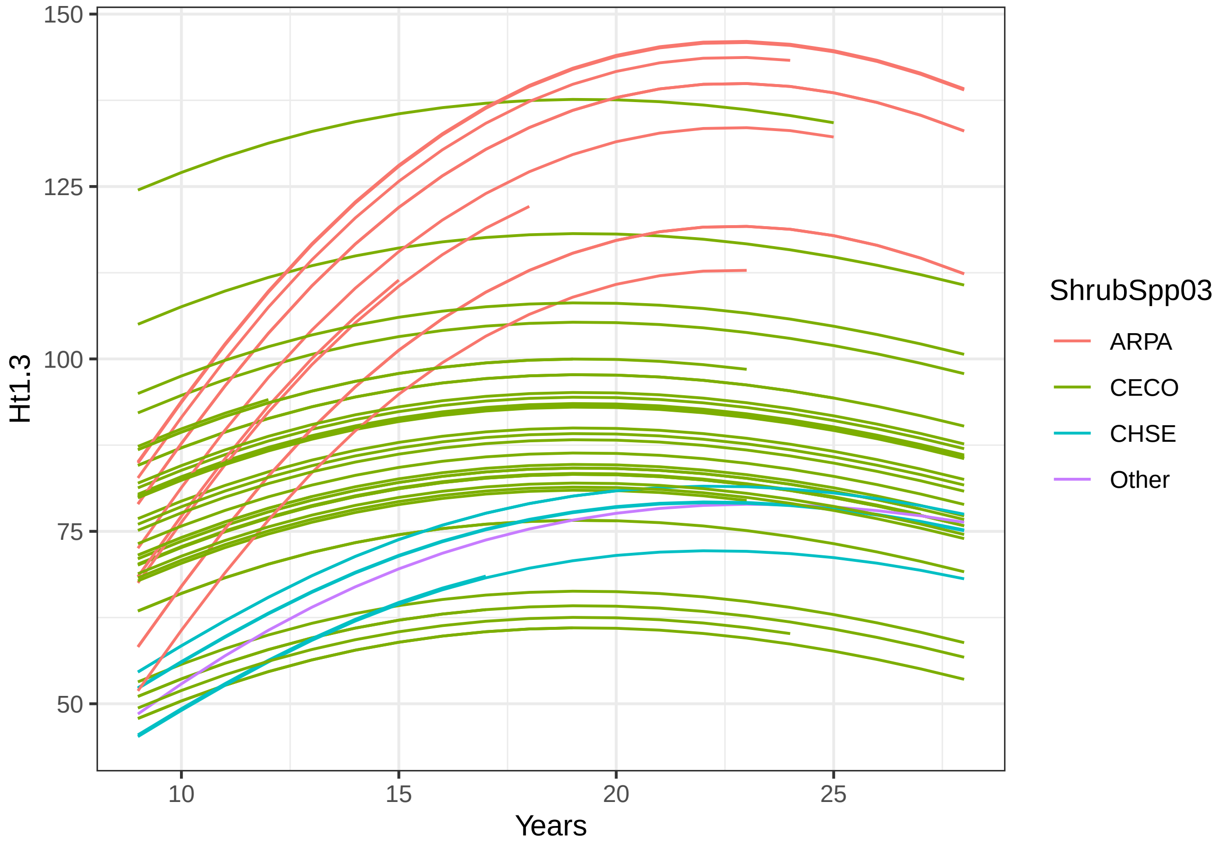
```
time <- Sys.time()
suppressMessages(sim(years))
print(paste("One simulation run took", Sys.time()-time, "seconds"))
```

```
## [1] "One simulation run took 5.85290002822876 seconds"
```

