Vignette

In the first part of this module, you explored data collected by K'avi tribal forestry managers for different sites of a forest that they managed. Based on data analysis and other qualitative factors, you decided which site would be the best for a lumber harvest.

Now that you have explored this forestry data set, it is time to present what you have learned to the broader community. You will create and present suggestions on the management of these forests to community leaders, using the data science tools you have developed throughout the semester. These leaders would like for you to present your analysis and the reasons for your choice of site.

We should flesh this out a little more.

WORKSHOP: Learning goals (More for us as a team, rather than for SKC students)
The goal of the second day of the forestry module is to have the students fully explore what creating and presenting recommendations for management based on data science means with a strong emphasis on how to best make and present graphs that are accessible and helpful for the general public.

Outlining your presentation

Determining your audience

The first step to making an effective presentation is having a clear picture of what you are trying to present. This comes from having a clear overview of the main outcomes of your data analysis. An important aspect of doing this successfully is what type of audience you are interested in presenting to and what type of information is most relevant to that audience.

First up lets work to identify the audience you are presenting to. Who would be involved in the process of deciding where to set up a new forestry area? Are there multiple groups that would be interested/involved in this process or just one? Who is included in these decision making groups? If you have identified multiple groups that would be involved in this process, for the sake of this exercise select one group you would like to present to.

Once you have identified who you will be giving your presentation to lets narrow down what kind of presentation they will be most interested in. What kind of background do they have in forestry? What kind of data do you think they would find most helpful in deciding where to set up this forest plot?

Determining What Data to Focus on

Now that you have identified your audience it is time to loo back at your data analysis from part 1 of this module. What were your main findings from that analysis? How do those findings relate to selecting a site for a new forestry patch?

Refining your graphs

For the rest of this module we will be focusing on different ways we can improve and customize ggplot graphs with a focus on honing those graphs specifically for a presentation for a more general audience. We will be demonstrating these concepts with data on the number of trees in the same potential forestry plots as in day 1 of this module and then letting you explore how to apply these concepts to your data on the yield over time in each of these sites.

```r
# Cleaning environment and setting working directory
rm(list = ls())
current_path = rstudioapi::getActiveDocumentContext()$path
setwd(dirname(current_path))
set.seed(1)
# Load libraries
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.2.3

## Warning: package 'ggplot2' was built under R version 4.2.3

## Warning: package 'tibble' was built under R version 4.2.3

## Warning: package 'tidyr' was built under R version 4.2.3

## Warning: package 'readr' was built under R version 4.2.3

## Warning: package 'purrr' was built under R version 4.2.3

## Warning: package 'dplyr' was built under R version 4.2.3

## Warning: package 'stringr' was built under R version 4.2.3

## Warning: package 'forcats' was built under R version 4.2.3

## Warning: package 'lubridate' was built under R version 4.2.3

## -- 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 ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- 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
```

```r
#Read in and reformat data
site_1 <- read.csv("site_1.csv") %>%
  mutate(site_ID = 1)
site_2 <- read.csv("site_2.csv") %>%
  mutate(site_ID = 2)
site_3 <- read.csv("site_3.csv") %>%
  mutate(site_ID = 3)
site_4 <- read.csv("site_4.csv") %>%
  mutate(site_ID = 4)
forest<- rbind(site_1, site_2, site_3, site_4)
```

The first part of making an effective graph is picking what kind of graph you want to make. We have generally been using scatterplots *and* _____ throughout this course but ggplot offers a wide variety of different plot types for you to play with. Generally the first step for picking a plot type is determining how many variables you are trying to represent and what your goal for the plot is.

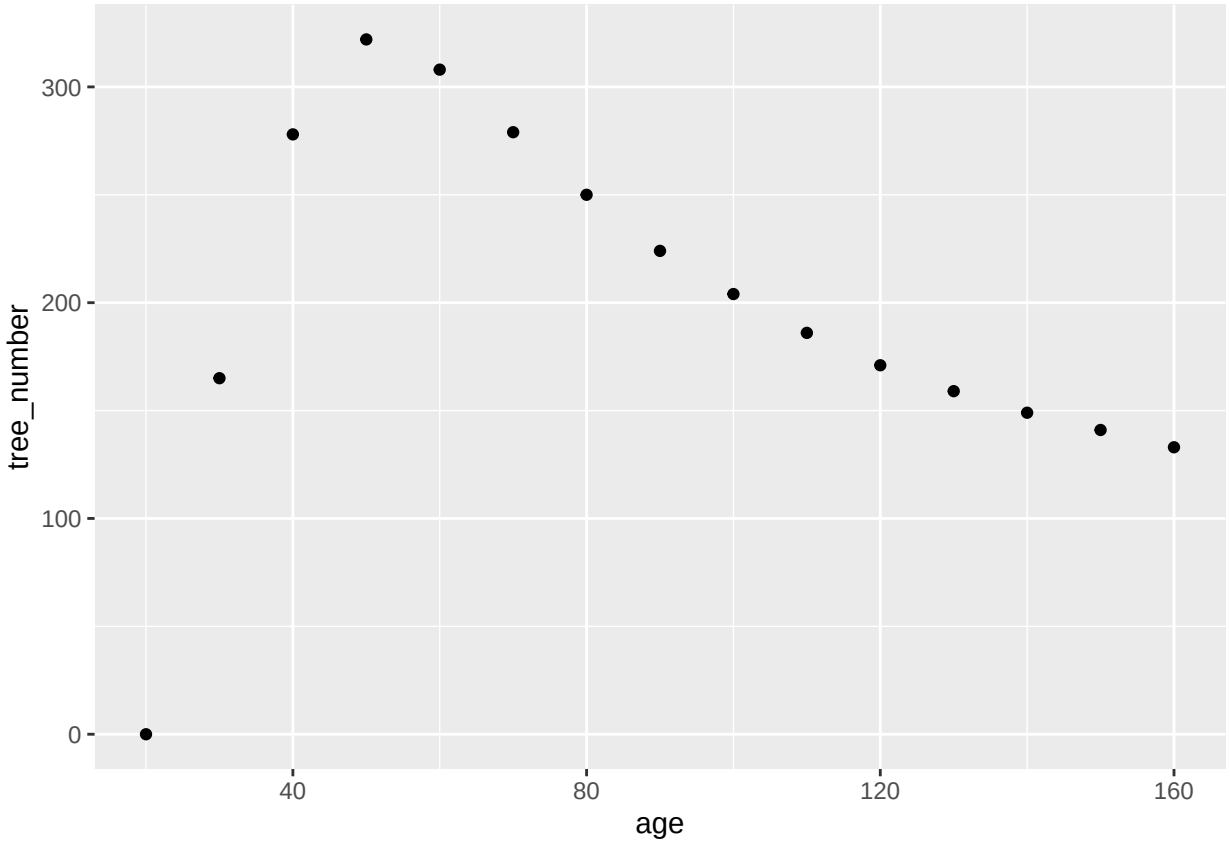
Here is a link to a graphic that explains what kinds of graphs are most useful in depicting different types of data, depending on the amount and kinds of variables that make up that data. Take a look at the graphic, but keep in mind that there are many graphs that we will not be discussing.

