Data Science for Everyone – Grammar of Graphics

Dr. Ab Mosca (they/them)

Slides based off slides courtesy of Jordan Crouser (https://jcrouser.github.io/)

Plan for Today

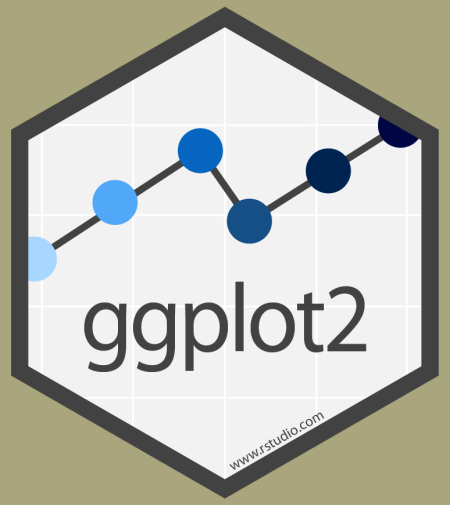
- Connect what we know about visualizations to ggplot in R

`ggplot2`

- Library for creating plots in R
- The "gg" stand for **g**rammar of **g**raphics

Big idea behind a grammar of graphics:

- Independently specify plot building blocks and combine them to create graphical displays



ggplot2

- Plot building blocks
  - data
  - aesthetic mappings (how we draw that stuff)
  - geometric objects (the literal stuff we draw)
  - statistical transformations (underlying model)
  - scales (range of values, colors, etc.)
  - faceting (small multiples)

ggplot2

- Plot building blocks
  - data
  - aesthetic mappings (how we draw that stuff)
  - geometric objects (the literal stuff we draw)
  - statistical transformations (underlying model)
  - scales (range of values, colors, etc.)
  - faceting (small multiples)

ggplot2

- Plot building blocks
  - data
  - aesthetic mappings (how we draw that stuff)
  - geometric objects (the literal stuff we draw)
  - statistical transformations (underlying model)
  - scales (range of values, colors, etc.)
  - faceting (small multiples)

```
ggplot(data, aes()) +
   geom_*
```

data

aesthetic mapping

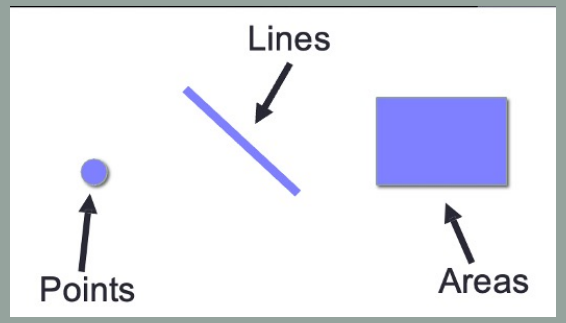
geometric object

ggplot2

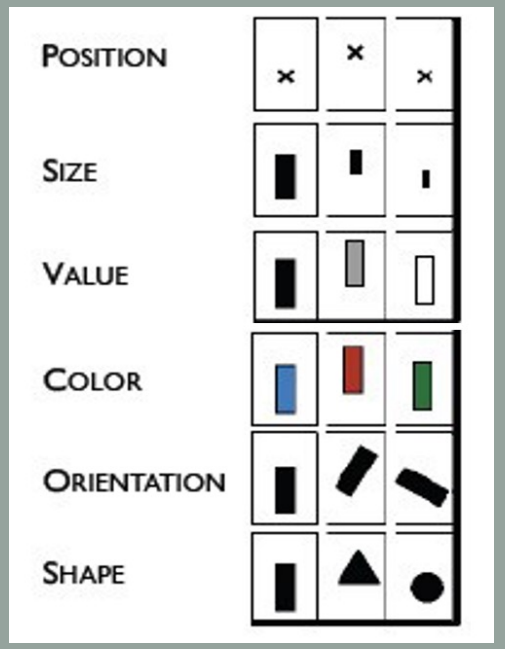
- Plot building blocks
  - data
  - aesthetic mappings (how we draw that stuff)
  - geometric objects (the literal stuff we draw)

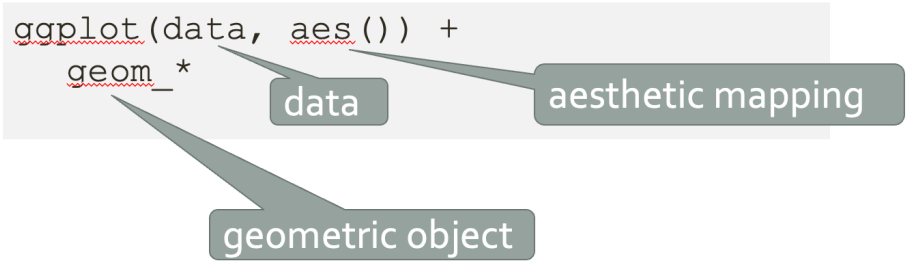
Sound Familiar?