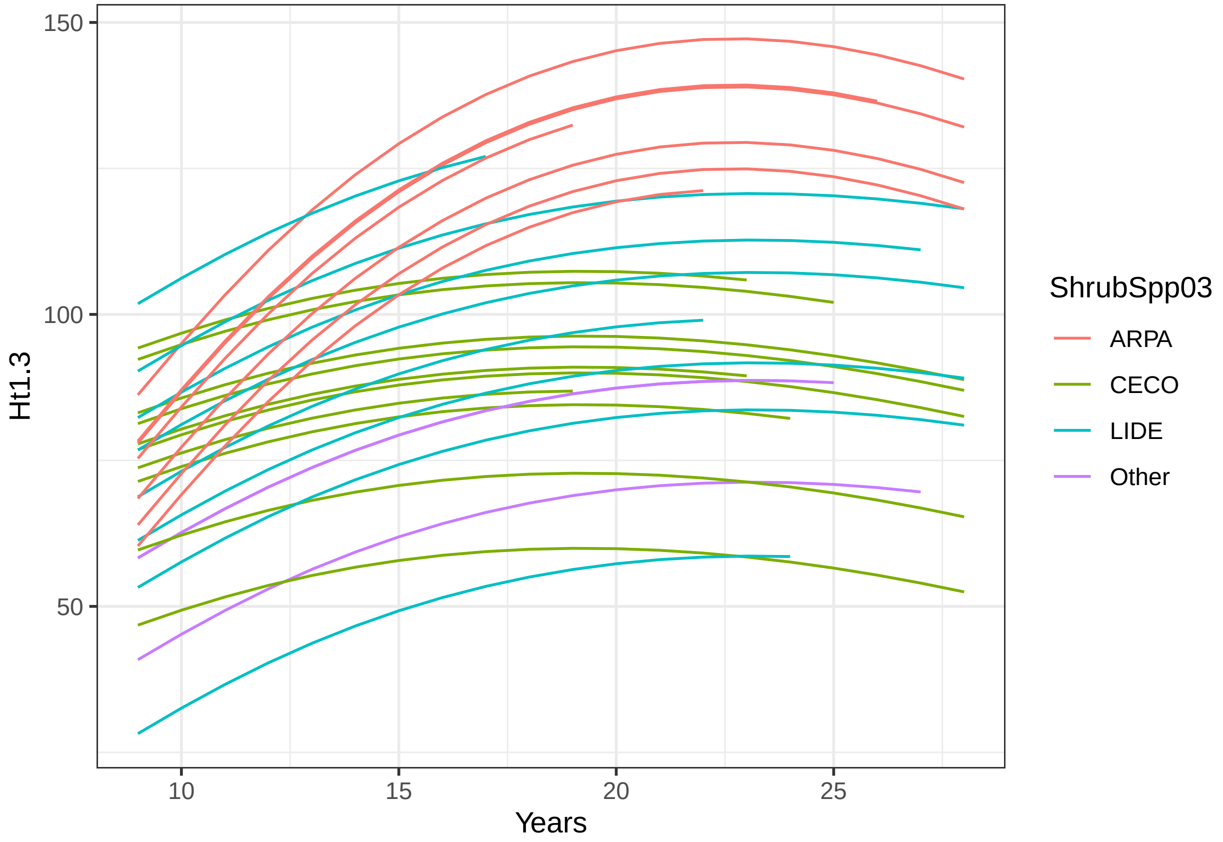
Tests

Shrub growth patterns

```
ggplot(dfsimall %>% filter(Species == "ABCO"))+
  geom_line(aes(x = Years, y = Ht1.3, col = ShrubSpp03, group = Sdlg))+
  theme_bw()
```



```
ggplot(dfsimall %>% filter(Species == "PIPO"))+
  geom_line(aes(x = Years, y = Ht1.3, col = ShrubSpp03, group = Sdlg))+
  theme_bw()
```

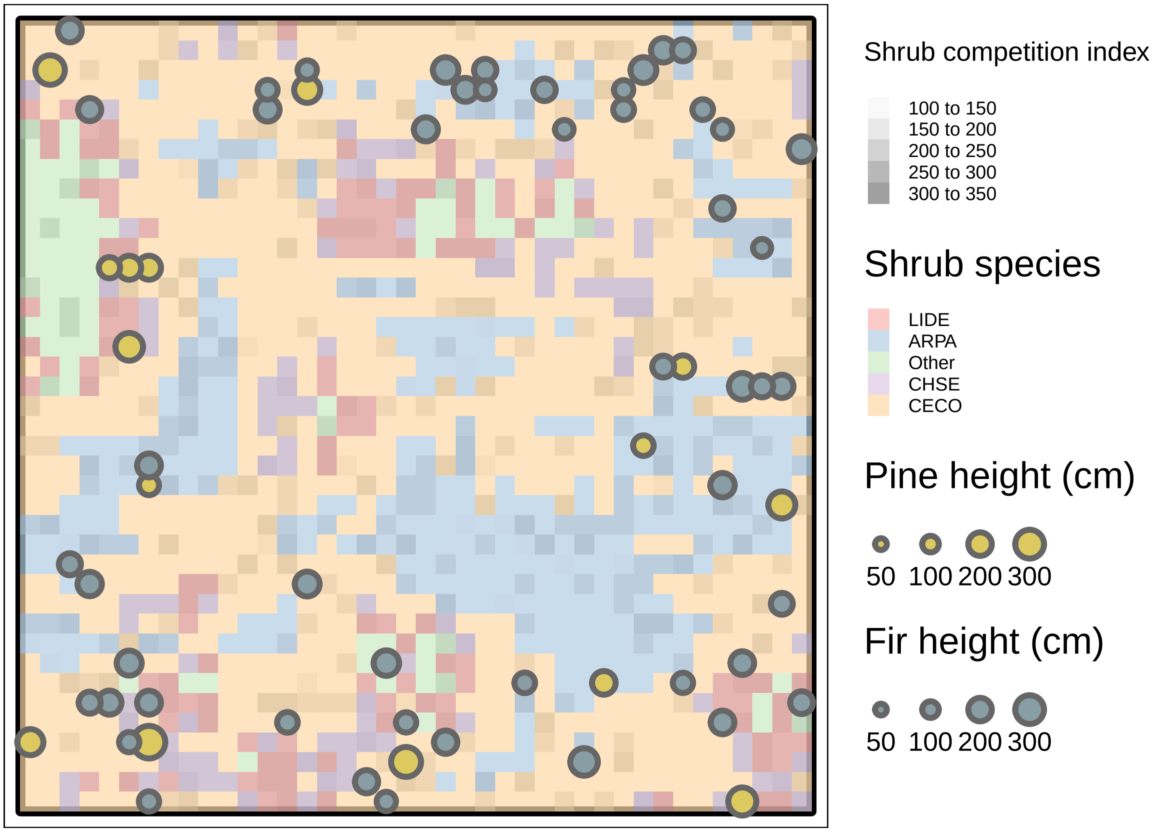


```
pts.sf.pipo.graph <- pts.sf.pipo %>%
  rename("Pine height (cm)" = Ht_cm1)
pts.sf.abco.graph <- pts.sf.abco %>%
  rename("Fir height (cm)" = Ht_cm1)
```