NOte: find something to put here. It also probably makes more sense to move this segment up

In our example of the number of trees over time at site 1 we have two different variables, time and number of trees. Both time and number of trees are continuously changing variables (not groups) so we would choose either a scatterplot or a line plot to represent them.

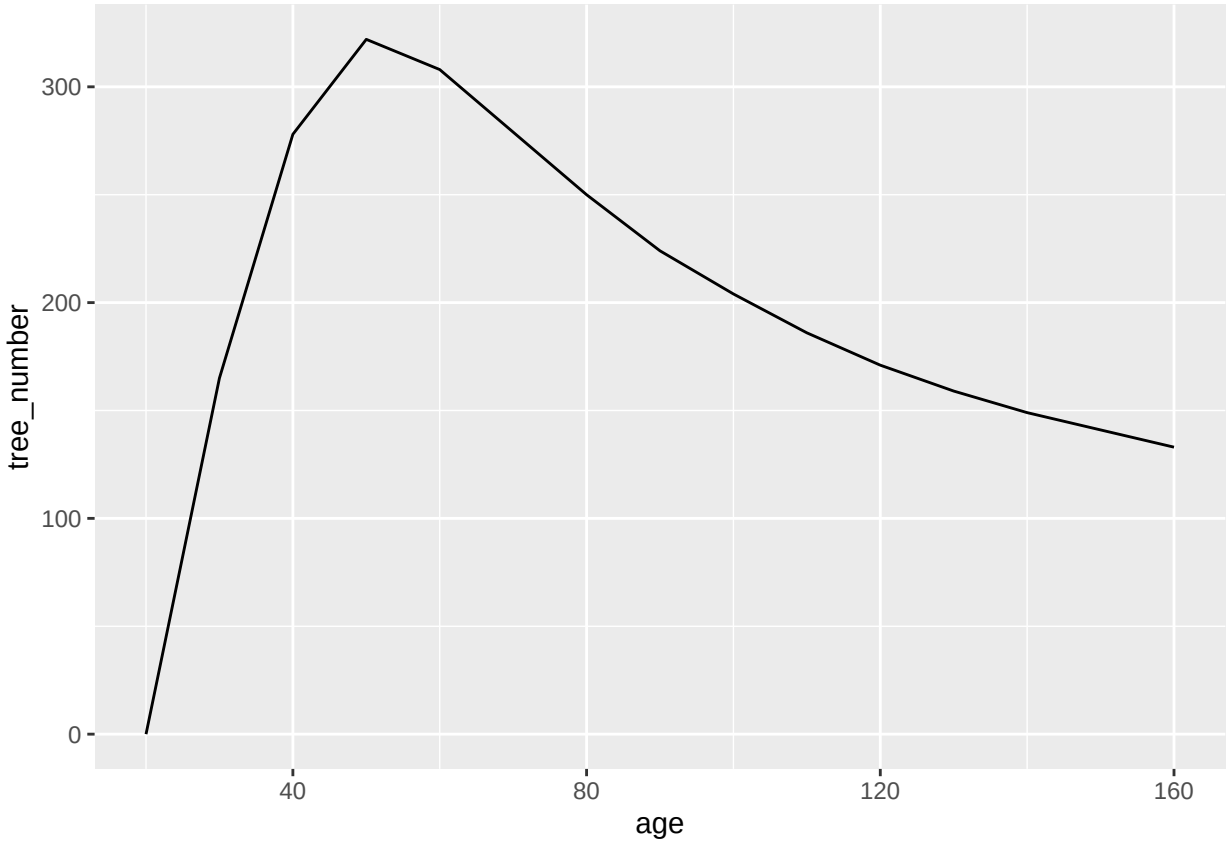
Here is an example of a scatterplot:

```
ggplot(site_1, aes(x = age, y = tree_number))+
  geom_point()
```



Here is a line plot:

```
ggplot(site_1, aes(x = age, y = tree_number))+
  geom_line()
```

