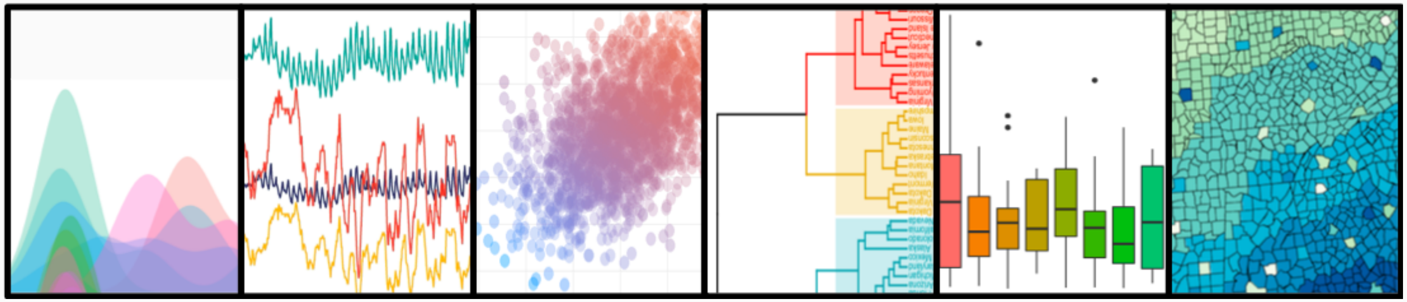
Open RStudio

1. go to www.github.com/collnell/GWU-visual to access workshop materials
2. download the `GWU_ggplot.R` script
3. open it in RStudio
4. make sure you have the `ggplot2` and `dplyr` packages installed

```r
install.packages('ggplot2')
install.packages('dplyr')

library(ggplot2)
library(dplyr)
```

Intro guide to R: www.rpubs.com/collnellphd/rbasics

Data visualization with R



Wednesdays 1-3 pm, SEH room 1800

Jan 30th: grammar of graphics in ggplot2

Feb 6th: Publication-ready figures

Feb 13th: Complex visualizations

Workshop materials:
www.github.com/collnell/GWU-visual

Colleen Nell

email: collnell@gwu.edu
website: www.collnell.com
twitter: @collnell
office: SEH 6880

Workshop materials

Workshop materials: www.github.com/collnell/GWU-visual

1. download the `GWU_ggplot.R` script
2. open in RStudio
3. modify arguments, adapt to data

Installing `ggplot2` and `dplyr` packages:

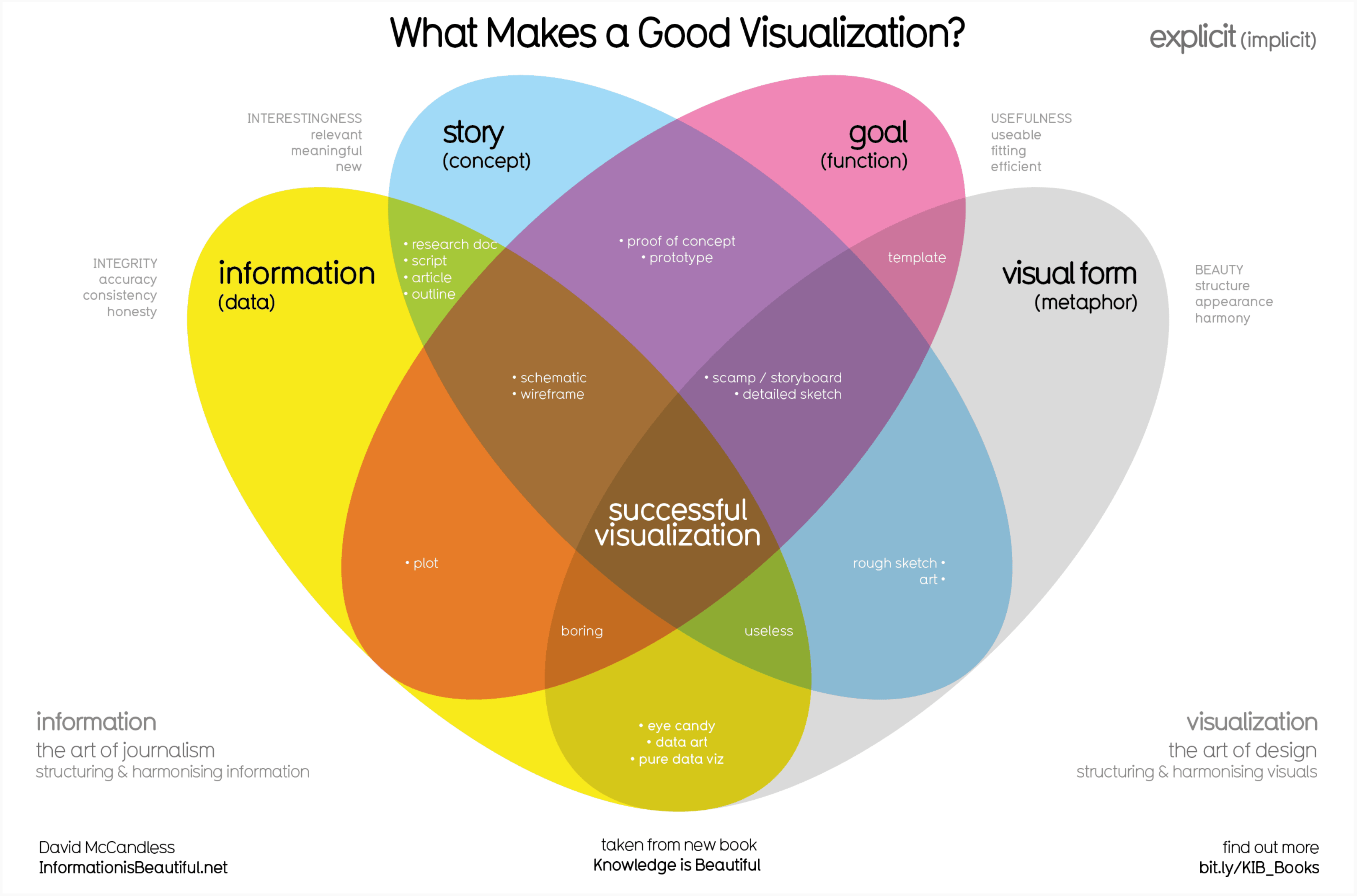
```r
install.packages('ggplot2')
install.packages('dplyr')

library(ggplot2)
library(dplyr)
```

Today's objectives

1. overview of ggplot2 capabilities
2. understand logic behind ggplot2 syntax
3. create common/useful figures types
4. customize plot appearance

Workshop materials: www.github.com/collnell/GWU-visual

Information is Beautiful



10 rules for better figures

modified from Rougier, Droettbroom, & Bourne (2014)

1. Know your audience
2. Identify your message
3. Adapt to medium (publication, poster, presentation)
4. Label, annotate, describe
5. Do not trust defaults
6. Use color intentionally
7. Represent data accureately, do not mislead the reader
8. Avoid unnecessary elements, less is more
9. Message > Beauty
10. Use the right chart & tools

Top 10 worst scientific graphs
R Graph Gallery

Workshop materials: www.github.com/collnell/GWU-visual

ggplot2

ggplot2 reference
ggplot2 cheatsheet
`ggplot2` is part of the tidyverse

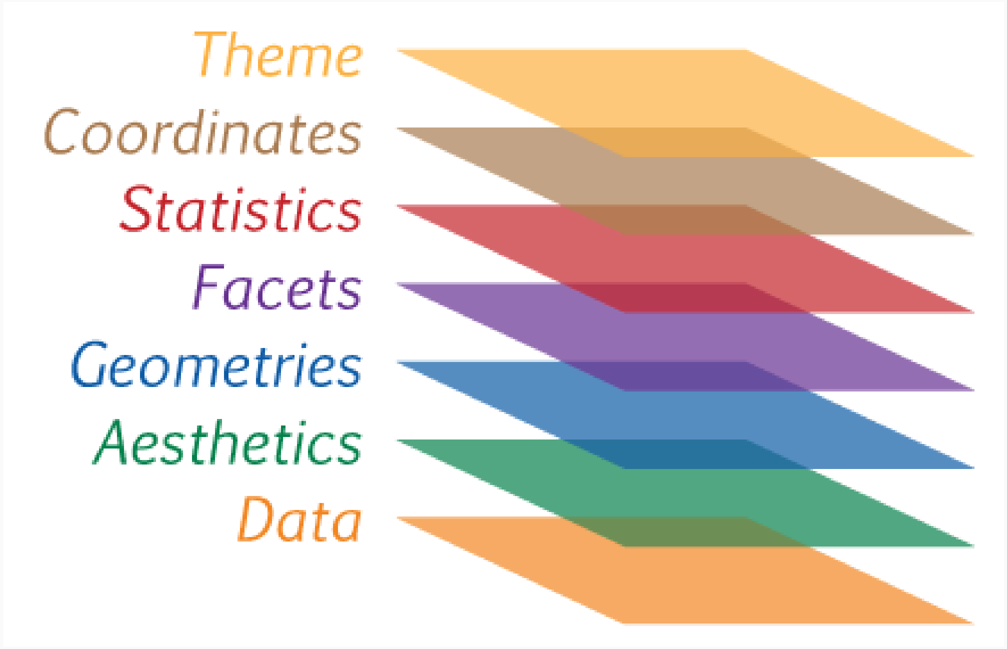
```r
install.packages('ggplot2')
library(ggplot2)

install.packages('tidyverse')
library(tidyverse)
```

```r
library(tidyverse)
```

Workshop materials: www.github.com/collnell/GWU-visual

ggplot2: a layered grammar of graphics

PLOT = data + geometric objects + coordinate system



data: tree diversity experiment

From a field experiment at the UADY forest diversity experiment in Mexico. Compared forest plots with 1 or 4 tree species (monoculture or polyculture) to test for tree diversity effect on:

- tree height
- bird communities - abundance, functional diversity, cwm.inv
- predation rates on clay caterpillars

data: tree diversity experiment

```r
# read in directly from github
birds<-read.csv('https://raw.githubusercontent.com/collnell/GWU-visual/master/bird_pred.csv')
```

What kind of figure?