Plot

```
tm_shape(p)+
  tm_borders(col = "black", lwd= 5)+
tm_shape(r)+
  tm_raster(col = "sqrt_shrubarea3", title = "Shrub competition index", palette = "Greys", alpha = .5)+
tm_shape(r)+
  tm_raster(col = "ShrubSpp03", alpha = .7, title = "Shrub species", palette = "Pastel1")+
  tm_layout(asp=1:1, legend.outside = T, legend.title.size = 4, legend.text.size = 2)+
tm_shape(pts.sf.pipo.graph)+
  tm_symbols(size = "Pine height (cm)", col = "#DCC960", size.max = 300, border.lwd = 3)+
tm_shape(pts.sf.abco.graph)+
  tm_symbols(size = "Fir height (cm)", col = "#899DA4", size.max = 300, border.lwd = 3)
```

```
## Note that 3 values of the variable "Pine height (cm)" (the highest being 384.738575673223) are large
```



Iterate

```
time <- Sys.time()
iterate(iterations)
```

```
## [1] "Done with 10 iterations in 58.9 minutes"
## [1] "Done with 20 iterations in 1.8 minutes"
## [1] "Done with 30 iterations in 2.7 minutes"
## [1] "Done with 40 iterations in 3.5 minutes"
## [1] "Done with 50 iterations in 4.4 minutes"
## [1] "Done with 60 iterations in 5.5 minutes"
## [1] "Done with 70 iterations in 6.4 minutes"
## [1] "Done with 80 iterations in 7.2 minutes"
## [1] "Done with 90 iterations in 8.1 minutes"
## [1] "Done with 100 iterations in 9 minutes"
## [1] "Done with 110 iterations in 9.8 minutes"
## [1] "Done with 120 iterations in 10.7 minutes"
## [1] "Done with 130 iterations in 11.5 minutes"
## [1] "Done with 140 iterations in 12.4 minutes"
## [1] "Done with 150 iterations in 13.2 minutes"
## [1] "Done with 160 iterations in 14.1 minutes"
## [1] "Done with 170 iterations in 15 minutes"
## [1] "Done with 180 iterations in 15.8 minutes"
## [1] "Done with 190 iterations in 16.7 minutes"
## [1] "Done with 200 iterations in 17.6 minutes"
## [1] "Done with 210 iterations in 18.5 minutes"
## [1] "Done with 220 iterations in 19.3 minutes"
## [1] "Done with 230 iterations in 20.2 minutes"
## [1] "Done with 240 iterations in 21.1 minutes"
## [1] "Done with 250 iterations in 22 minutes"
## [1] "Done with 260 iterations in 22.8 minutes"
## [1] "Done with 270 iterations in 23.8 minutes"
## [1] "Done with 280 iterations in 24.6 minutes"
## [1] "Done with 290 iterations in 25.5 minutes"
## [1] "Done with 300 iterations in 26.4 minutes"
## [1] "Done with 310 iterations in 27.2 minutes"
## [1] "Done with 320 iterations in 28.1 minutes"
## [1] "Done with 330 iterations in 29 minutes"
## [1] "Done with 340 iterations in 29.9 minutes"
## [1] "Done with 350 iterations in 30.8 minutes"
## [1] "Done with 360 iterations in 31.7 minutes"
## [1] "Done with 370 iterations in 32.6 minutes"
## [1] "Done with 380 iterations in 33.5 minutes"
## [1] "Done with 390 iterations in 34.4 minutes"
## [1] "Done with 400 iterations in 35.3 minutes"
## [1] "Done with 410 iterations in 36.1 minutes"
## [1] "Done with 420 iterations in 37.1 minutes"
## [1] "Done with 430 iterations in 38.1 minutes"
## [1] "Done with 440 iterations in 39 minutes"
## [1] "Done with 450 iterations in 39.9 minutes"
## [1] "Done with 460 iterations in 40.8 minutes"
## [1] "Done with 470 iterations in 41.7 minutes"
## [1] "Done with 480 iterations in 42.6 