Marks



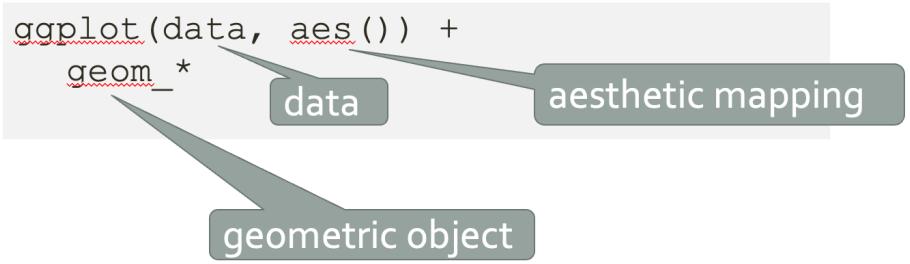
Channels

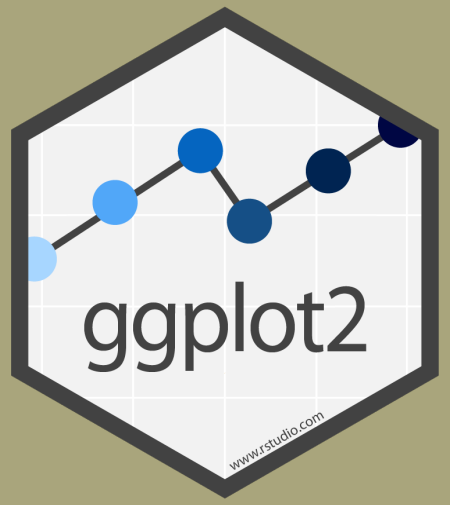




Data

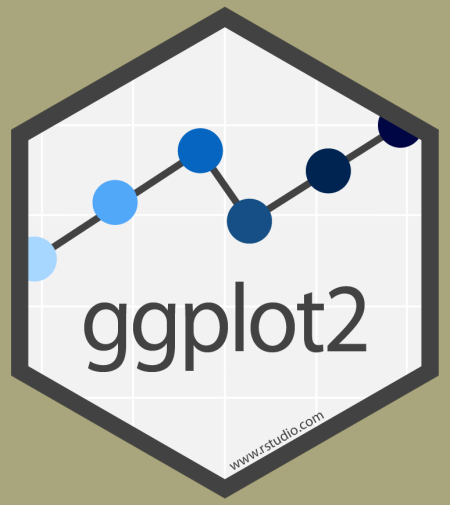
- First argument to `ggplot` is the data you want to plot
- We will use the iris dataset
- Open a new R Markdown and take a glimpse at the iris dataset





Data

- First argument to `ggplot` is the data you want to plot
- We will use the iris dataset
- Open a new R Markdown and take a glimpse at the iris dataset
- Let's make a plot with this dataset




```
ggplot(data, aes()) +
  geom_*
```

data

aesthetic mapping

geometric object

Data

- First argument to `ggplot` is the data you want to plot

```
ggplot(iris)
```


```
ggplot(data, aes()) +
  geom_*
```

data

aesthetic mapping

geometric object

Aesthetic Mapping (aes())

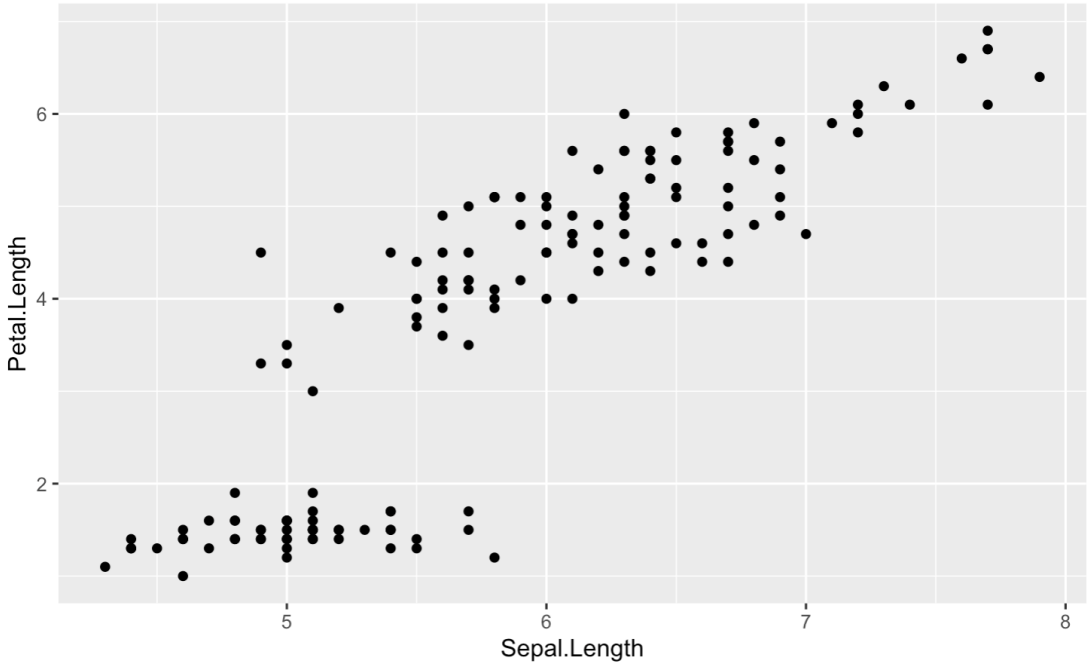
- Second argument is aes, the data to visual channels mapping

```
ggplot(iris, aes(x = Sepal.Length, y = Petal.Length))
```