data types

quantitative/numeric - continuous, discrete, counts, proportions
qualitative/categorical - groups, binary, ordinal

multivariate, 2+ variables

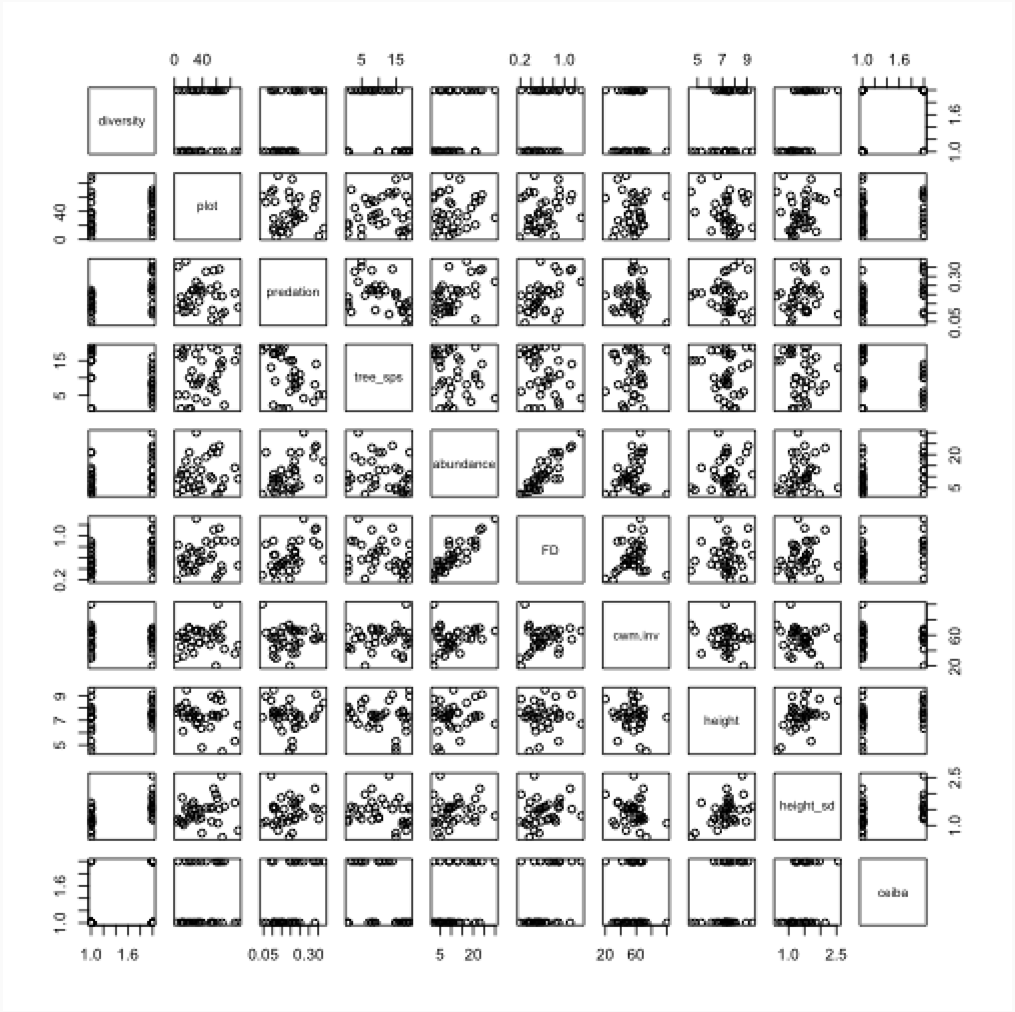
- complex structure - multiple treatments, groups, species, sites
- timeseries - dates, repeated measures
- spatial
- many observations, replicates, covariates, scales

```
str(birds)
```

```
## 'data.frame':    34 obs. of  10 variables:
##  $ diversity: Factor w/ 2 levels "M","P": 1 1 1 1 1 1 1 1 1 1 ...
##  $ plot     : int  3 9 12 13 17 20 21 23 30 35 ...
##  $ predation: num  0.12 0.186 0.152 0.104 0.106 ...
##  $ tree_sps : Factor w/ 19 levels "A","ABCD","ABCE",..: 17 1 18 17 19 1 10 1 17 19 ...
##  $ abundance: int  2 11 4 7 11 5 21 8 3 9 ...
##  $ FD       : num  0.189 0.581 0.48 0.332 0.649 0.352 0.788 0.456 0.352 0.534 ...
##  $ cwm.inv  : num  30 45 56.7 33 56 ...
##  $ height   : num  7.65 8.07 6.11 7.23 9.45 7.88 7.23 7.22 7.78 7.3 ...
##  $ height_sd: num  1.12 1.12 1.08 1.02 1.59 1.19 1.26 1.34 1.52 1.51 ...
##  $ ceiba    : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 2 1 1 1 ...
```

know your data

```
plot(birds)
```



ggplot()

Figures are made by **layering** different **geoms** that are defined by their **aestetic** mappings

```
ggplot()
```

Every plot starts the same way:

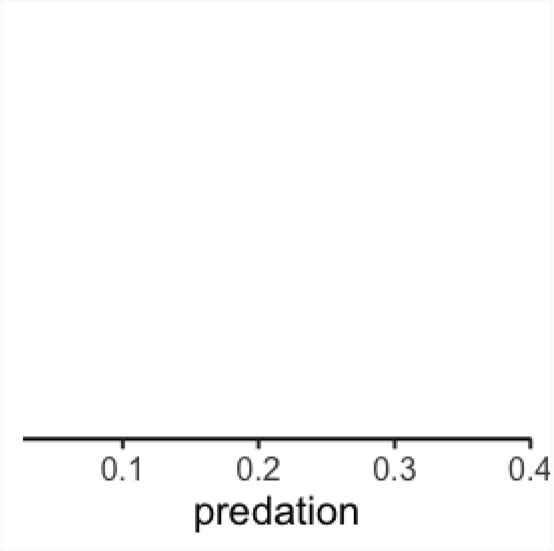
```
ggplot(data=birds, aes())
```

aes()

Aesthetic mappings = how data variables are tied to visual properties of `geoms`

- assign coordinates `(x,y)`
- color, fill, shape, size, alpha ++
- `aes()` maps data to the geom

```
# Assign 'predation' variable to the x-axis:
ggplot(data=birds, aes(x=predation))
```

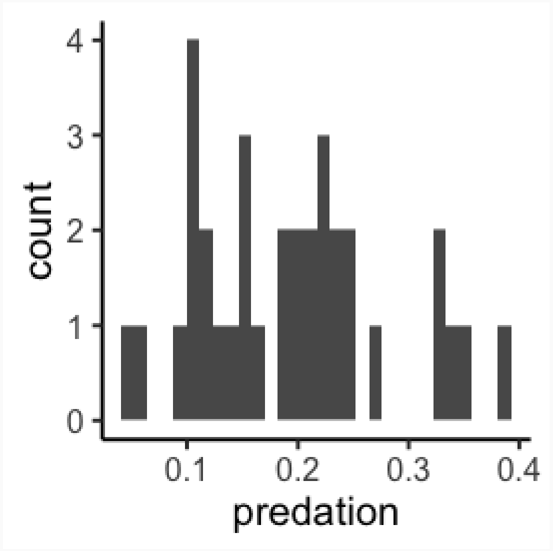


geoms

geometric objects define what shapes are used to represent the data
ggplot2 cheatsheet

```r
# Assign 'predation' variable to the x-axis:
ggplot(data=birds, aes(x=predation))+
  geom_histogram()
```

```
## stat_bin() using bins = 30. Pick better value with binwidth.
```



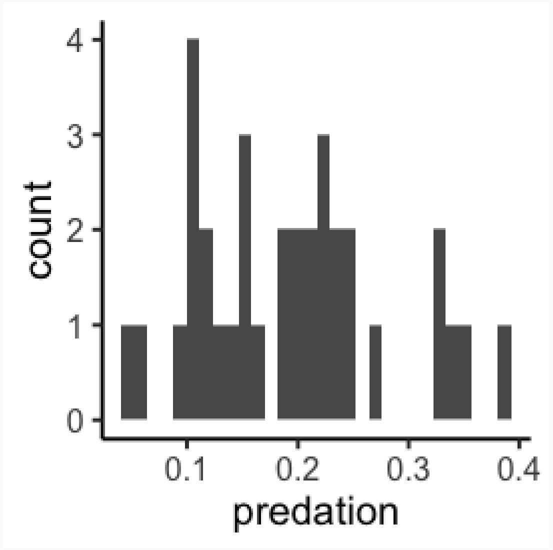
distributions

one variable - continuous

geom_histogram(), geom_density(), geom_freqpoly(), geom_area(), geom_dotplot()

```
ggplot(data=birds, aes(x=predation))+
  geom_histogram()
```

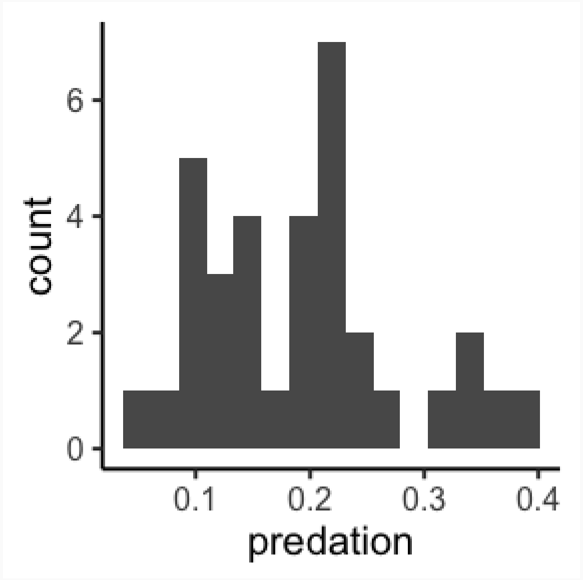
```
## stat_bin() using bins = 30. Pick better value with binwidth.
```



distributions: histogram

arguments: `bins`, `binwidth`, and `breaks`

```r
# bin = number of bins
ggplot(data=birds, aes(x=predation))+
  geom_histogram(bins = 15)
```

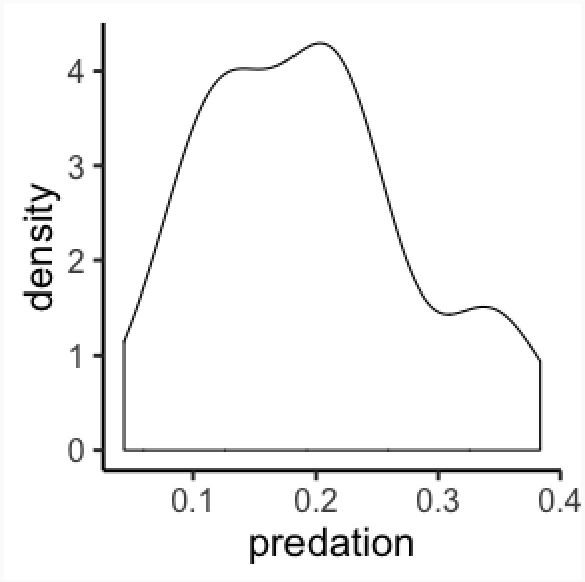


```r
# binwidth = size of bins (in units of variable)
ggplot(data=birds, aes(x=predation))+
  geom_histogram(binwidth = .05)
```

distributions: density plot

`bw` = smoothing bandwidth

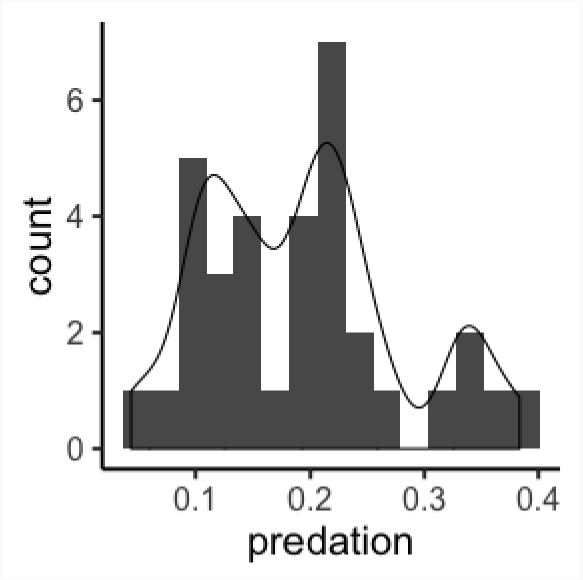
```
ggplot(data=birds, aes(x=predation))+
  geom_density()
```