minutes"
## [1] "Done with 490 iterations in 43.4 minutes"
## [1] "Done with 500 iterations in 44.4 minutes"
## [1] "Done with 510 iterations in 45.2 minutes"
## [1] "Done with 520 iterations in 46.1 minutes"
## [1] "Done with 530 iterations in 47 minutes"
## [1] "Done with 540 iterations in 47.9 minutes"
## [1] "Done with 550 iterations in 48.9 minutes"
## [1] "Done with 560 iterations in 49.8 minutes"
## [1] "Done with 570 iterations in 50.7 minutes"
## [1] "Done with 580 iterations in 51.7 minutes"
## [1] "Done with 590 iterations in 52.6 minutes"
## [1] "Done with 600 iterations in 53.5 minutes"
## [1] "Done with 610 iterations in 54.4 minutes"
## [1] "Done with 620 iterations in 55.4 minutes"
## [1] "Done with 630 iterations in 56.3 minutes"
## [1] "Done with 640 iterations in 57.2 minutes"
## [1] "Done with 650 iterations in 58.2 minutes"
## [1] "Done with 660 iterations in 59.1 minutes"
## [1] "Done with 670 iterations in 1 minutes"
## [1] "Done with 680 iterations in 1 minutes"
## [1] "Done with 690 iterations in 1 minutes"
## [1] "Done with 700 iterations in 1 minutes"
## [1] "Done with 710 iterations in 1.1 minutes"
## [1] "Done with 720 iterations in 1.1 minutes"
## [1] "Done with 730 iterations in 1.1 minutes"
## [1] "Done with 740 iterations in 1.1 minutes"
## [1] "Done with 750 iterations in 1.1 minutes"
## [1] "Done with 760 iterations in 1.1 minutes"
## [1] "Done with 770 iterations in 1.2 minutes"
## [1] "Done with 780 iterations in 1.2 minutes"
## [1] "Done with 790 iterations in 1.2 minutes"
## [1] "Done with 800 iterations in 1.2 minutes"
## [1] "Done with 810 iterations in 1.2 minutes"
## [1] "Done with 820 iterations in 1.2 minutes"
## [1] "Done with 830 iterations in 1.3 minutes"
## [1] "Done with 840 iterations in 1.3 minutes"
## [1] "Done with 850 iterations in 1.3 minutes"
## [1] "Done with 860 iterations in 1.3 minutes"
## [1] "Done with 870 iterations in 1.3 minutes"
## [1] "Done with 880 iterations in 1.3 minutes"
## [1] "Done with 890 iterations in 1.4 minutes"
## [1] "Done with 900 iterations in 1.4 minutes"
## [1] "Done with 910 iterations in 1.4 minutes"
## [1] "Done with 920 iterations in 1.4 minutes"
## [1] "Done with 930 iterations in 1.4 minutes"
## [1] "Done with 940 iterations in 1.4 minutes"
## [1] "Done with 950 iterations in 1.4 minutes"
## [1] "Done with 960 iterations in 1.5 minutes"
## [1] "Done with 970 iterations in 1.5 minutes"
## [1] "Done with 980 iterations in 1.5 minutes"
## [1] "Done with 990 iterations in 1.5 minutes"
## [1] "Done with 1000 iterations in 1.5 minutes"
```

```r
print(paste("That took", round(Sys.time()-time, 1), "minutes"))
```

```
## [1] "That took 1.5 minutes"
```

```r
dfsimallreps %>%
  group_by(rep) %>%
  summarize(mean(Ht_cm1))
```

```
## # A tibble: 1,000 x 2
##      rep `mean(Ht_cm1)`
##    <int>          <dbl>
##  1     1           120.
##  2     2           116.
##  3     3           113.
##  4     4           114.
##  5     5           115.
##  6     6           121.
##  7     7           116.
##  8     8           114.
##  9     9           116.
## 10    10           126.
## # ... with 990 more rows
```