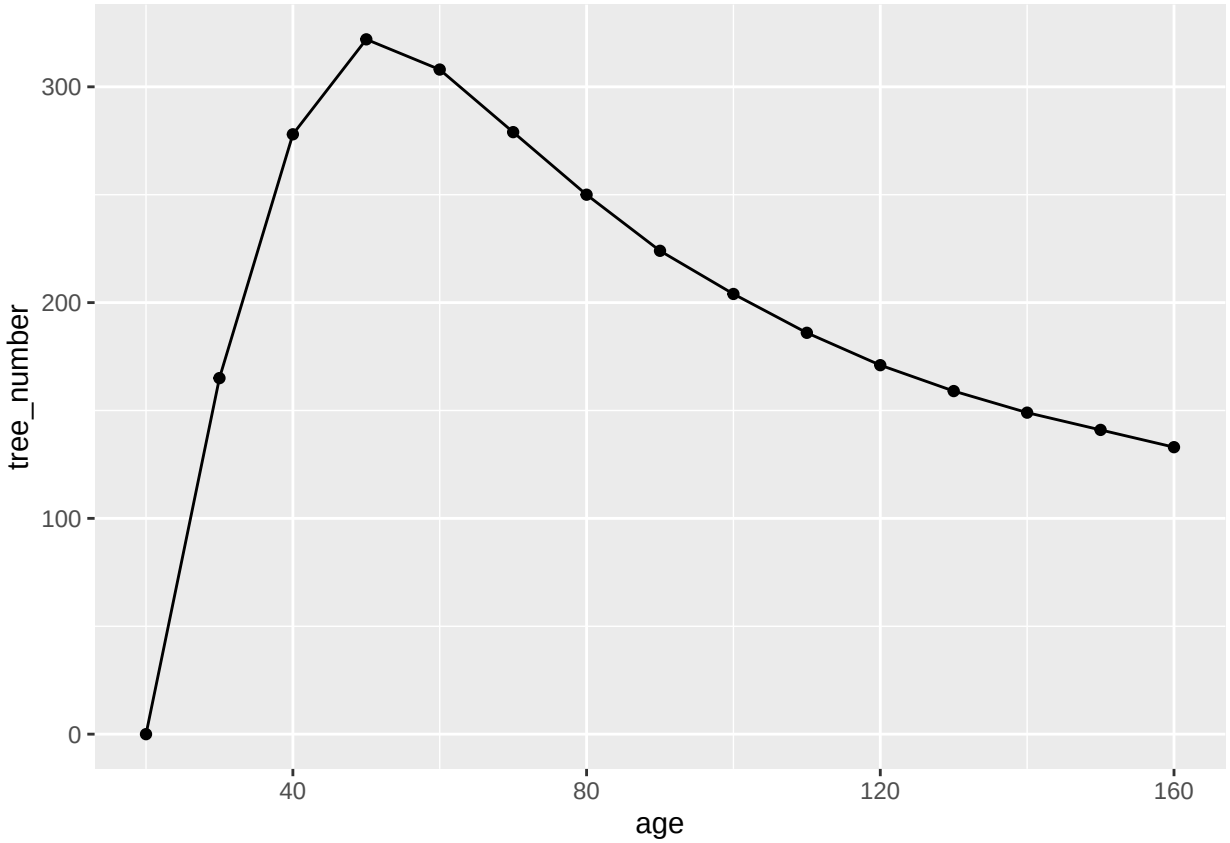


In this case the scatterplot focuses on the number of trees at each time point whereas the line plot focuses more on the overall trend in trees over time. Which of these do you think is more important for your presentation on yield over time at this site? What story do you want to tell?

You can plot whichever plot you decide on here:

Another cool thing that ggplot lets us do is layer different plots on top of one another. Using this we can show both the scatterplot and the line plot if we want to.

```
ggplot(site_1, aes(x = age, y = tree_number))+
  geom_line()+
  geom_point()
```



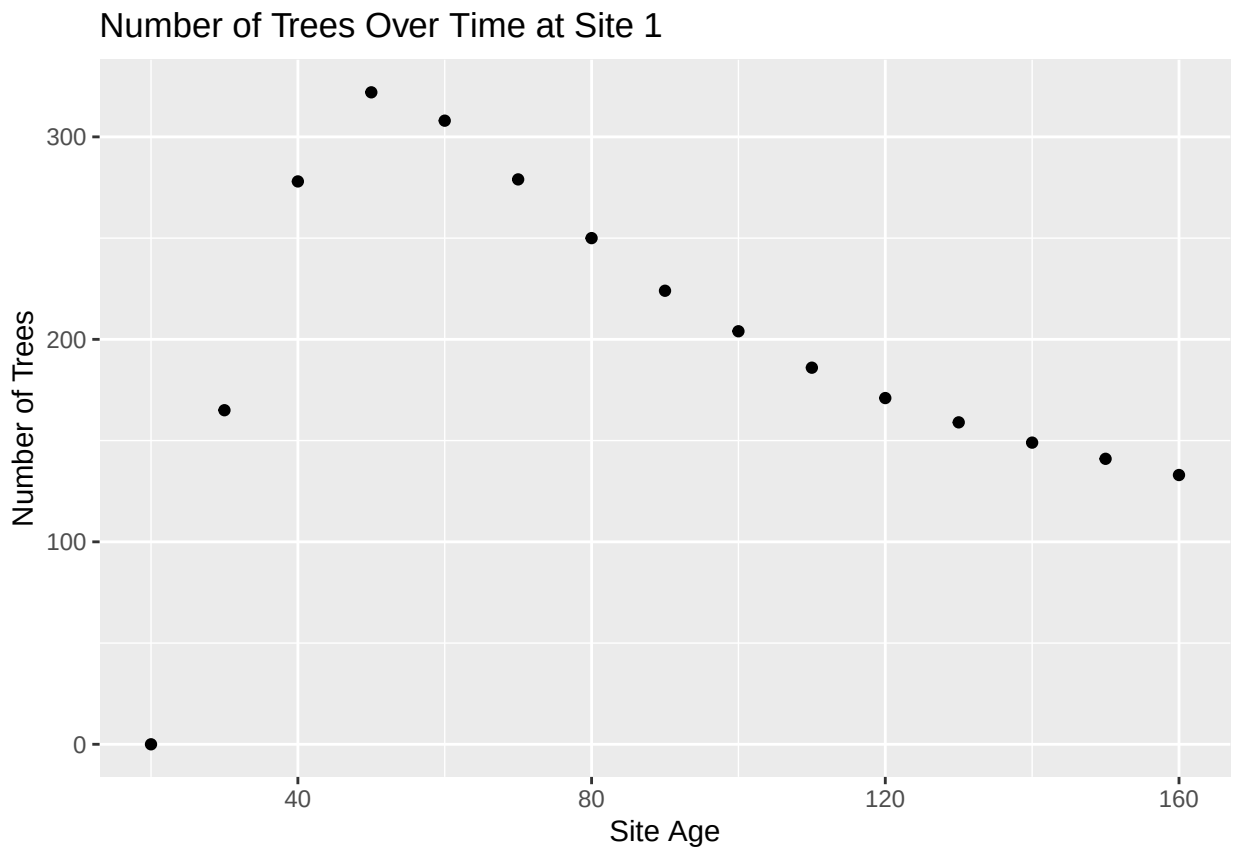
Try it out with your plot here:

Now that you have created a basic plot that technically has all of the information in it, if you could change anything about the graph what would you change about it to better fit the audience you identified in part 1?