layering

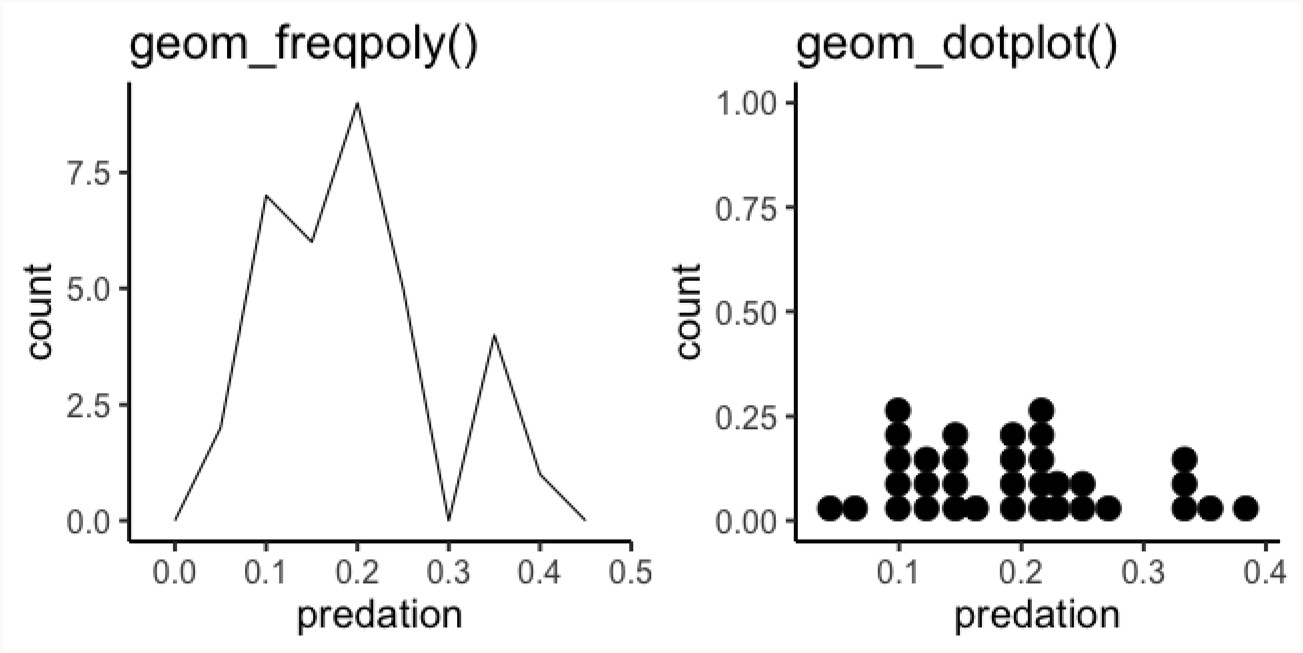
use `+` to add components to a plot

```
ggplot(data=birds, aes(x=predation))+
  geom_histogram(bins=15)+
  geom_density(bw = .02)
```



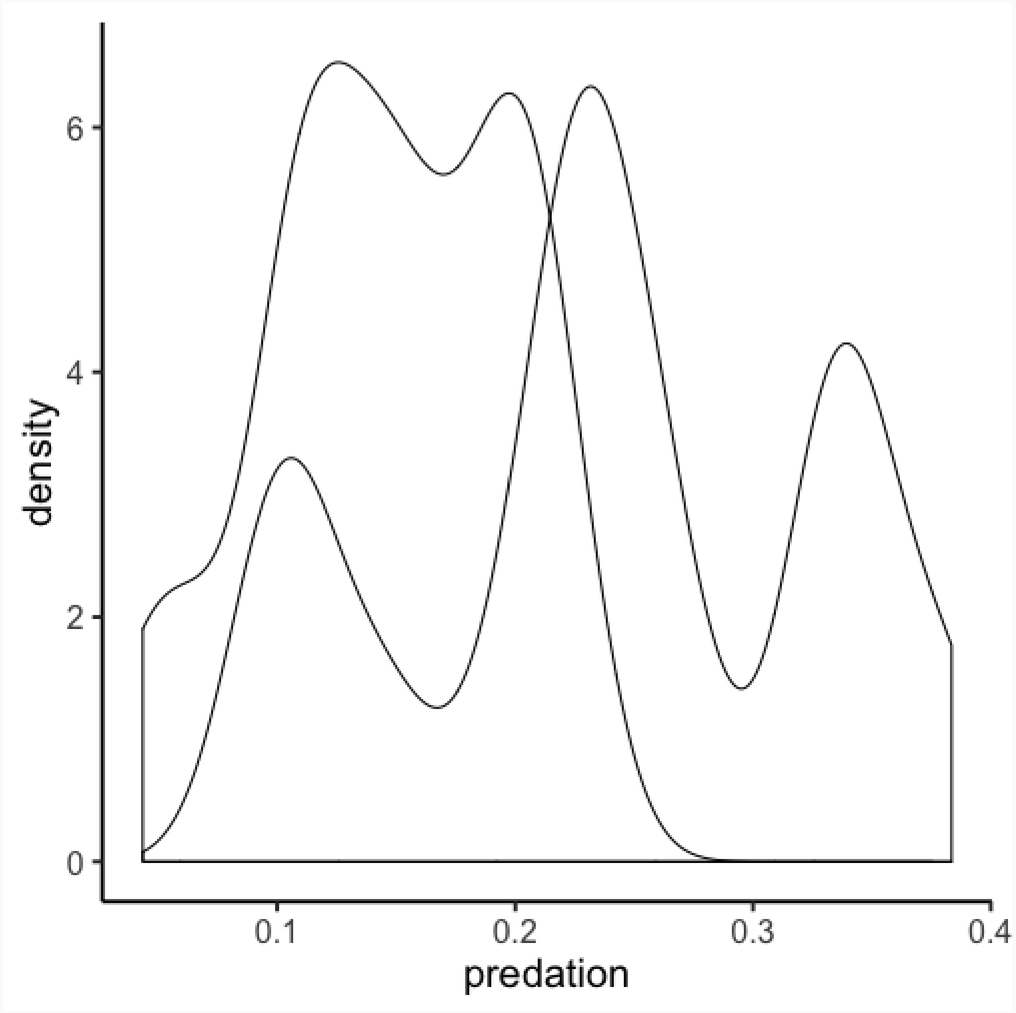
frequency polygon & dotplot

```r
ggplot(birds, aes(x=predation))+
  geom_freqpoly(binwidth = .02)

ggplot(birds, aes(x=predation))+
  geom_dotplot(binwidth= .02, method='dotdensity')# method = 'histodot' or 'dotdensity'
```



distributions: by categorical variable

```
ggplot(birds, aes(x=predation, group = diversity))+
  geom_density(bw = .02)
```

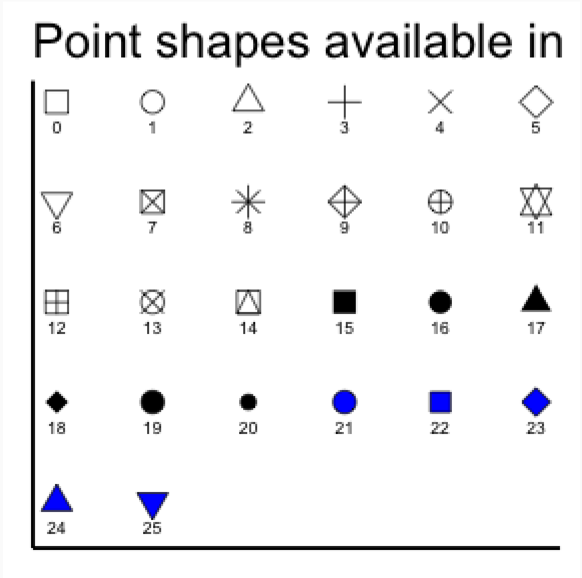


aesthetics

arguments inside the `aes()` are assigned to a variable, outside `aes()` is fixed

- position (x & y axes)
- `color` (“outside” color)
- `fill` (“inside” color)
- `shape` (of points)
- linetype
- `size` (width in mm)
- `alpha` (transparency; 0-1)
- `stroke`

```
## Scale for 'y' is already present. Adding another scale for 'y', which
## will replace the existing scale.
```

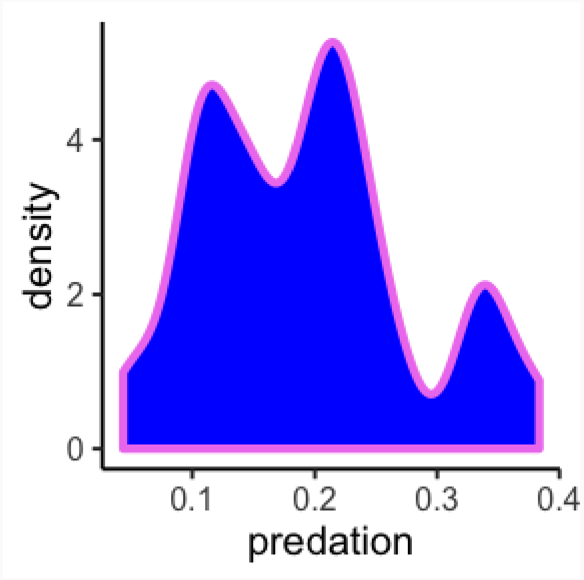


```
colors()
```

aesthetics: fill vs. color

`color` = 'outside' color
`fill` = 'inside' color

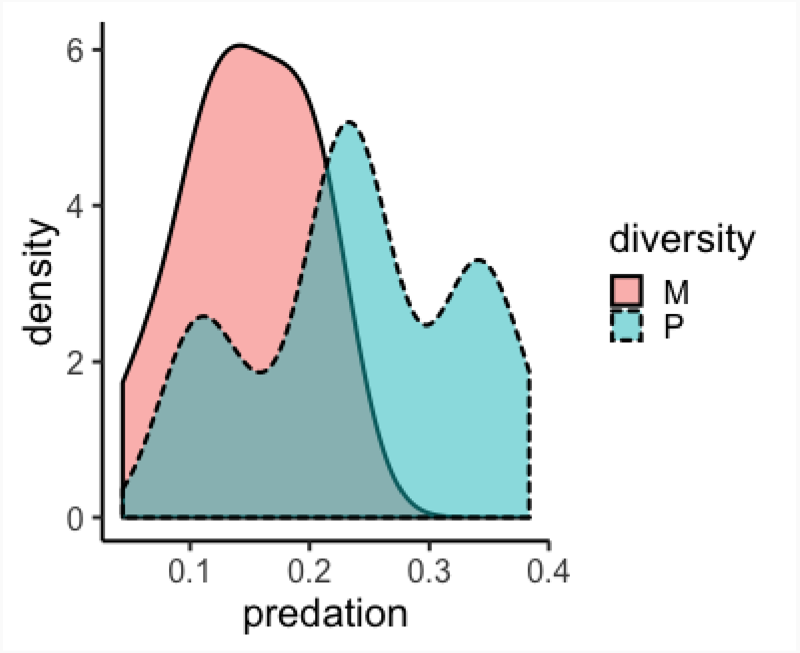
```
ggplot(birds, aes(x=predation))+
  geom_density(bw = .02, fill = 'blue', color='violet', size=2)
```



Challenge

Using the birds dataset:

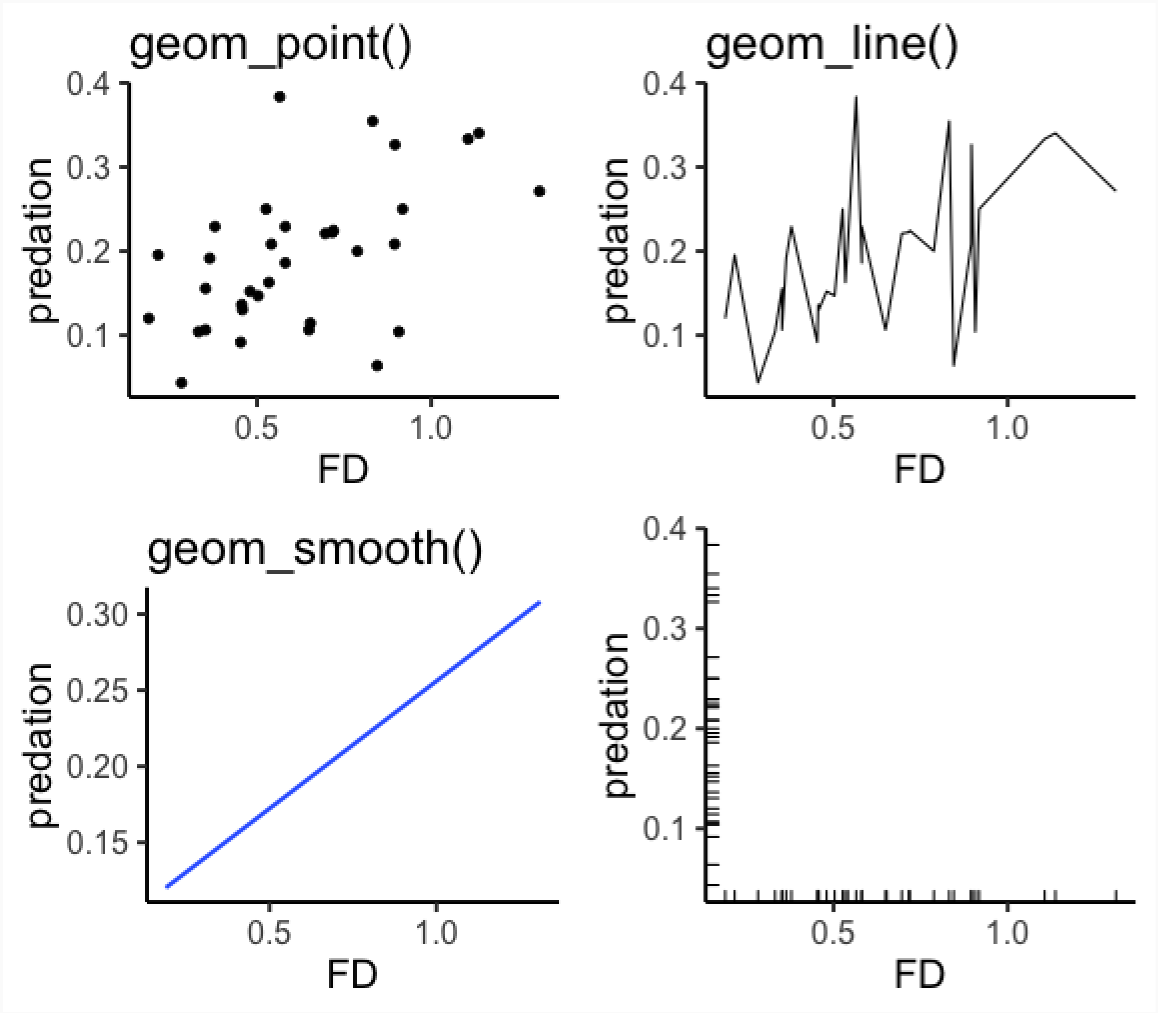
1. Make a density plot of predation by tree diversity, adjust binwidth (bw) as needed
2. Map fill & linetype to tree diversity
3. Adjust transparency (alpha) so overlapping area is visible
4. Increase line thickness to be able to distinguish linetypes



relationship: 2 numeric variables

relationships: 2 numeric variables

`geom_point()`, `geom_line()`, `geom_smooth()`, `geom_rug()`



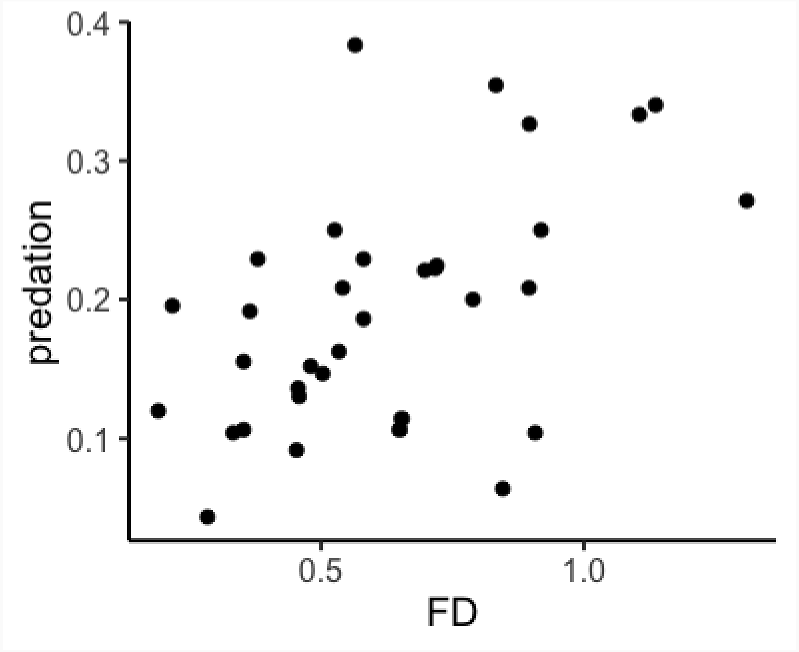
relationships

Show the relationship between predation and another numeric variable

relationships

Show the relationship between predation and another numeric variable

```
g<-ggplot(birds, aes(x=FD, y=predation))+
  geom_point(size=3)

g
```



add trendline

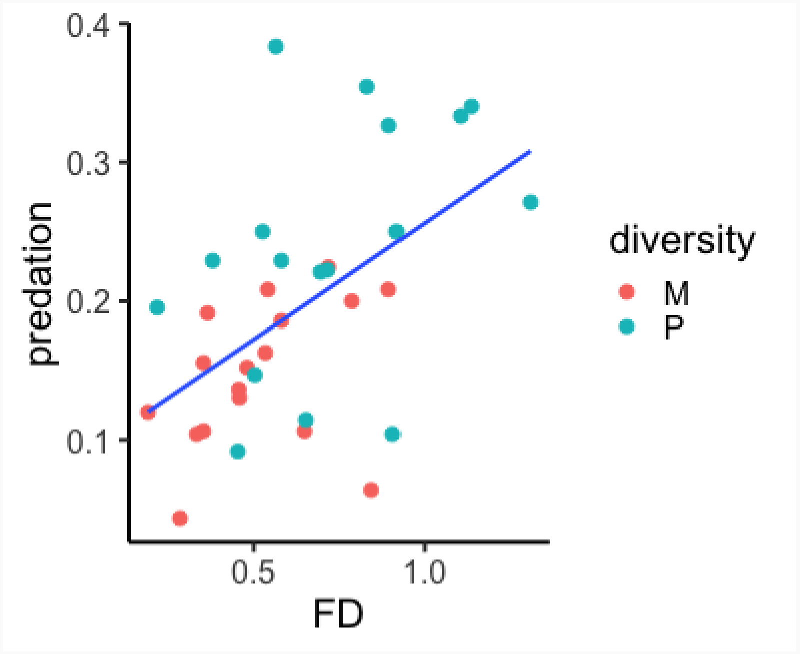
```r
g<-plot.fd<-ggplot(birds, aes(x=FD, y=predation))+
  geom_point(size=2)

g+geom_smooth(method='lm')
g+geom_smooth(method='loess') # smoothed fit curve
g+geom_smooth(method='lm', se=FALSE) # remove confidence around line
g+geom_smooth(method='lm', se=FALSE)+geom_rug()
```

layering & aesthetics

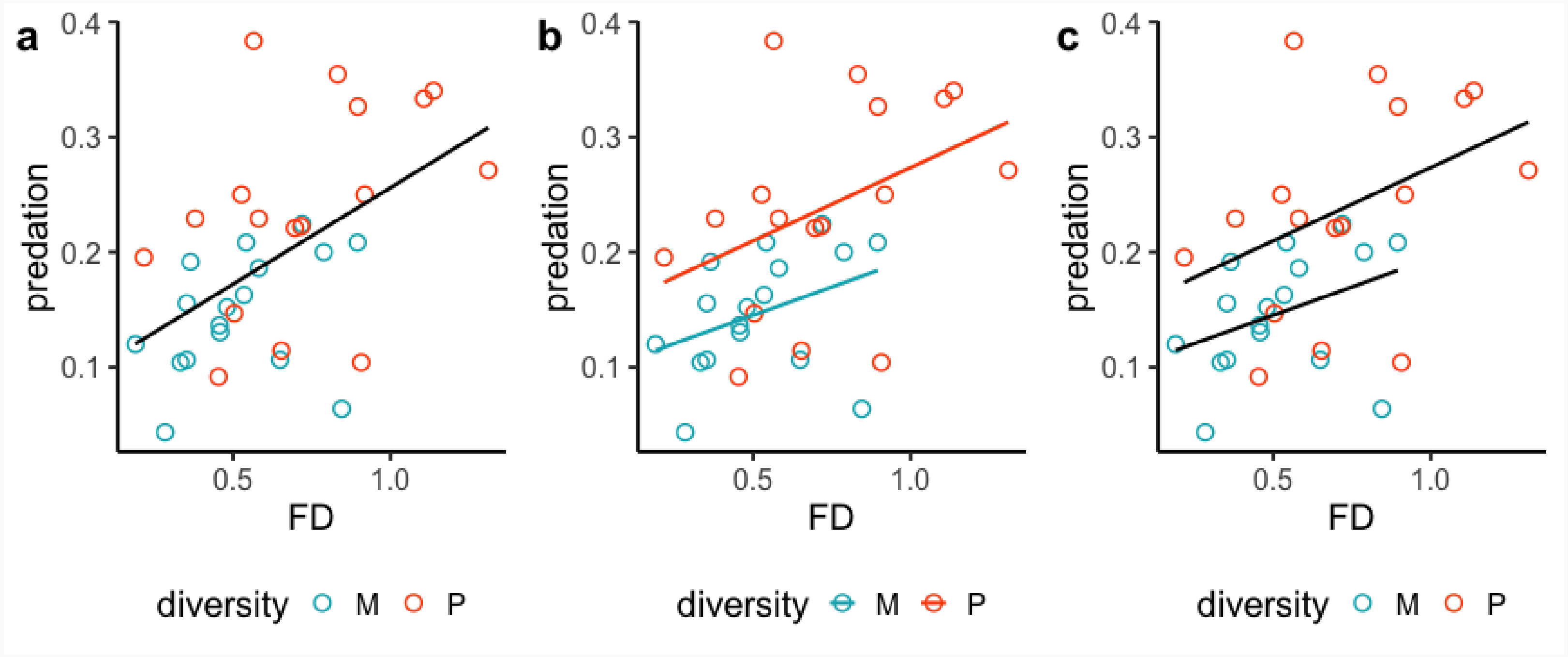
- arguments in `ggplot()` are applied to ALL layers
- data and `aes()` can also be mapped to individual geoms