Summarize

Height by year

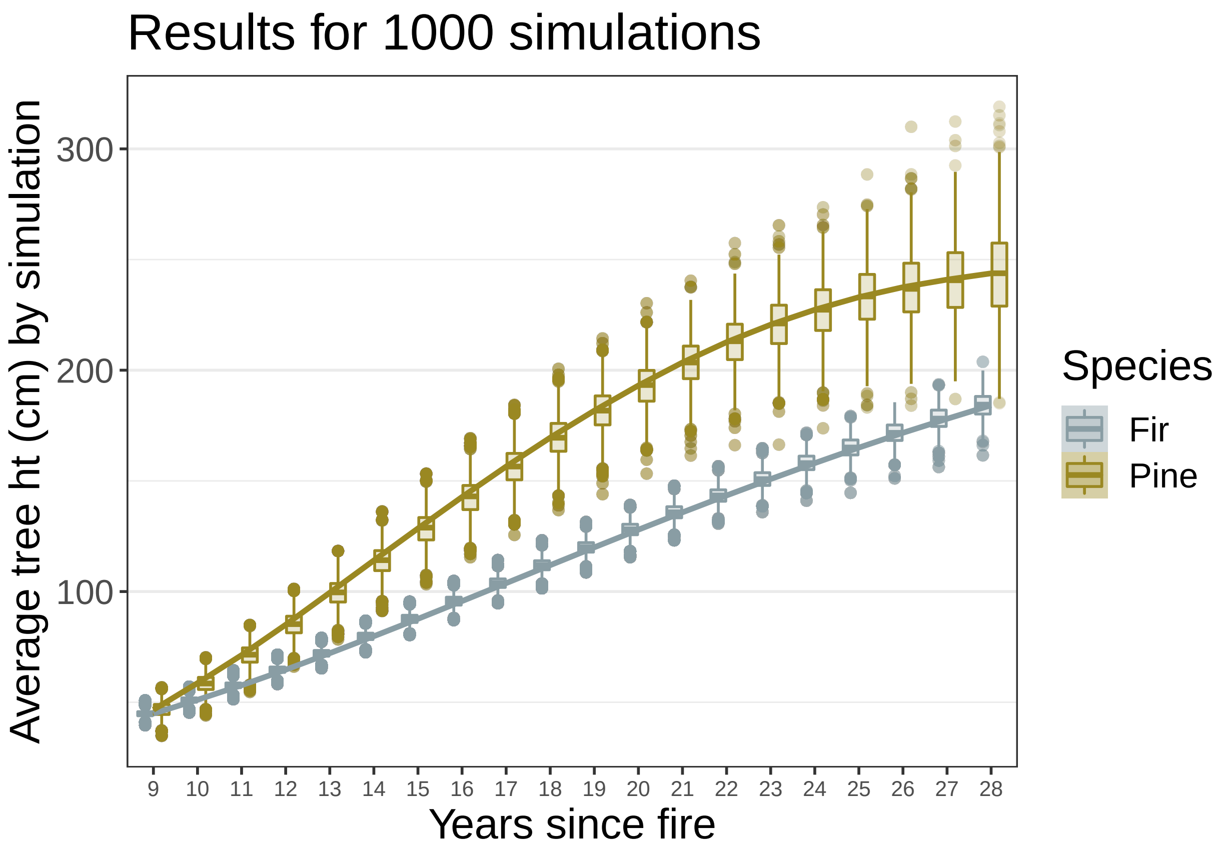
```
dfsimallreps_summary <- dfsimallreps %>%
  ungroup() %>%
  mutate(rep = as.factor(paste(rep))) %>%
  group_by(rep, Years, Species) %>%
  mutate(mean_ht_years = mean(Ht_cm1))
dfsimallreps_summary %>% dplyr::select(rep, Years, mean_ht_years) %>% summary()
```

```
## Adding missing grouping variables: `Species`
```

```
##   Species              rep              Years        mean_ht_years
##  ABCO:1770373   745     :   3068   Min.   : 9.00   Min.   : 34.77
##  PIPO:1034750   540     :   3050   1st Qu.:12.00   1st Qu.: 71.33
##                 648     :   3039   Median :16.00   Median :110.12
##                 564     :   3032   Mean   :16.74   Mean   :117.62
##                 738     :   3026   3rd Qu.:21.00   3rd Qu.:156.82
##                 9       :   3023   Max.   :28.00   Max.   :319.05
##                 (Other):2786885
```

```
ggplot(dfsimallreps_summary, aes(x = as.factor(Years), y = mean_ht_years, fill = Species, col = Species
  geom_boxplot(alpha = .2, outlier.alpha = .02)+
  geom_smooth(aes(x = as.factor(Years), y = mean_ht_years, group = Species, col = Species), size = 1)+
  ggtitle("Results for 1000 simulations")+
  xlab("Years since fire")+
  ylab("Average tree ht (cm) by simulation")+
  theme_bw()+
  scale_color_manual(values = c("#899DA4", "#9A8822"), labels = c("Fir", "Pine"))+
  scale_fill_manual(values = c("#899DA4", "#9A8822"), labels = c("Fir", "Pine"))+
  theme(text = element_text(size = 16),
        panel.grid.minor.x = element_blank(),
        panel.grid.major.x = element_blank(),
        axis.text.x = element_text(size = 8))
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