Geometric Objects (geom)

- Last, add geom, which identifies the marks

```
ggplot(iris, aes(x = Sepal.Length, y = Petal.Length)) +
   geom_point()
```

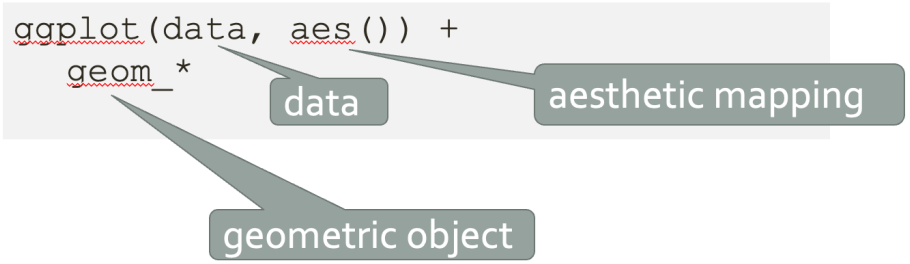



```
ggplot(data, aes()) +
   geom_*
```

data

aesthetic mapping

geometric object

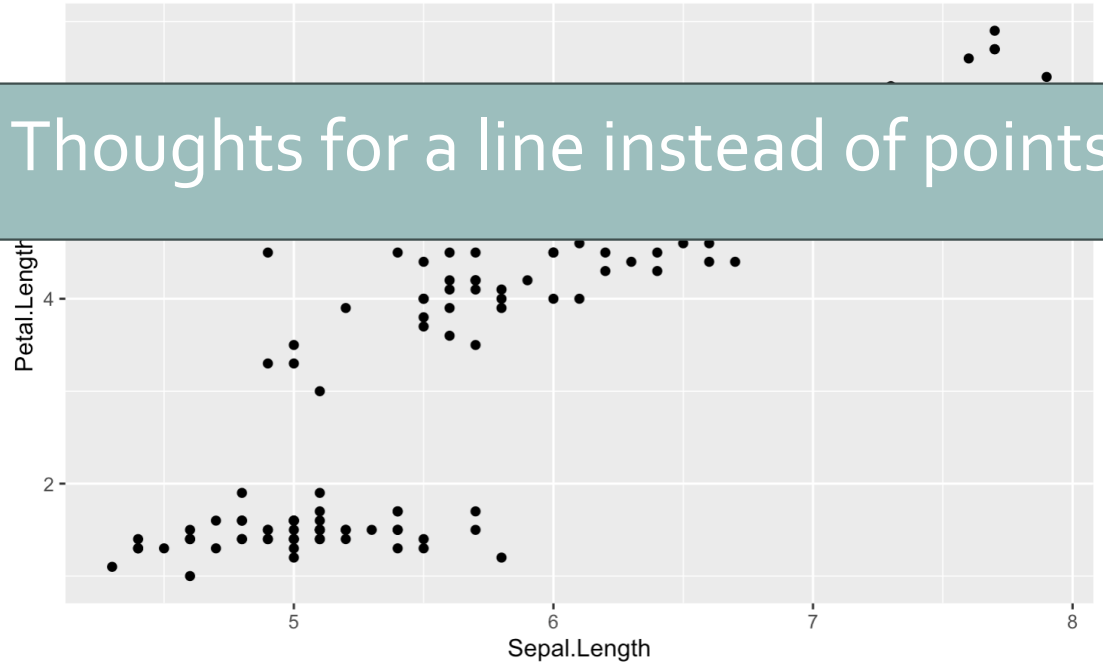


Geometric Objects (geom)

- Last, add geom, which identifies the marks

```
ggplot(iris, aes(x = Sepal.Length, y = Petal.Length)) +
   geom_point()
```