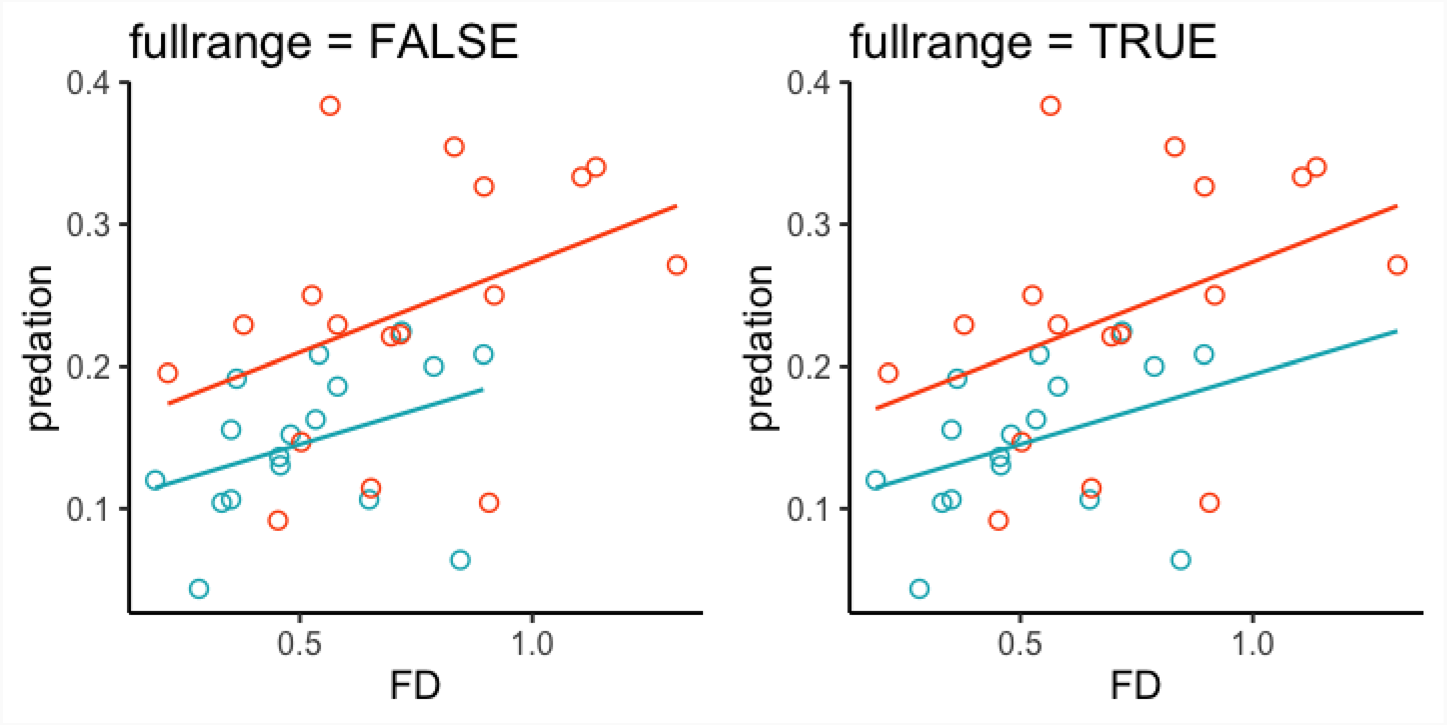
`color` by diversity:

```
ggplot(birds, aes(x=FD, y=predation))+
  geom_point(size=3, aes(color=diversity))+
  geom_smooth(method = 'lm', se=FALSE)
```

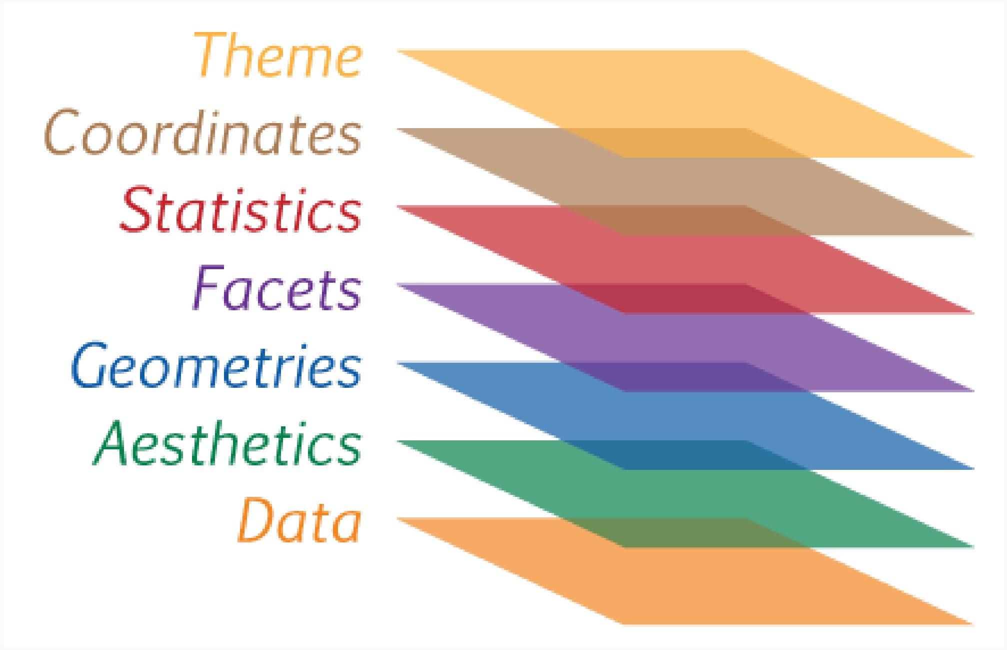


how are they different?




fullrange = FALSE
fullrange = TRUE
predation
FD
0.1
0.2
0.3
0.4
0.5
1.0

what's missing?

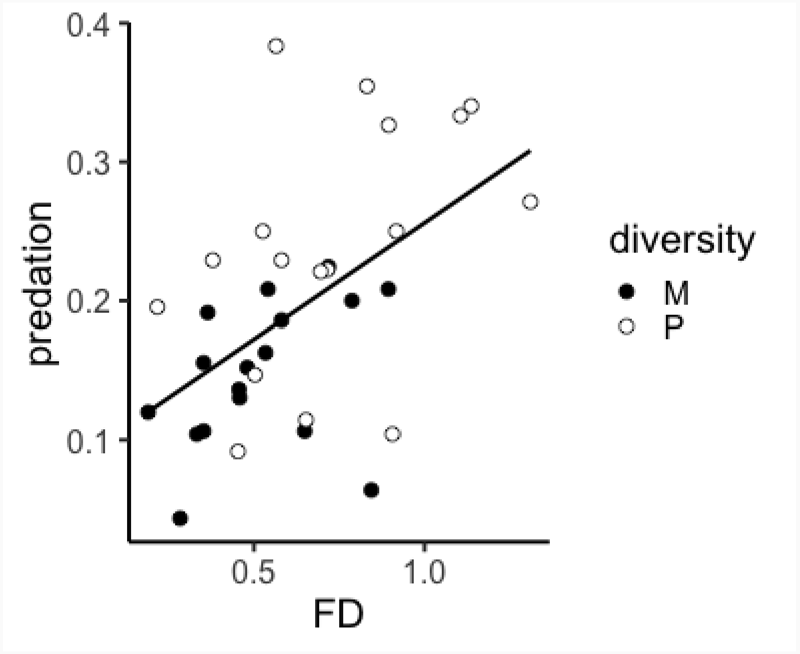


scales: controlling aesthetic mapping

how a variable is mapped to an aesthetic (`color`, `size`, `shape` etc)

`scale_[aesthetic]_[type]()`
`scale_color_manual()`, `scale_fill_manual()`, `scale_shape_manual()`, `scale_linetype_manual()`
`scale_color_manual()`, `scale_color_discrete()`, `scale_color_gradient()`

```
ggplot(birds, aes(x=FD, y=predation))+
  geom_point(size=3, shape = 21, aes(fill=diversity))+
  geom_smooth(method = 'lm', se=FALSE, color='black')+
  scale_fill_manual(values=c('black', 'white'))
```



scales: gradients

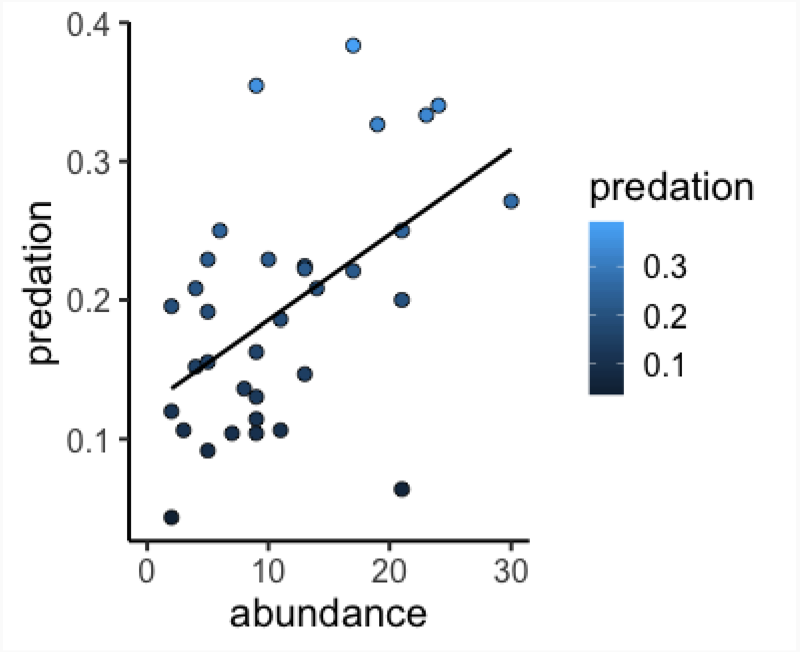
`scale_color_gradient()` - 2 color gradient
`scale_color_gradientn()` - with n colors
`scale_color_gray()`
`scale_color_gradient2(low = , mid= , high = )` - diverging color gradient

```r
g<-ggplot(birds, aes(x=abundance, y=predation))+
  geom_point(size=3, shape = 21, aes(fill=predation))+
  geom_smooth(method = 'lm', se=FALSE, color='black')


g+scale_fill_gradient(low = 'red',high = 'yellow')
```

scales: axes

scales are also used to set axis limits

```r
g+xlim(0, NA) # ensure axis starts at 0
```



```r
#g+xlim(0, NA)+ylim(0, NA)
```

scales: position

`scale_x_discrete()`

`scale_x_continuous()`

`scale_x_log10()`

`scale_x_sqrt()` `scale_x_reverse()`

legends

```r
g<-ggplot(birds, aes(x=FD, y=predation))+
  geom_point(size=3, shape = 21, aes(fill=diversity))+
  geom_smooth(method = 'lm', se=FALSE, color='black')

g+scale_fill_manual(values=c('black', 'white'), name='Tree diversity', labels=c('Monoculture', 'Po
  theme(legend.position = 'bottom') # top, bottom, left, right or none
```