Adding context

One of the first things we can do to make the graph more interpretable is change the text on the graph so that it is meaningful and helpful. We can do this using the labs() layer in ggplot. To add this layer all you need to do is add a plus sign at the end of the last layer (geom_point or geom_line) and then add the labs layer in the next line. You can use ?labs to learn about all the different things you can do in this layer but basically, it lets you add a title or caption, change the axis labels, etc. In the following plot we have changed the axis labels and added a title.

```
ggplot(data = site_1, aes(x = age, y = tree_number))+
  geom_point()+
  labs(x = "Site Age", y = "Number of Trees", title = "Number of Trees Over Time at Site 1")
```

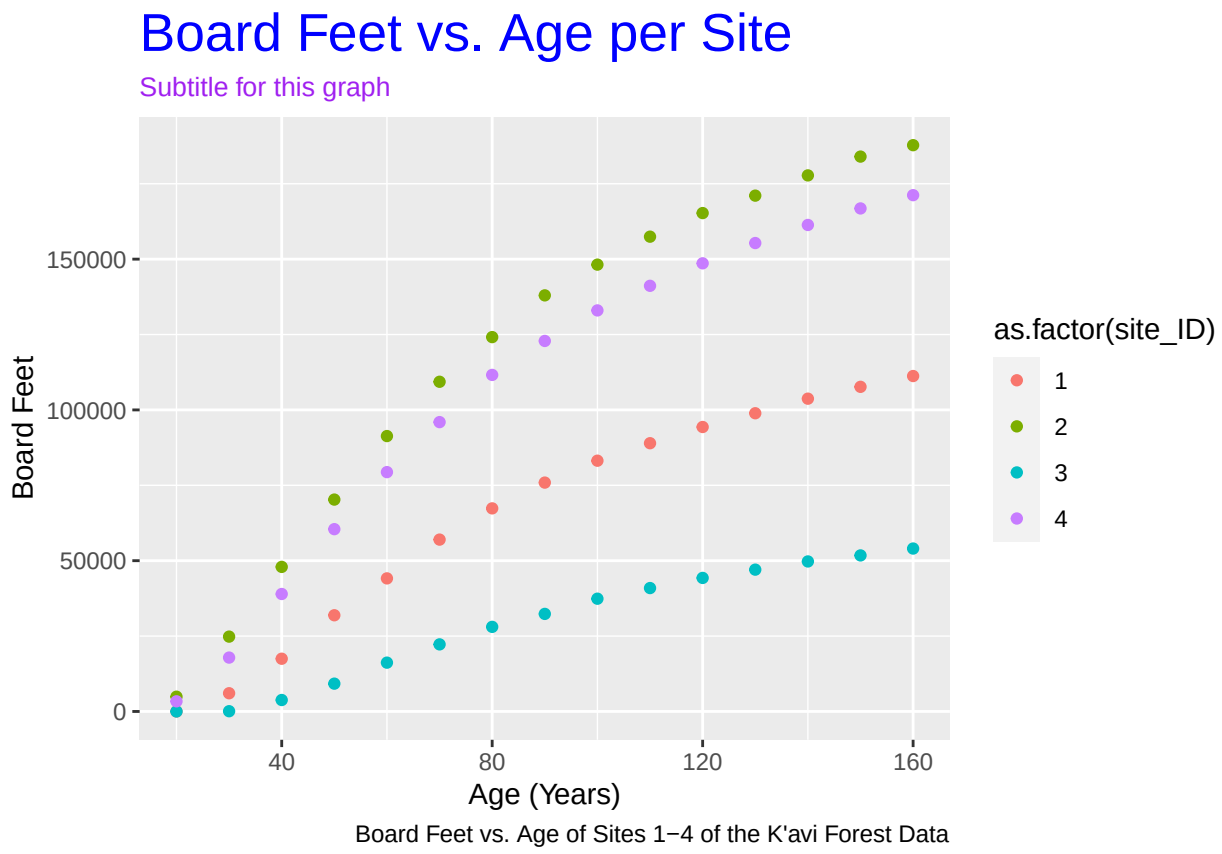



```
#The labs function allows you to adjust the labels of many different aspects of
#the graph simultaneously. Caption allows you to add a caption to the graph,
#while x and y adjust the names of the axes respectively. Finally, title allows
#you to rename the title of the graph and subtitle to add extra information.
```

While you add labels to your grpah below, think about the data that you are trying to convey to your chosen audience. How much background information/context will be useful to include on the graph that you are presenting to them? Will they know the units of the axes?

Take a look at the graph below. What are the differences between it and the one above?

```
#Calls the function ggplot (we have seen this before), add which data frame
#the graph is taking information from, and assign a variable to the x and y axes
ggplot(forest, aes(x = age, y = board_feet))+
#Tell ggplot what kind of graph to create (a scatter plot), in the aesthetics of
#the graph tell it to differentiate each site by a different color
  geom_point(aes(color = as.factor(site_ID)))+
#The labs function allows you to adjust the labels of many different aspects of
#the graph simultaneously. Caption allows you to add a caption to the graph,
#while x and y adjust the names of the axes respectively. Finally, title allows
#you to rename the title of the graph and subtitle to add extra information.
  labs(caption = "Board Feet vs. Age of Sites 1-4 of the K'avi Forest Data", x="Age (Years)", y="Board
#This allows you to adjust finer details of the graph. For instance, you can
#adjust the size and color of the title and subtitle.
  theme(plot.title = element_text(size= "20", color="blue"), plot.subtitle = element_text(size = "10",
```



Adding More Sites

This may need to be substantially edited depending on what previous lessons covered

Now that we have a plot that describes the trends at one site lets add in the information from the other sites. There are two main ways to do this, either by creating a different plot for each site and then arranging the plots next to one another, or adding all the sites to one plot.