Thoughts for a line instead of points?



```
ggplot(data, aes()) +
  geom_*
```

data

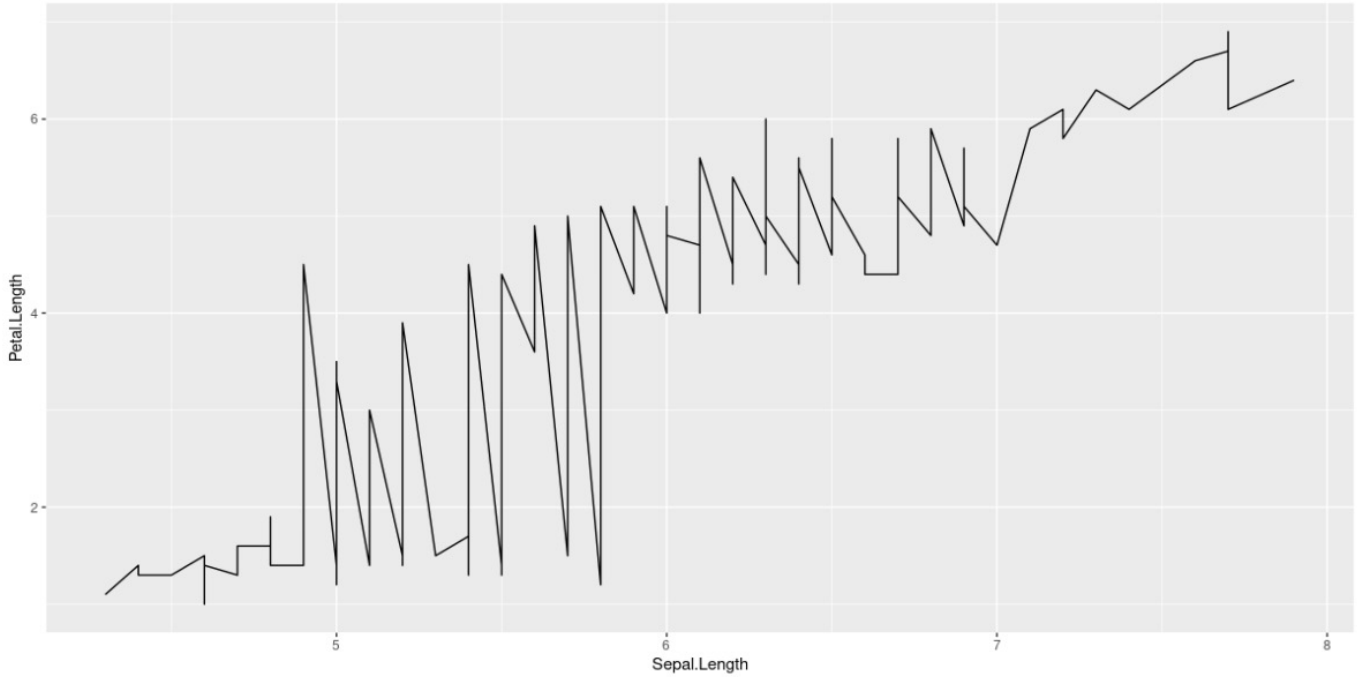
aesthetic mapping

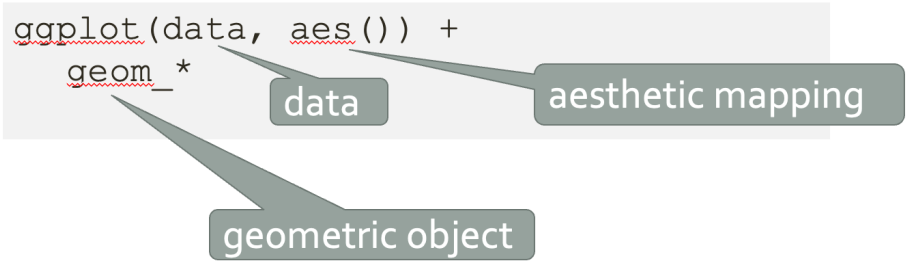
geometric object

Geometric Objects (geom)

- Last, add geom, which identifies the marks

```
ggplot(iris, aes(x = Sepal.Length, y = Petal.Length)) +
  geom_line()
```