+ labs()

arguments: `x =`, `y =`, `title =`

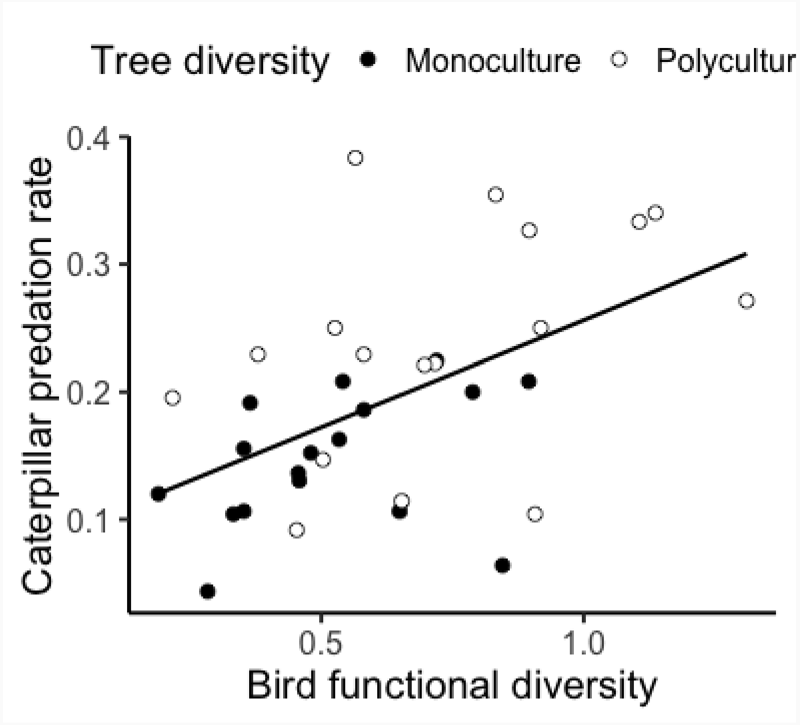
```
gg<-g+scale_fill_manual(values=c('black', 'white'), name='Tree diversity', labels=c('Monoculture',
  theme(legend.position = 'bottom')+
  labs(x = 'Bird functional diversity', y='Caterpillar predation rate')
gg
```

themes

themes set the appearance of non-data elements

- plot background
- gridlines
- fonts
- legend appearance

```
gg+theme(legend.position='top')
```



built-in themes

Customizing themes

The function `theme()` is used to control non-data parts of the graph including:

Line elements: axis lines, minor and major grid lines, plot panel border, axis ticks background color, +++
`axis.line`, `axis.line.x`, `axis.line.y`, `plot.grid.major`, `panel.border`
Text elements: plot title, axis titles, legend title and text, axis tick mark labels, +++
`axis.title`, `axis.title.x`
Rectangle elements: plot background, panel background, legend background, ++
`panel.border`

There is a specific function to modify each of these three elements:
`element_line()` to modify the line elements of the theme
`element_text()` to modify the text elements
`element_rect()` to change the appearance of the rectangle elements
`element_blank()` to remove theme element

Customizing themes

```
gg+theme_classic(base_size = 18)

gg+theme_classic(base_size = 18)+theme(panel.border = element_rect(fill=NA, size=2))

gg+theme_classic(base_size = 18)+theme(axis.line = element_blank())
```

Challenge

Using any data:

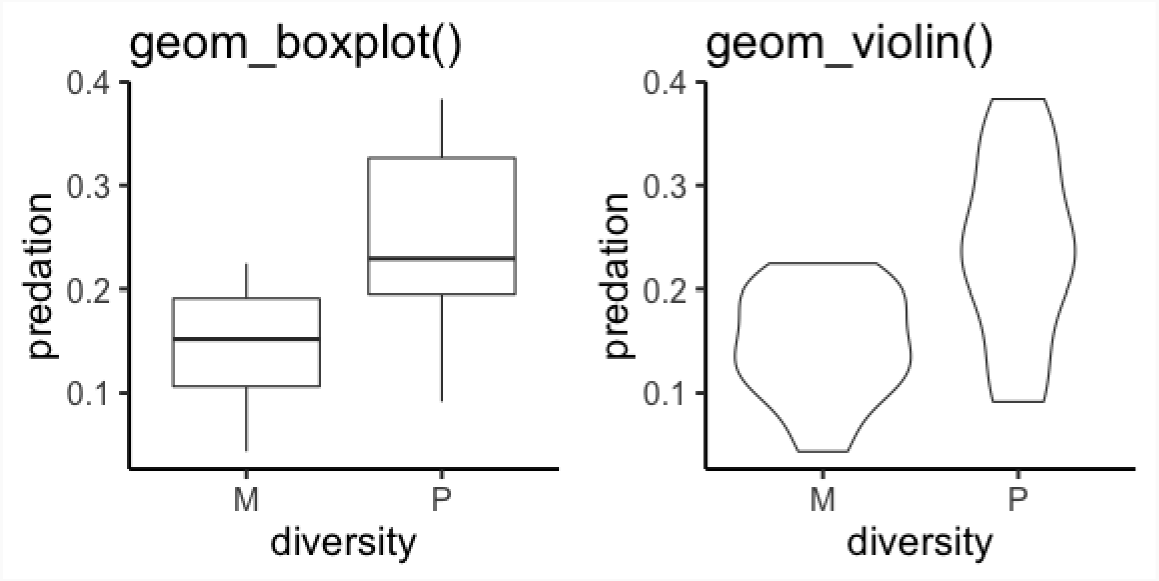
1. Show the relationship between 2 continuous variables
2. Use aesthetics and scales to customize appearance
3. Adjust labels and theme

comaprison: categorical & numeric

comparisons

`geom_boxplot()`, `geom_violin()`, `geom_bar()`, `geom_point()`, `geom_jitter()`, `geom_linerange()` ++

```
ggplot(birds, aes(x = diversity, y = predation))+
  geom_boxplot()

ggplot(birds, aes(x = diversity, y = predation))+
  geom_violin()
```

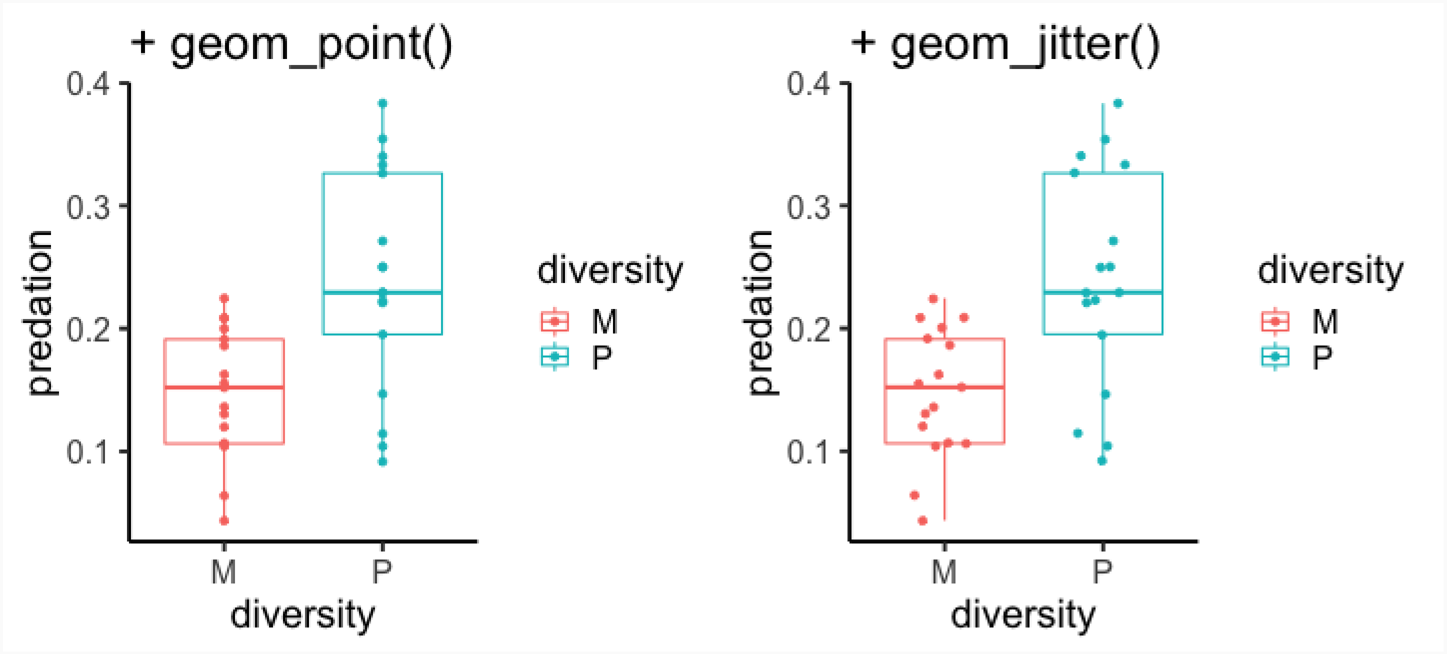


???
with comparisons we often want to show the range of values, error, not just means

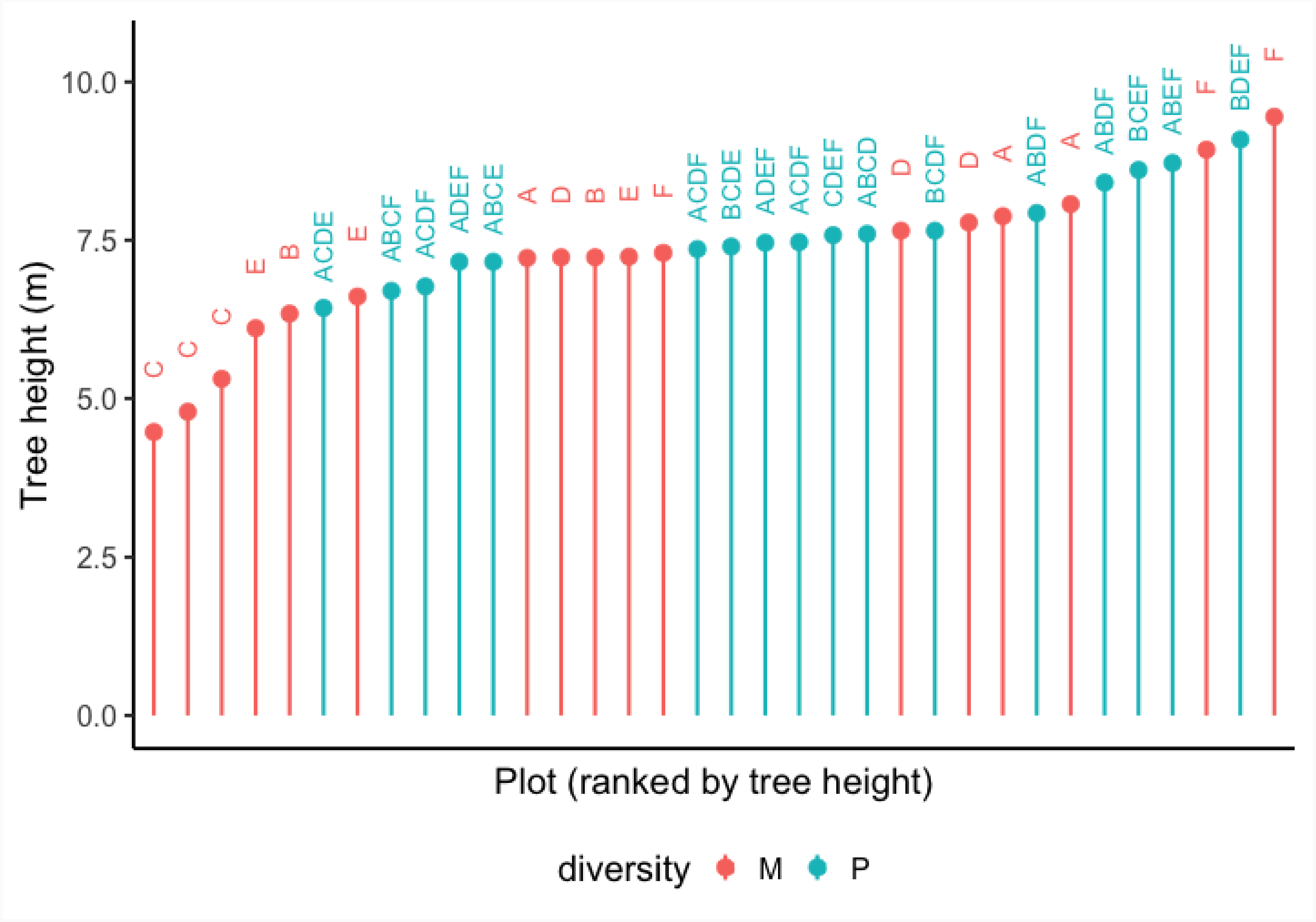
+ geom_point()