First we will show you how to arrange multiple different plots side by side using the package patchwork. The first thing to do is download and then load the package itself.

```
# If you do not already have patchwork downloaded on your computer, delete the # in the next line to un
#install.packages(patchwork)

# Loading the package into this session of R
library(patchwork)
```

```
## Warning: package 'patchwork' was built under R version 4.2.3
```

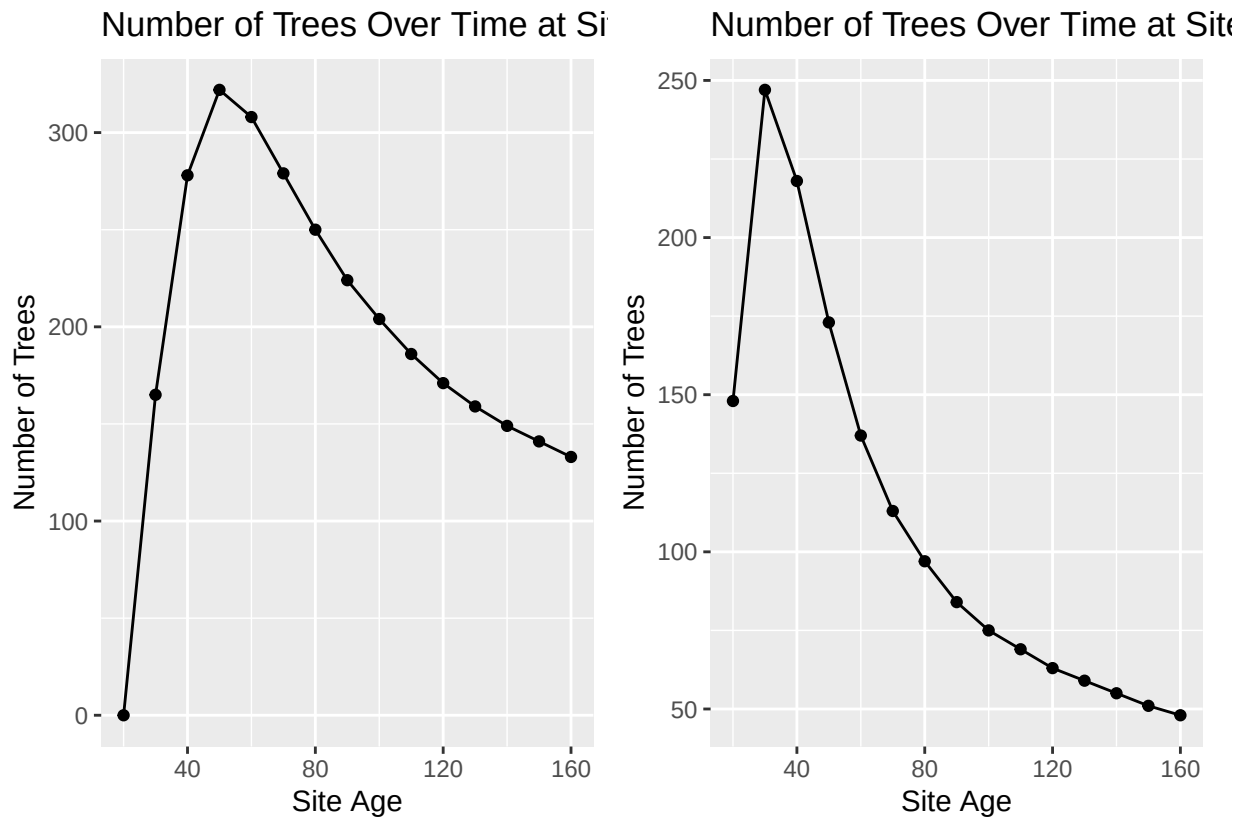
To use patchwork you need to create graphs for each of the sites, and then save each graph as its own variable the same way you would for any other variable.

```
site_1_plot <- ggplot(data = site_1, aes(x = age, y = tree_number))+
  geom_point()+
  geom_line()+
  labs(x = "Site Age", y = "Number of Trees", title = "Number of Trees Over Time at Site 1")
site_2_plot <- ggplot(data = site_2, aes(x = age, y = tree_number))+
```

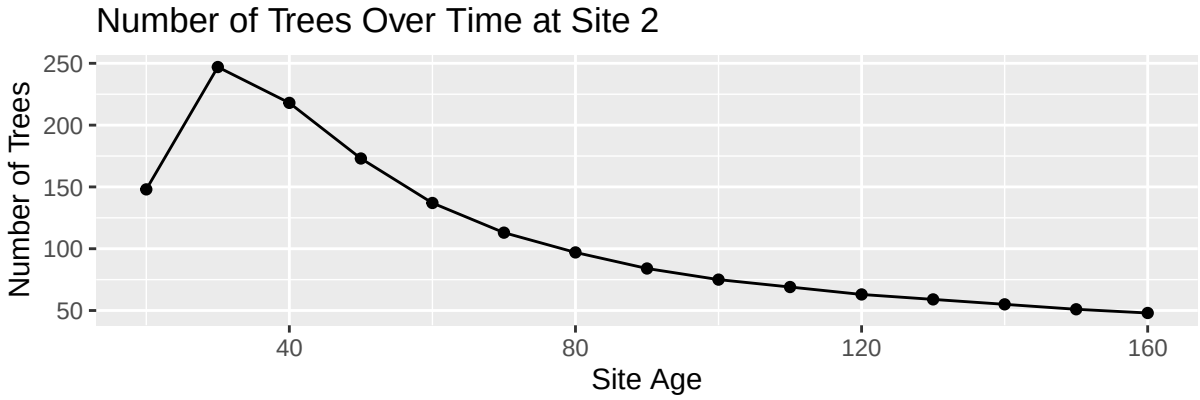
```
  geom_point()+
  geom_line()+
  labs(x = "Site Age", y = "Number of Trees", title = "Number of Trees Over Time at Site 2")
site_3_plot <- ggplot(data = site_3, aes(x = age, y = tree_number))+
  geom_point()+
  geom_line()+
  labs(x = "Site Age", y = "Number of Trees", title = "Number of Trees Over Time at Site 3")
site_4_plot <- ggplot(data = site_4, aes(x = age, y = tree_number))+
  geom_point()+
  geom_line()+
  labs(x = "Site Age", y = "Number of Trees", title = "Number of Trees Over Time at Site 4")
```