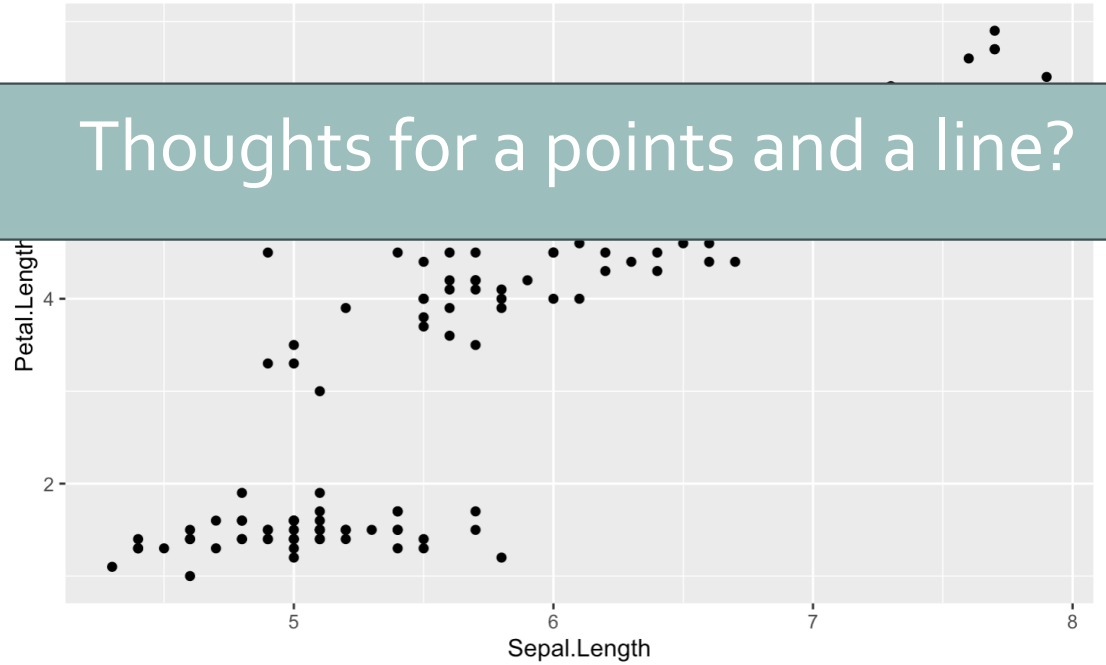


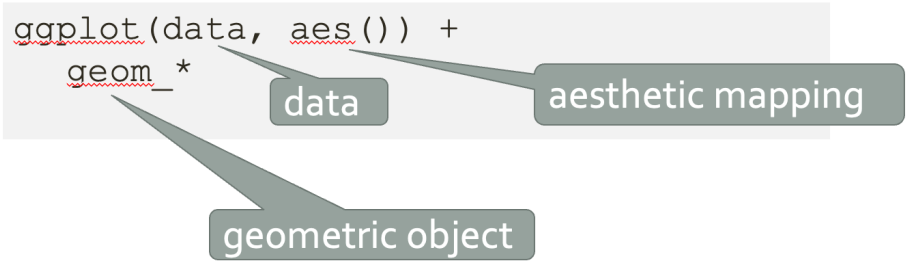
Geometric Objects (geom)

- Last, add geom, which identifies the marks

```
ggplot(iris, aes(x = Sepal.Length, y = Petal.Length)) +
   geom_point()
```

Thoughts for a points and a line?

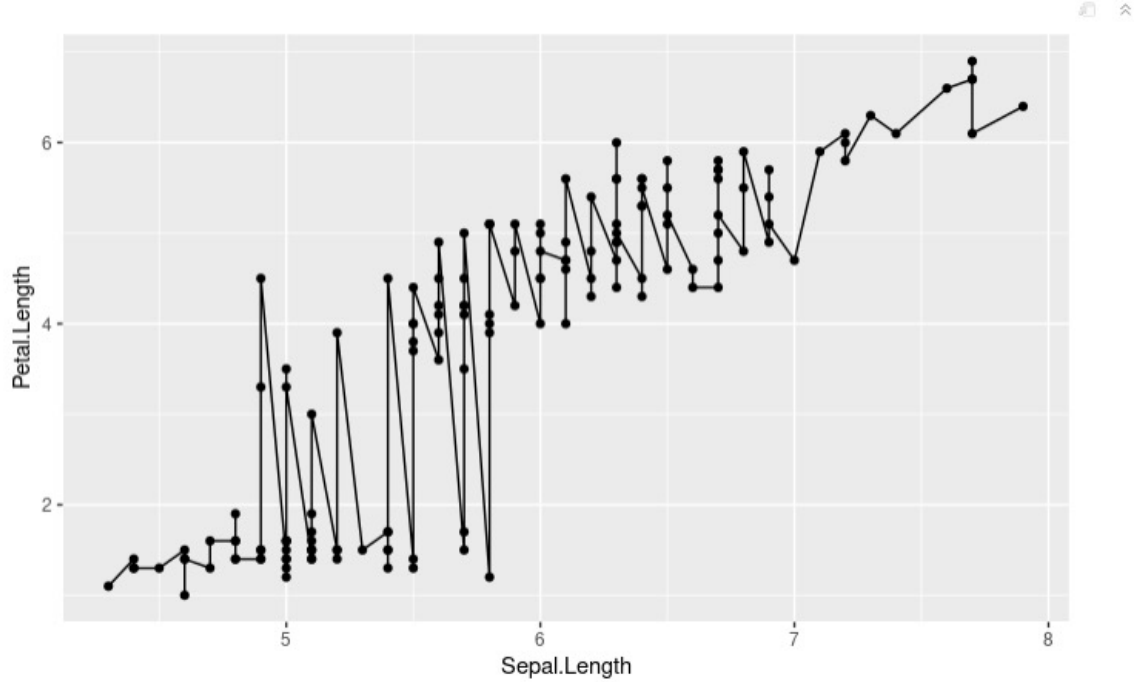




Geometric Objects (geom)