```
ggsave(file = "../../results/figures/sim_1000_hts.png", width = 6, height =4, dpi = 400)
```

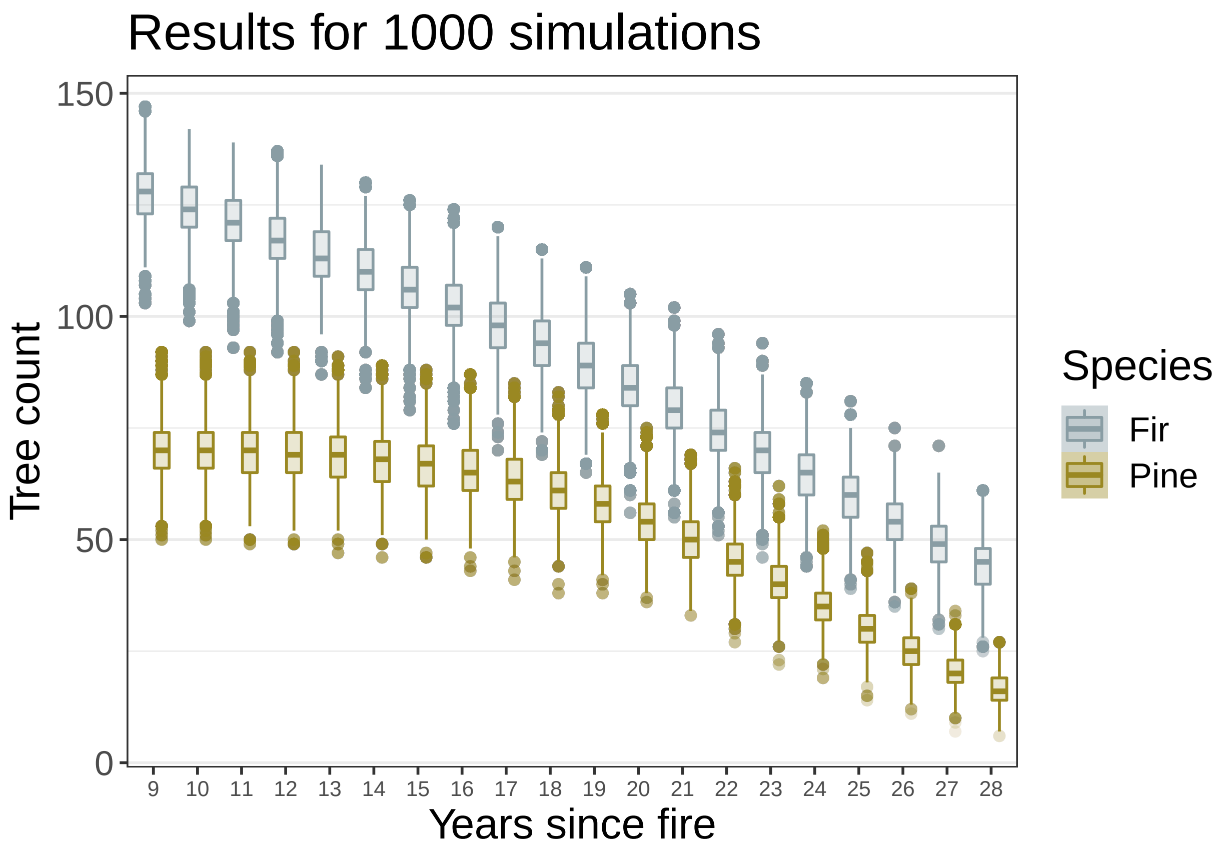
```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

Counts by year

```
dfsimallreps_summary <- dfsimallreps %>%
  ungroup() %>%
  group_by(rep, Years, Species) %>%
  mutate(count = n()) %>%
  mutate(count = as.numeric(count))
```

```
ggplot(dfsimallreps_summary, aes(x = as.factor(Years), y = count, fill = Species, col = Species))+
  geom_boxplot(alpha = .2, outlier.alpha = .02)+
  geom_smooth(aes(x = as.factor(Years), y = count, fill = Species, col = Species), size = 1)+
  ggtitle("Results for 1000 simulations")+
  xlab("Years since fire")+
  ylab("Tree count")+
  theme_bw()+
  scale_color_manual(values = c("#899DA4", "#9A8822"), labels = c("Fir", "Pine"))+
  scale_fill_manual(values = c("#899DA4", "#9A8822"), labels = c("Fir", "Pine"))+
  theme(text = element_text(size = 16),
        panel.grid.minor.x = element_blank(),
        panel.grid.major.x = element_blank(),
        axis.text.x = element_text(size = 8))
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