Then, to arrange the the graphs you can use + and / with the plots you saved above. I have provided a couple of examples below but you can see all the different ways you can sue this package to arrange graphs here: https://patchwork.data-imaginist.com/

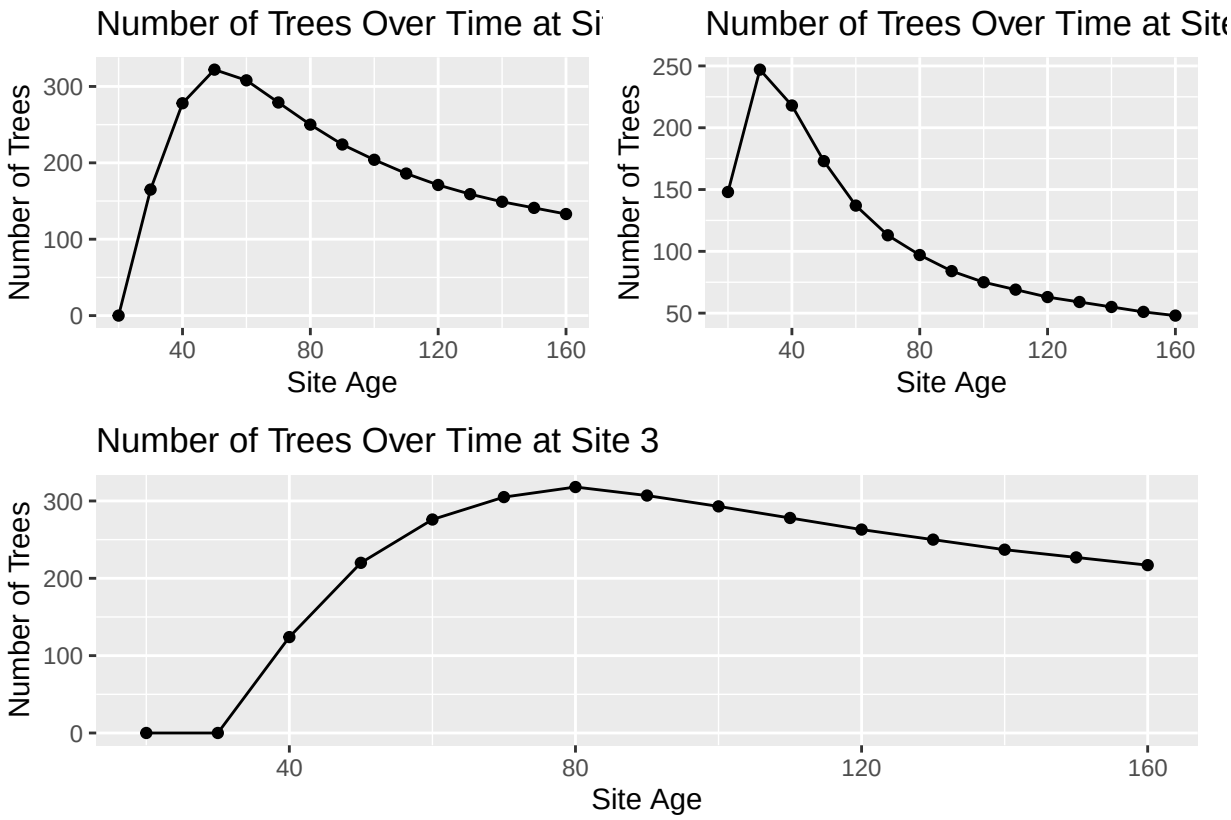
```
# + puts the two graphs next to one another
site_1_plot + site_2_plot
```




```
# / Puts the graphs on top of one another
site_1_plot / site_2_plot
```



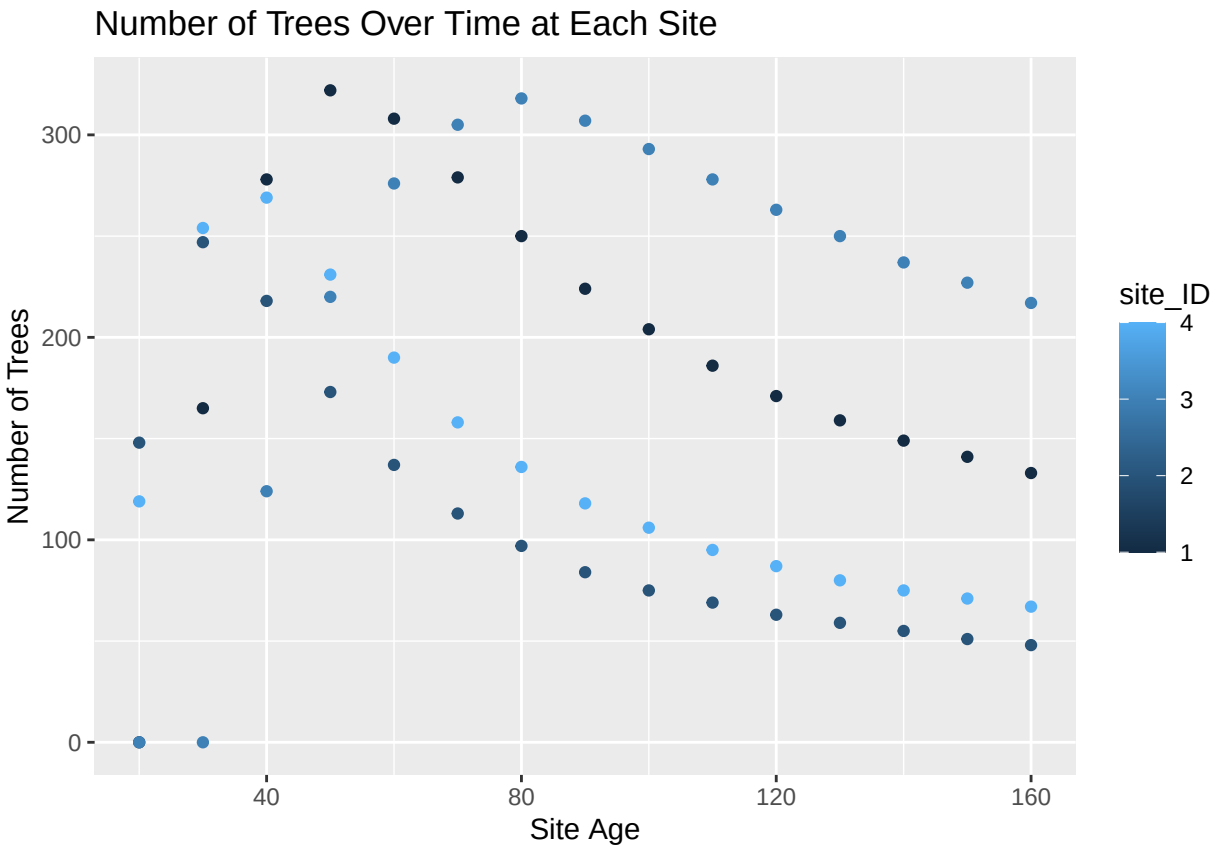

```
# Then you can get fancy with it!
(site_1_plot + site_2_plot) / site_3_plot
```