- Last, add geom, which identifies the marks

```
ggplot(iris, aes(x = Sepal.Length, y = Petal.Length)) +
   geom_point() + geom_line()
```



```
ggplot(data, aes()) +
  geom_*
```

data

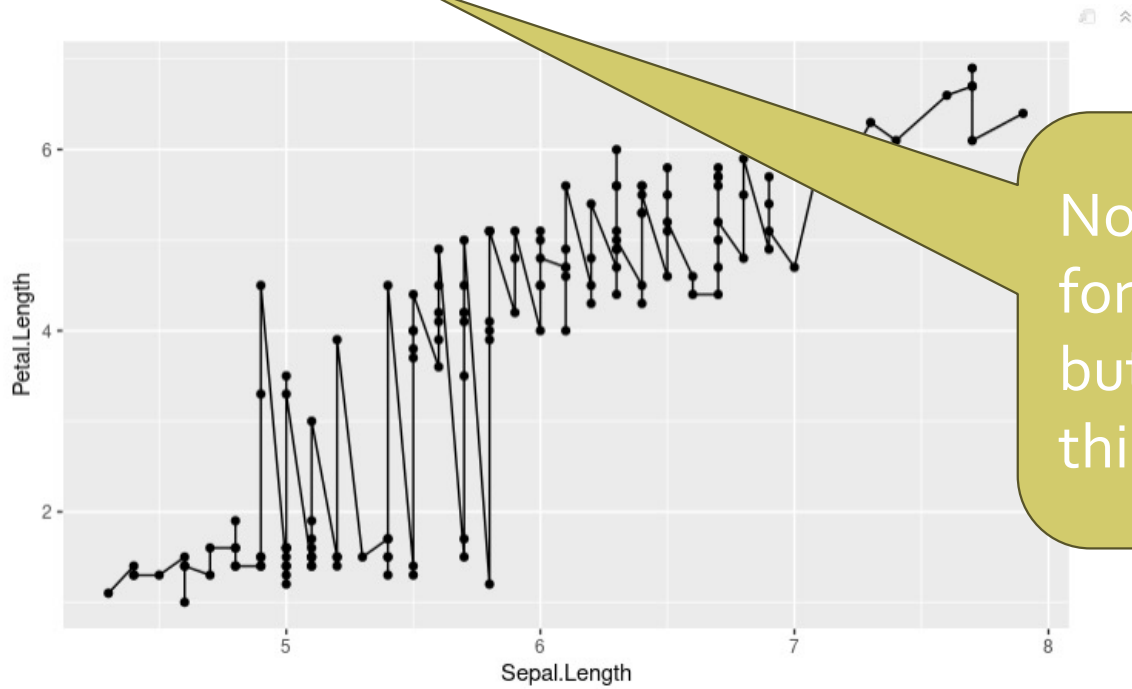
aesthetic mapping

geometric object

Geometric Objects (geom)

- Last, add geom, which identifies the marks

```
ggplot(iris, aes(x = Sepal.Length, y = Petal.Length)) +
   geom_point() + geom_line()
```



No upper limit for geometries, but do make things readable!

```
ggplot(data, aes()) +
  geom_*
```

data

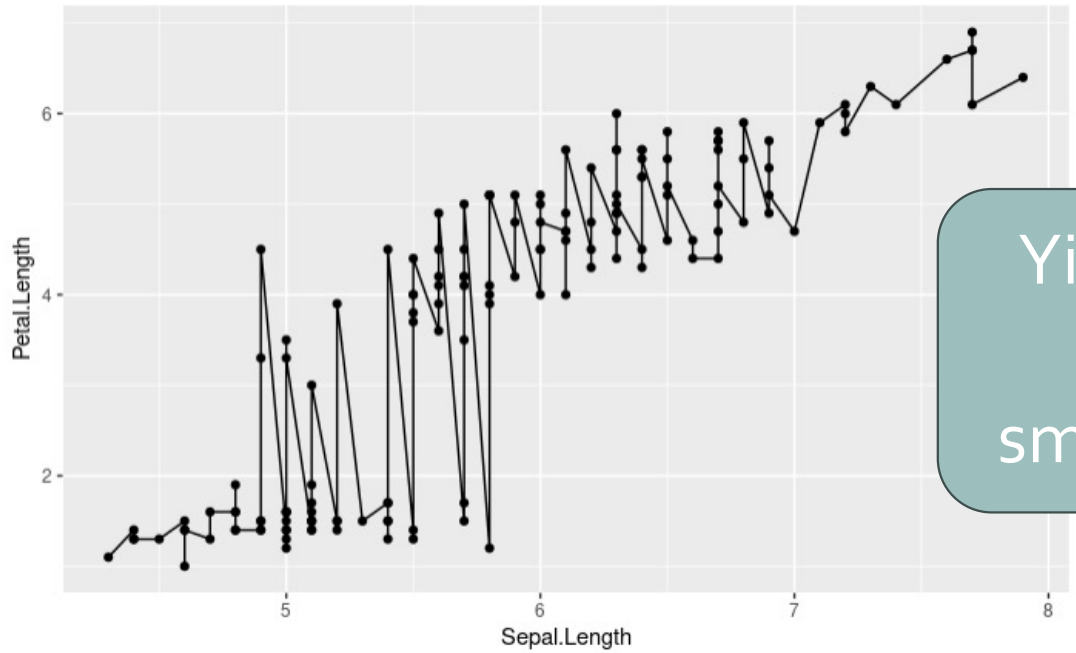
aesthetic mapping

geometric object

Geometric Objects (geom)