Use geom_point() to show raw data over boxplots
include `position = position_jitter(.1)` to spread overlapping points

```
plot.box<-ggplot(birds, aes(y = predation, x = diversity, color = diversity))+
  geom_boxplot()

plot.box + geom_point()
plot.box + geom_jitter(width=0.1)
```

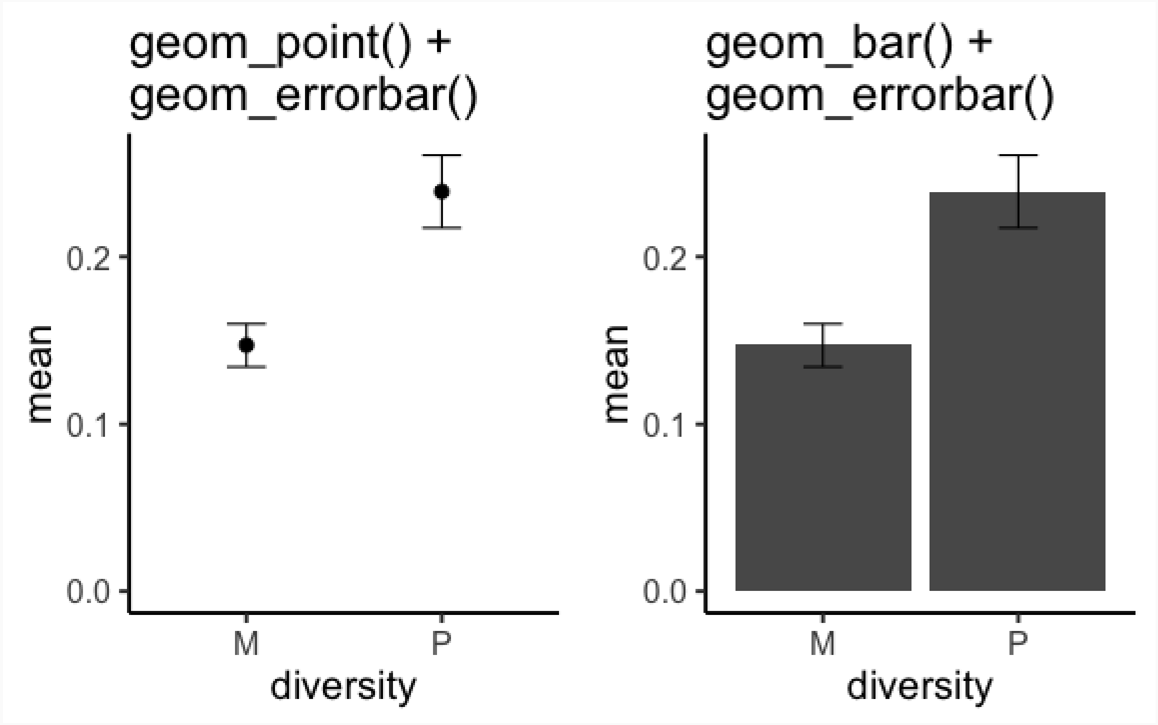


lollipop chart



comparisons: means with error bars

means: `geom_bar()`, `geom_point()`
error: `geom_errorbar()`, `geom_linerange()`, `geom_pointrange()`
error bars require `ymin` and `ymax`



summarizing data with dplyr

se = standard error = standard deviation / sqrt of n

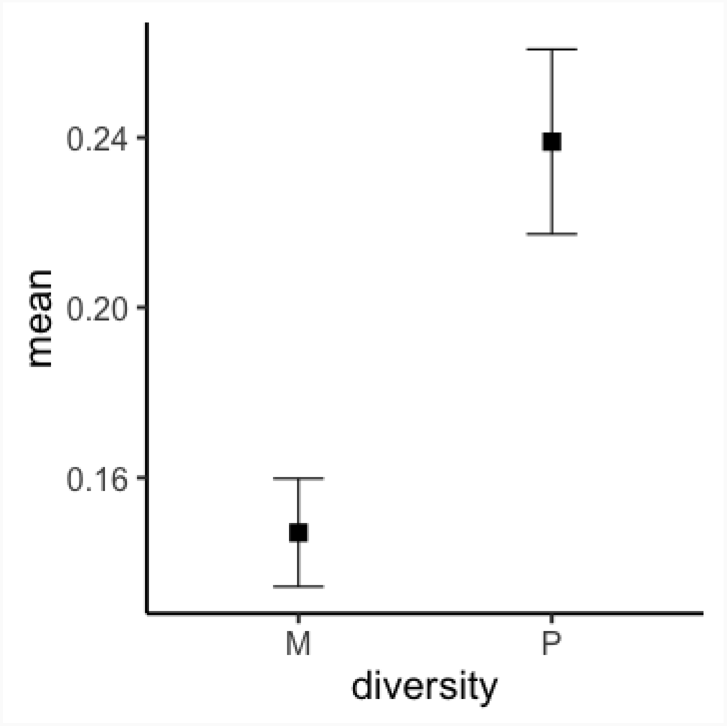
```r
#install.packages('dplyr')
library(dplyr)

# summarize statistics for predation
summary.df<-birds%>%
  group_by(diversity)%>%
  summarize(mean = mean(predation), n = length(predation), sd = sd(predation))%>%
  mutate(se = sd/sqrt(n))
```

```r
summary.df
```

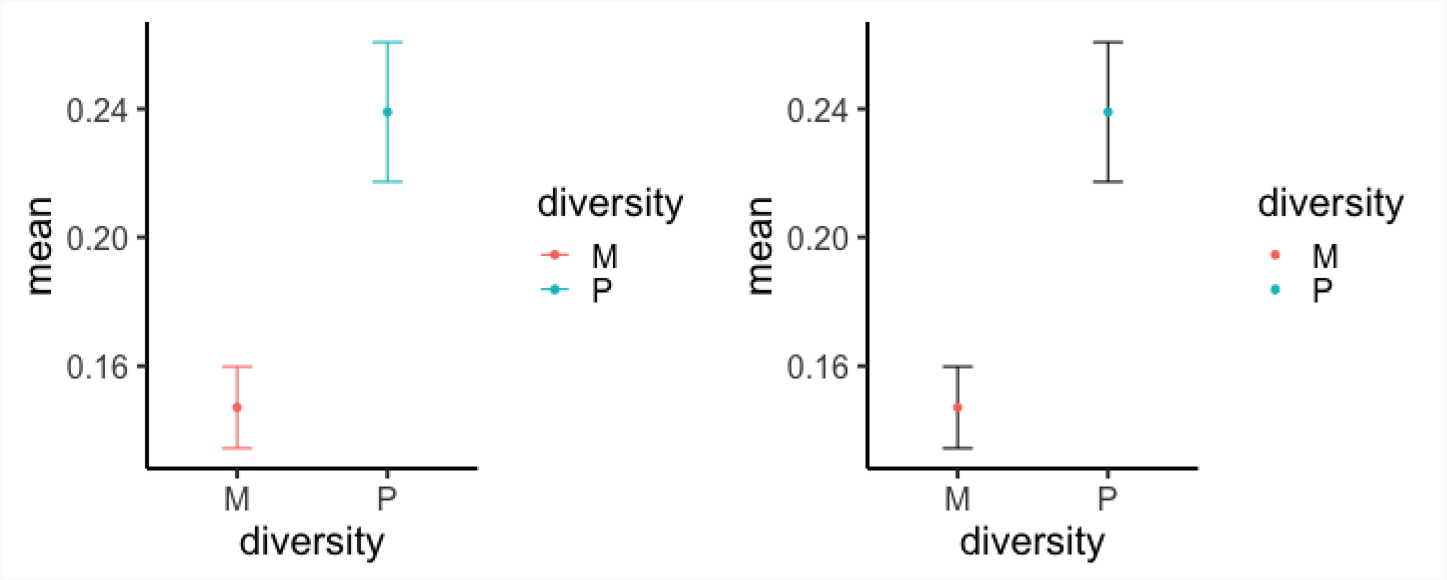
```
## # A tibble: 2 x 5
##   diversity  mean     n     sd     se
##   <fct>     <dbl> <int>  <dbl>  <dbl>
## 1 M         0.147    17 0.0525 0.0127
## 2 P         0.239    17 0.0898 0.0218
```

plot mean and error

Generate a plot showing mean values with some measure of error



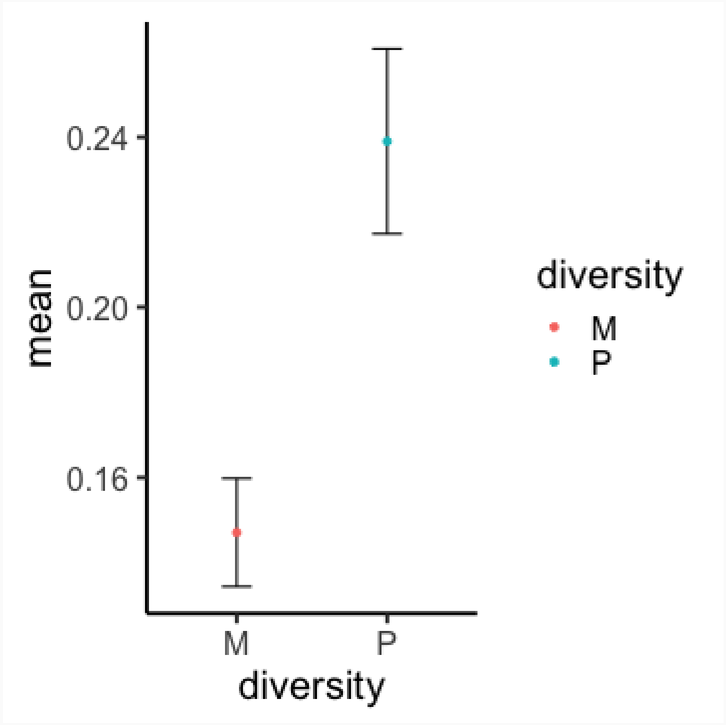
layering & aesthetics



layering multiple dataframes

Similarly, the dataframe can be assigned to specific geoms:

```
ggplot()+
  geom_errorbar(data=summary.df, aes(x = diversity, ymin = mean-se, ymax=mean+se), width=.2)+
  geom_point(data=summary.df, aes(x= diversity, y= mean, color=diversity))
```



tree-level data

```
trees<-read.csv('https://raw.githubusercontent.com/collnell/GWU-visual/master/tree_pred.csv')
head(trees)
```

```
##   diversity plot tree tree_n cater_n bird_bites predation
## 1         M    3    D     13      50          6 0.1200000
## 2         M    9    A     12      43          8 0.1860465
## 3         M   12    E     12      46          7 0.1521739
## 4         M   13    D     12      48          5 0.1041667
## 5         M   17    F     12      47          5 0.1063830
## 6         M   20    A     12      45          7 0.1555556
```

multiple grouping variables

`diversity` & `tree`

```
ggplot(trees, aes(diversity, predation))+
  geom_boxplot()+
  geom_jitter(aes(shape=tree))
```

aggregate by plot

```r
se<-function(x) sd(x, na.rm=TRUE)/sqrt(length(x))

plot.pred<-trees%>%
  group_by(diversity, tree)%>%
  summarize(mean=mean(predation), se=se(predation))
head(plot.pred)
```

```
## # A tibble: 6 x 4
## # Groups:   diversity [1]
##   diversity tree   mean      se
##   <fct>     <fct> <dbl>   <dbl>
## 1 M         A     0.159 0.0145
## 2 M         B     0.212 0.0122
## 3 M         C     0.203 0.00561
## 4 M         D     0.110 0.00495
## 5 M         E     0.109 0.0332
## 6 M         F     0.111 0.0287
```

plot means by tree

```r
# tree point
ggplot(plot.pred, aes(tree, mean, color=diversity))+
  geom_point()+
  geom_errorbar(aes(ymin=mean-se, ymax=mean+se), width=0.1)
```