Experiment with patchwork with your own graphs below until you find an arrangement you like.

The second way to show data from multiple sites is to put all of the sites on one graph. We can do this in the aes() portion of the ggplot function by telling R to represent which site a point is from with another aspect of the graph. In our example we will use the color of the points and lines .

```
ggplot(data = forest, aes(x = age, y = tree_number, color = site_ID))+
  geom_point()+
  labs(x = "Site Age", y = "Number of Trees", title = "Number of Trees Over Time at Each Site")
```

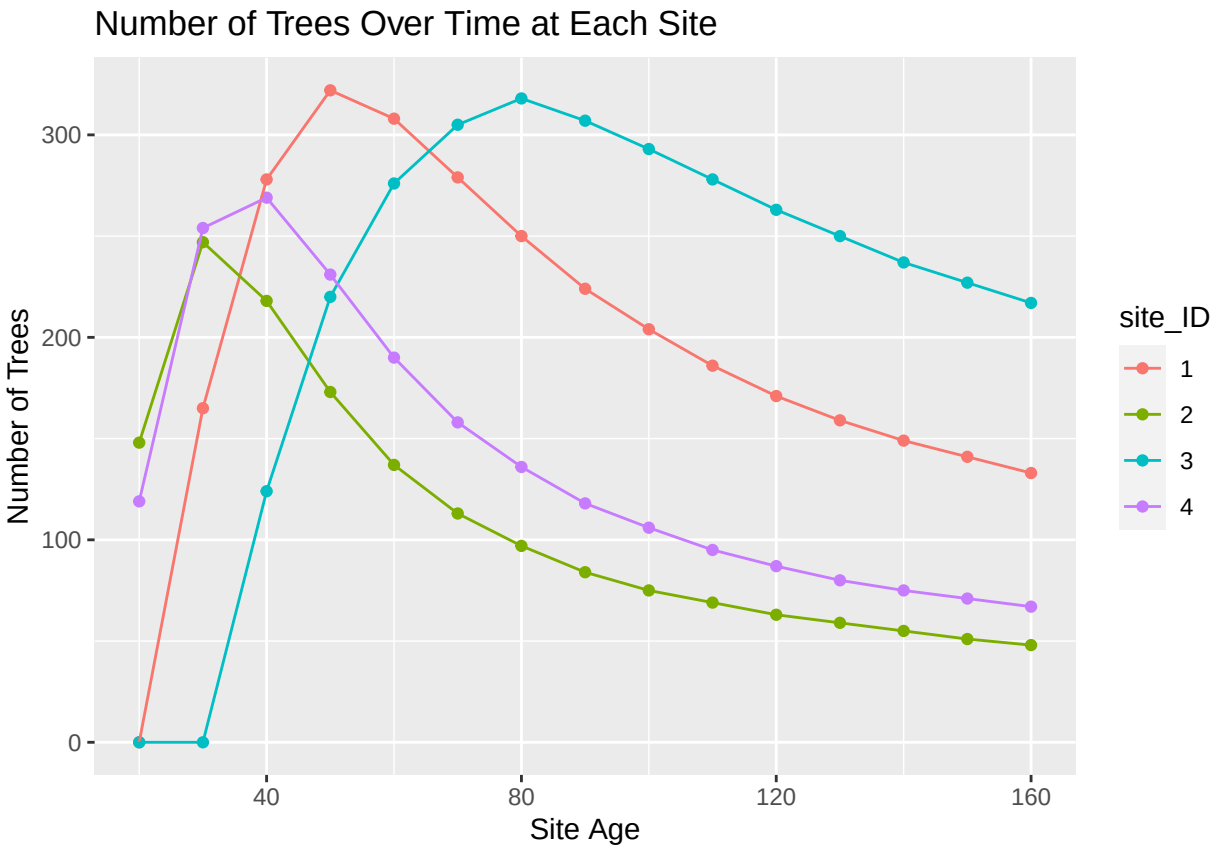


Note in this graph that the scale for the color on the right hand side of the plot is a continuous gradient of color rather than individual colors. This is because when we set up the site ID during the last lesson we set the ID of each site as a number. When we input a variable as a number, R automatically assumes that variable is a continuous, numerical variable. Since we are using the site ID as different categories for which site each observation is from we need this variable to be a factor (or categorical) variable rather than a number which we can do with the following code.

```
forest$site_ID <- as.factor(forest$site_ID)
```

Now we can use the exact same code as above and it will color the points in discrete colors for each site. We can also add back in the lines which were removed in the previous point because the line plots do not work well with the previous color scale.

```
ggplot(data = forest, aes(x = age, y = tree_number, color = site_ID))+
  geom_point()+
  geom_line()+
  labs(x = "Site Age", y = "Number of Trees", title = "Number of Trees Over Time at Each Site")
```



Try adding all the sites onto your graph here:

Both of these methods, arranging multiple graphs and adding colors, will show information on how the sites differ but they each have different pros and cons. In the arranged plots, each individual plot is cleaner and simpler where as there is a lot going on in the combined plot. However, the combined plot lets you see the differences between sites more clearly and it takes up less space in a slide or a paper.

Which format do you think will best convey the information to your proposed audience? Why?

While both options have merits, the combined plot with different colors lets us more easily address the other topics we want to cover about improving your graphs so moving forward in this lesson we will be refering back to the combined, color coded graph.