```
ggsave(file = "../../results/figures/sim_1000_count.png", width = 6, height = 4, dpi = 400)
```

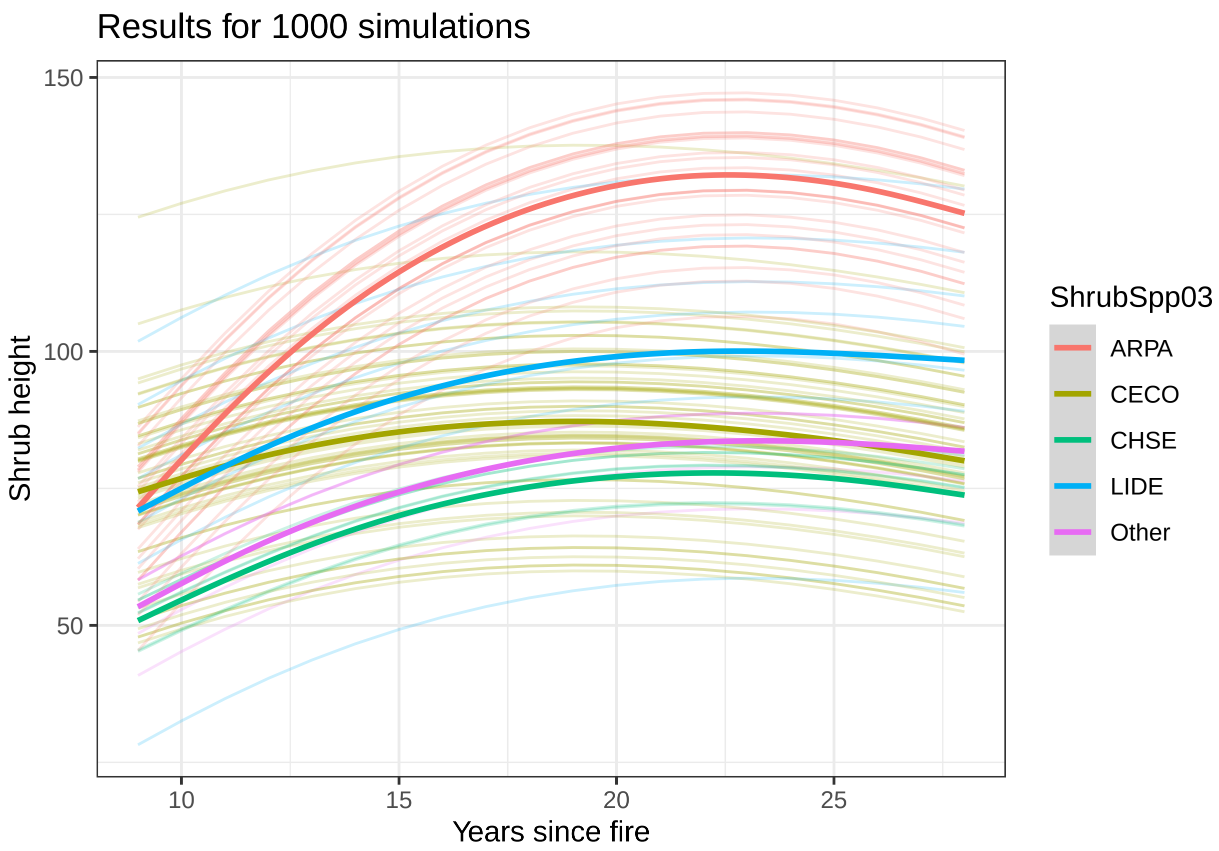
```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

Shrub height by year

```
dfsimallreps_summary <- dfsimallreps %>%
  ungroup() %>%
  group_by(rep, Years, Ht1.3, ShrubSpp03) %>%
  mutate(mean_shrub_ht = mean(Ht1.3))
```

```
ggplot(dfsimallreps_summary, aes(x = Years, y = mean_shrub_ht, group = ShrubSpp03, col = ShrubSpp03))+
  geom_line(aes(group = Sdlg), alpha = .2)+
  geom_smooth()+
  ggtitle("Results for 1000 simulations")+
  xlab("Years since fire")+
  ylab("Shrub height")+
  theme_bw()
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