- Last, add geom, which identifies the marks

```
ggplot(iris, aes(x = Sepal.Length, y = Petal.Length)) +
  geom_point() + geom_line()
```



Yikes, what about a smooth line?

```
ggplot(data, aes()) +
  geom_*
```

data

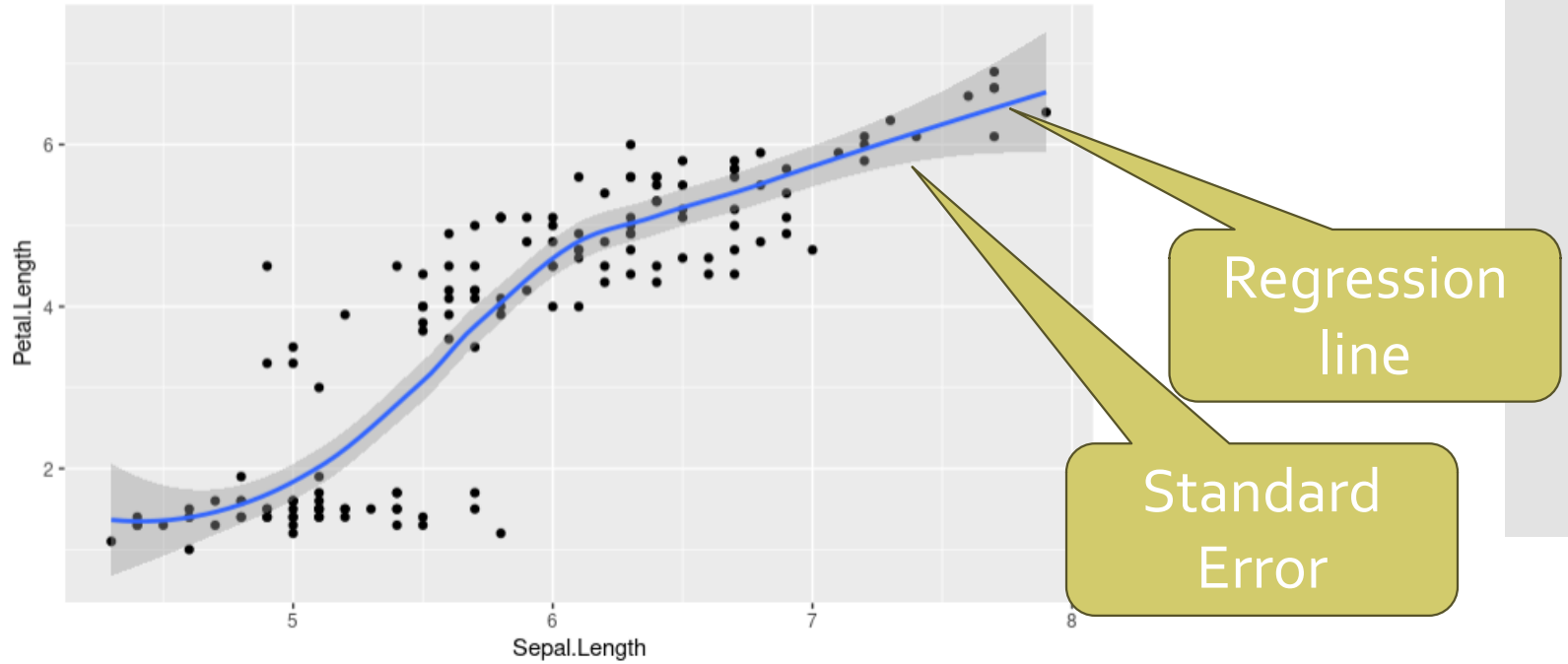
aesthetic mapping

geometric object

Geometric Objects (geom)