Legends

You may have noticed in the previous section that when you added color categories for different sites ggplot automatically created a legend on the right hand side of the plot to describe what each color means. However, like many things ggplot automatically generates, the legend is not particularly reader-friendly.

What would you like to change about your legend to tailor it to your specific audience?

Here are a couple of resources about how to change your legend in ggplot. They all cover similar things but you may prefer one over the others:

- http://www.cookbook-r.com/Graphs/Legends_(ggplot2)/
- https://www.datanovia.com/en/blog/ggplot-legend-title-position-and-labels/
- http://www.sthda.com/english/wiki/ggplot2-legend-easy-steps-to-change-the-position-and-the-appearance-of-a-graph-legend-in-r-software

Using one of the resources provided above or another resource that you prefer, change one thing about your legend that you have identified to make it more accessible to your audience.

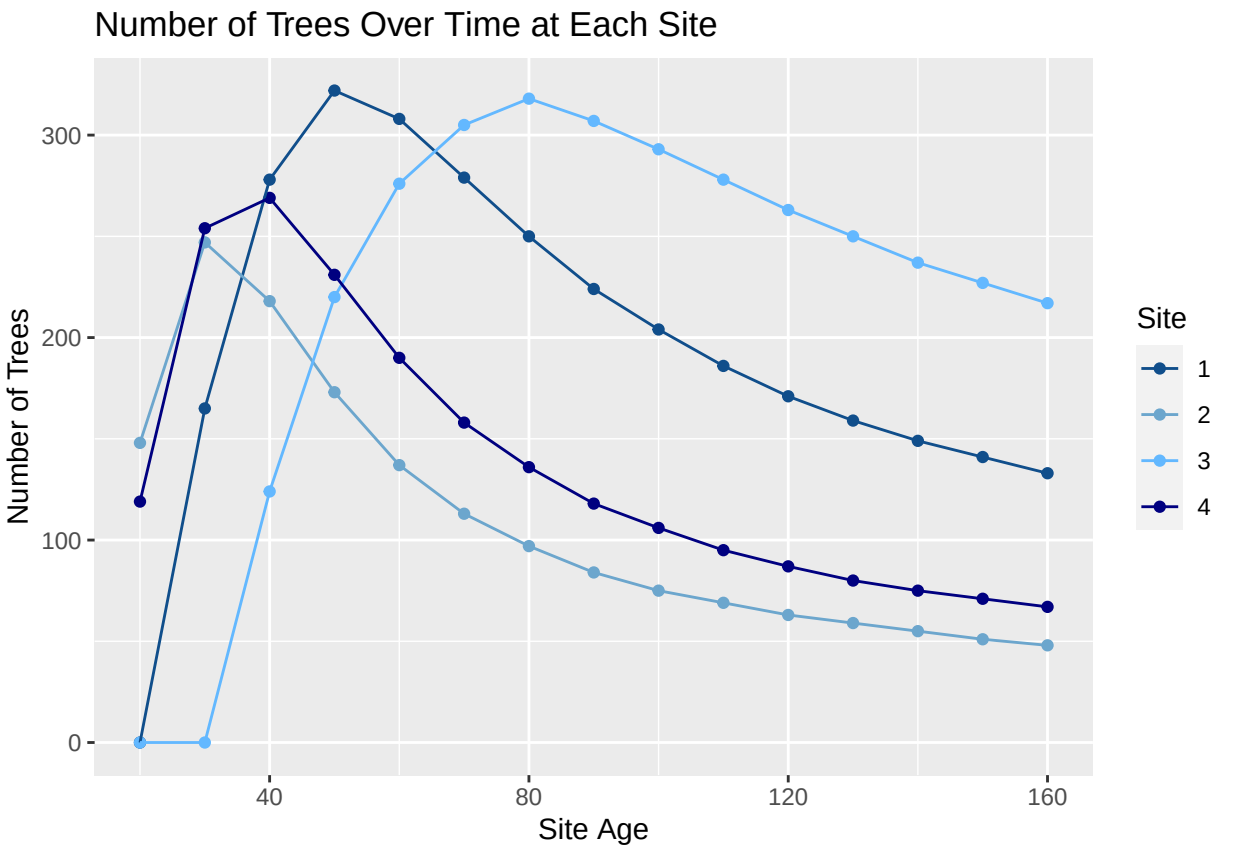
Adding visual elements

Now that you have a plot with multiple site, different colors, and a legend it's time to give you more control over what those colors are and other ways to show groupings.

To manually set what colors your plot you can use the layer scale_color_manual() which will let you individually choose each color in your plot (example below). The resources below will provide you with a list of all the colors that R automatically knows the names of. YOou can also use the function colors() to get a list of all 657 colors that R knows the name of but that does not actually show the color along with the name.

Colors: * https://www.datanovia.com/en/blog/awesome-list-of-657-r-color-names/ * http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf * https://r-graph-gallery.com/42-colors-names.html

```
ggplot(data = forest, aes(x = age, y = tree_number, color = site_ID))+
  geom_point()+
  geom_line()+
  labs(x = "Site Age", y = "Number of Trees", title = "Number of Trees Over Time at Each Site", color =
  scale_color_manual(values = c("dodgerblue4", "skyblue3", "steelblue1", "navyblue"))
```



Note that the number of colors you give R must equal the number of categories you have. Change the color in your plot here:

It is also important to note here that displaying important information with different colors in your plot can make reading your plots hard for colorblind people. There are a couple of different ways to help with

this including, picking a color palette that is colorblind friendly or using a couple of different methods to distinguish the different groups.

Go over the viridis package

Talk about changing the symbols for the points or the line types for the lines.

Talking about themes may be a stretch goal/not the most important part of this exercise. How do we feel about leaving this out? * Basic themes * Ways to make complicated themes

Reviewing your plots

Go back to the beginning of this document and look back at what you identified as the key takeaways from the data analysis. Take this as a chance to reflect on if your choice of graph is the best for summarizing this information

Review how you identified your audience, do your choices of added graph elements benefit this audience? If you could improve the graph in some way, how would you? (even if you don't know how to code this yet)

More Resources for Plotting in R