```
print(Sys.time() - strt)
```

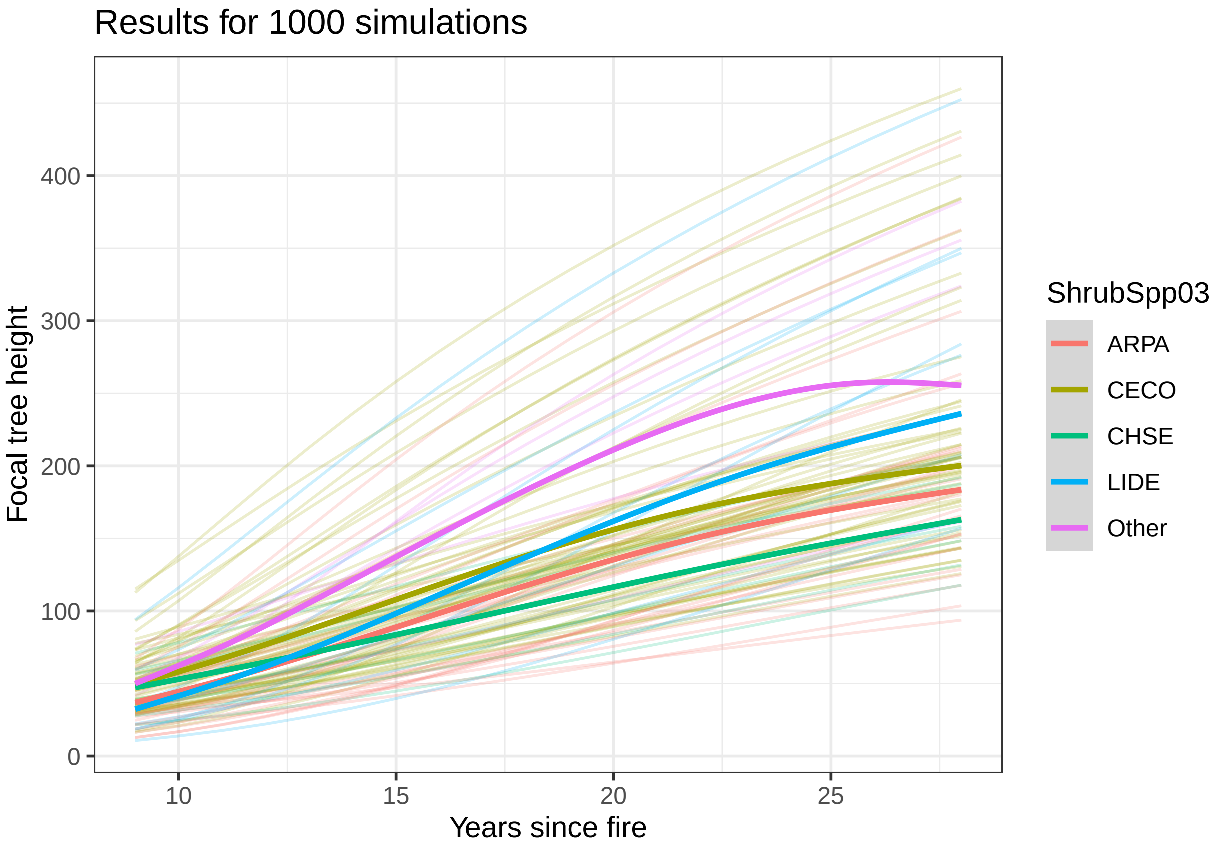
```
## Time difference of 1.608657 hours
```

Seedling height by shrub species and year

```
dfsimallreps_summary <- dfsimallreps %>%
  ungroup() %>%
  group_by(Species, rep, Years, Ht_cm1, ShrubSpp03) %>%
  mutate(mean_ht = mean(Ht_cm1))
```

```
ggplot(dfsimallreps_summary, aes(x = Years, y = Ht_cm1, group = ShrubSpp03, col = ShrubSpp03, symbol = 
  geom_line(aes(group = Sdlg), alpha = .2)+
  geom_smooth()+
  ggtitle("Results for 1000 simulations")+
  xlab("Years since fire")+
  ylab("Focal tree height")+
  theme_bw()
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



Next steps to improve the model

1. Use Kristen's data or Hugh's data for initial conditions
2. Improve dispersal kernel based on Kristen/Hugh's data
3. Improve shrub growth based on data - DONE
4. Include residual surviving trees and their seed dispersal
5. Include seed dispersal of post-fire regen once it reaches reproductive age
6. Add customization of patch size and shape
7. Add customization of whether the weather conditions reflect those of 2015, 2016, or 2017
8. Change sapling growth equations once they emerge from the shrub canopy

For next week: - Improve shrub growth based on data - DONE - display dominant shrub species - make the shrub grid dependent upon surrounding cells so it's not so checkerboard - DONE - Update display of shrub competition after simulation years - what does shrub competition mean for new recruitment? - "emergent year" = when 50% of trees are above shrub canopy - maybe submit to American Naturalist - Global Change Biology - mixing up the years - no overstory reproduction for now - apply to King, American River Complex, rest of the fires I measured - switch diameter equation to be from dendro work