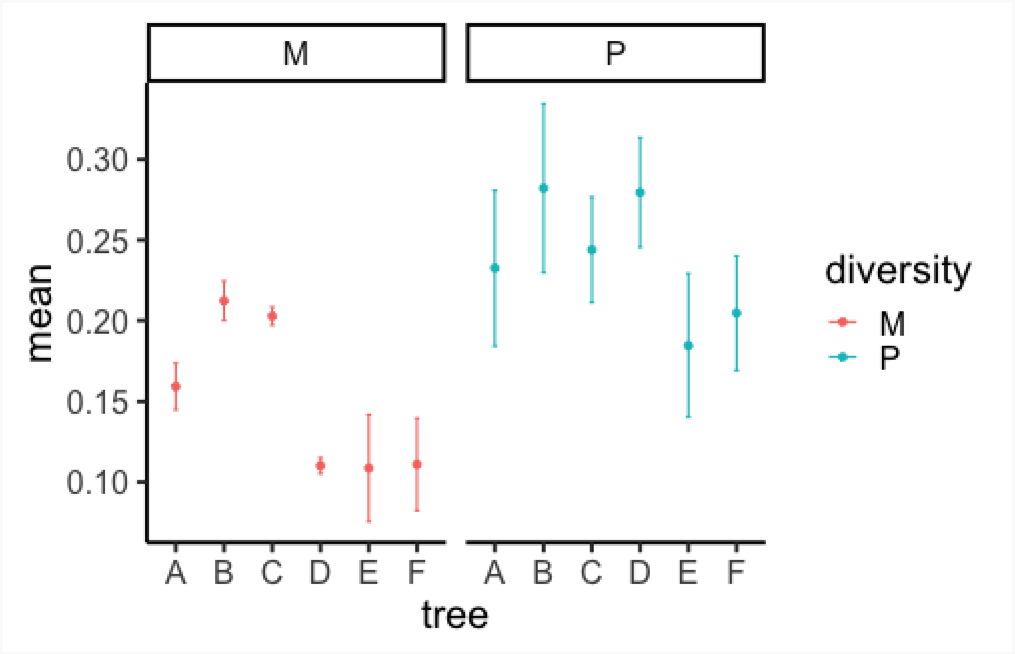
plot means by diversity

```r
# div point position dodge
ggplot(plot.pred, aes(diversity, mean, color=tree))+
  geom_point(position=position_dodge(1))+
  geom_errorbar(aes(ymin=mean-se, ymax=mean+se), width=0.1, position=position_dodge(1))

# div bar - position dodge
ggplot(plot.pred, aes(diversity, mean, fill=tree))+
  geom_bar(stat='identity', position=position_dodge(1))+
  geom_errorbar(aes(ymin=mean-se, ymax=mean+se), width=0.1, position=position_dodge(1))
```

facets

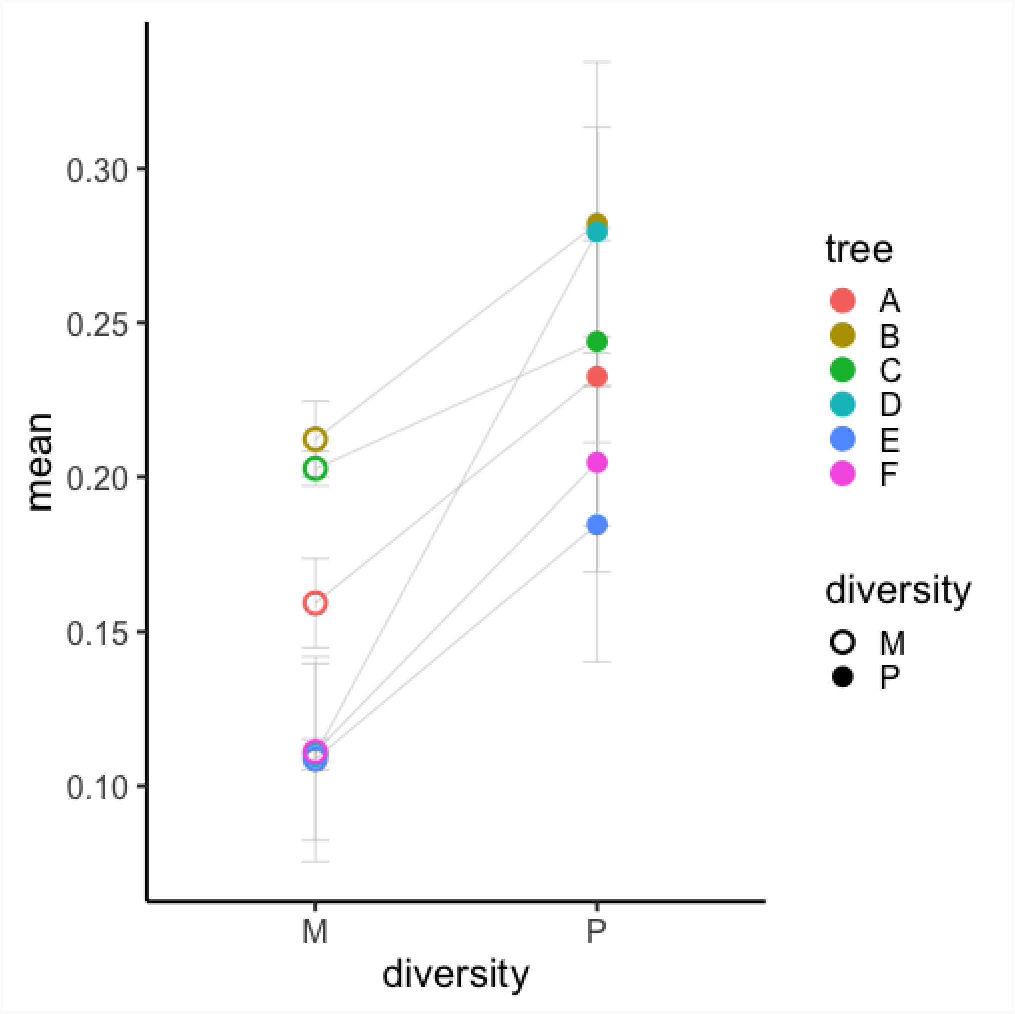
facets create small multiples of your plot based upon a categorical variable:

```
# facet by diversity
ggplot(plot.pred, aes(tree, mean, color=diversity))+
  geom_point(position=position_dodge(1))+
  geom_errorbar(aes(ymin=mean-se, ymax=mean+se), width=0.1, position=position_dodge(1))+
  facet_wrap(~diversity)
```

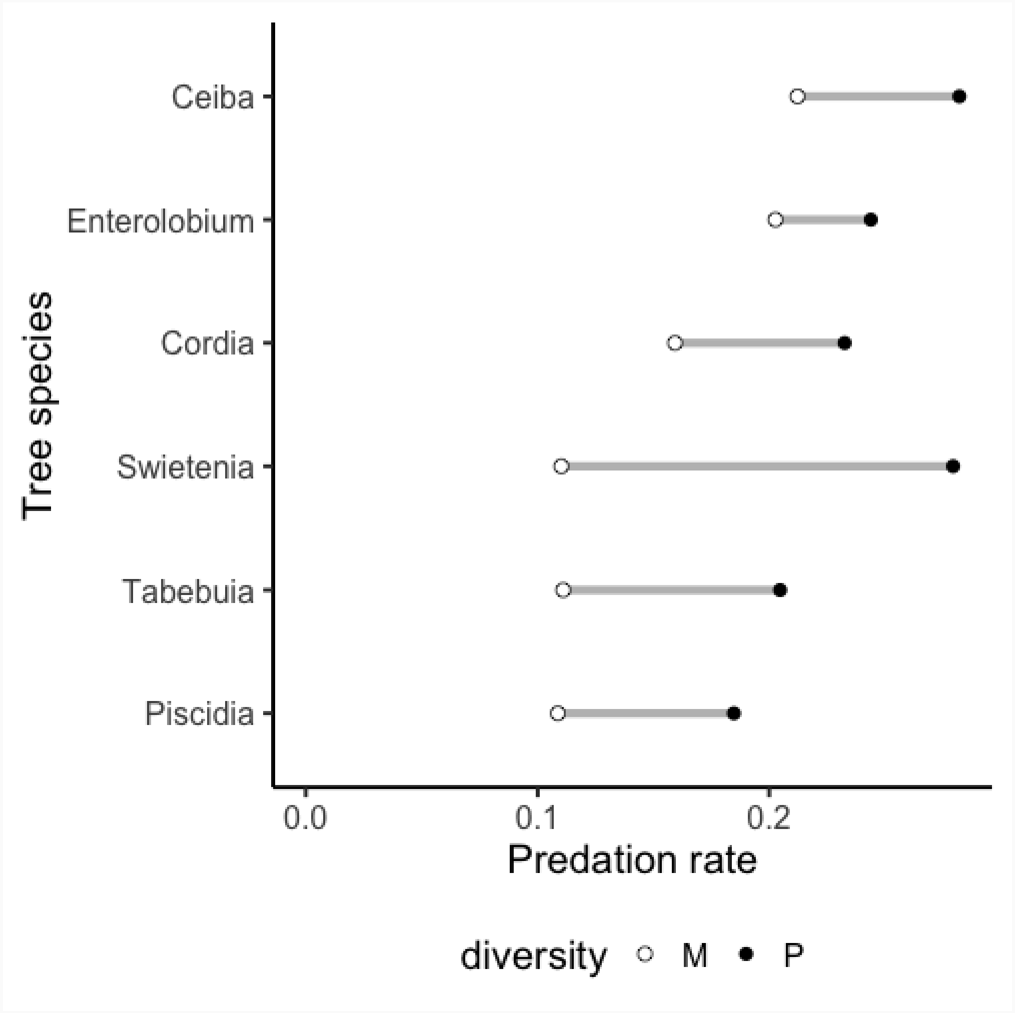


connect the dots

```
## Warning: Ignoring unknown parameters: stoke
```



+coord_flip()



Ceiba
Enterolobium
Cordia
Swietenia
Tabebuia
Piscidia
Tree species
0.0
0.1
0.2
Predation rate
diversity
○ M
● P

questions??

next week

Send me any examples of what your are trying to plot, chart types, interests

Publication ready figures

- Tweaking theme to fit journal guidelines
- Custom themes & color palettes
- Saving – figure format, resolution
- Multi-panel figs, faceting
- Annotating plots – adding pvalues, rsquared, highlighting data
- Working with scales – color scales
- Ggpubr, cowplot, fortify, gganimate
- Reordering axes