- Last, add geom, which identifies the marks

```
ggplot(iris, aes(x = Sepal.Length, y = Petal.Length)) +
   geom_point() + geom_smooth()
```



```
ggplot(data, aes()) +
  geom_*
```

data

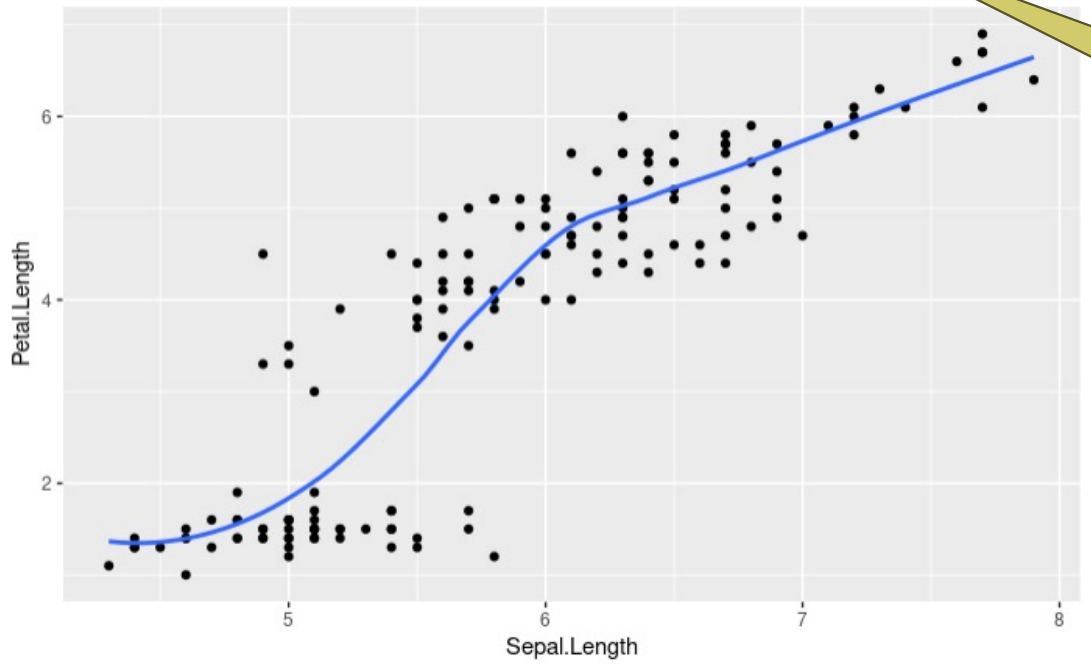
aesthetic mapping

geometric object

Geometric Objects (geom)

- Last, add geom, which identifies the marks

```
ggplot(iris, aes(x = Sepal.Length, y = Petal.Length)) +
  geom_point() + geom_smooth(se = FALSE)
```



to remove error bands

```
ggplot(data, aes()) +
  geom_*
```

data

aesthetic mapping

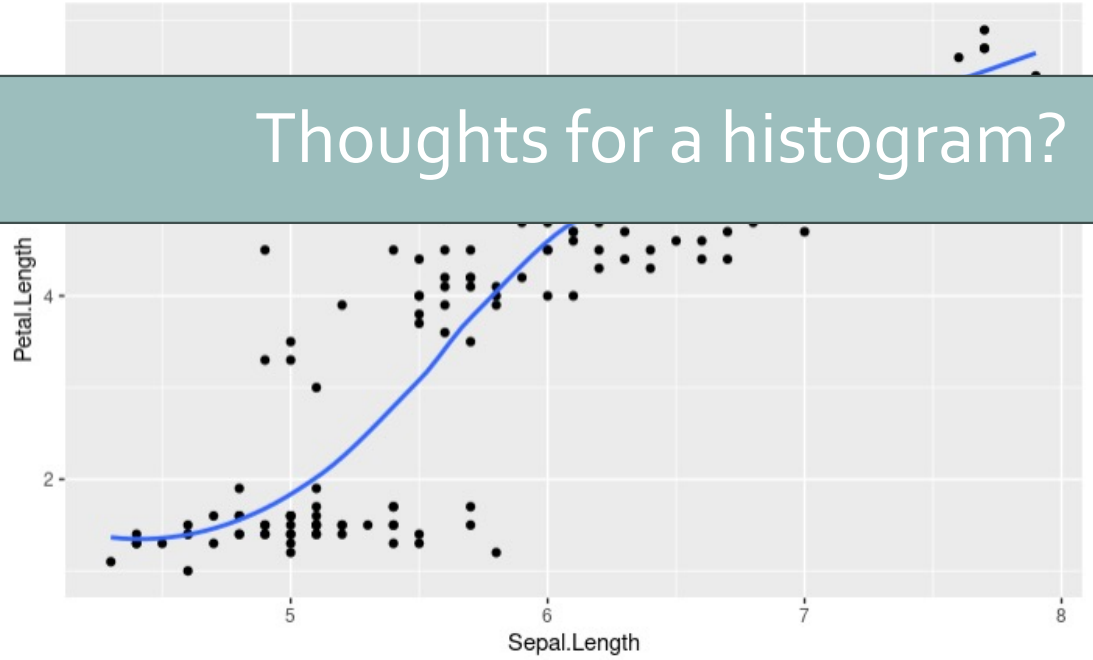
geometric object

Geometric Objects (geom)

- Last, add geom, which identifies the marks

```
ggplot(iris, aes(x = Sepal.Length, y = Petal.Length)) +
  geom_point() + geom_smooth(se = FALSE)
```

Thoughts for a histogram?



```
ggplot(data, aes()) +
  geom_*
```

data

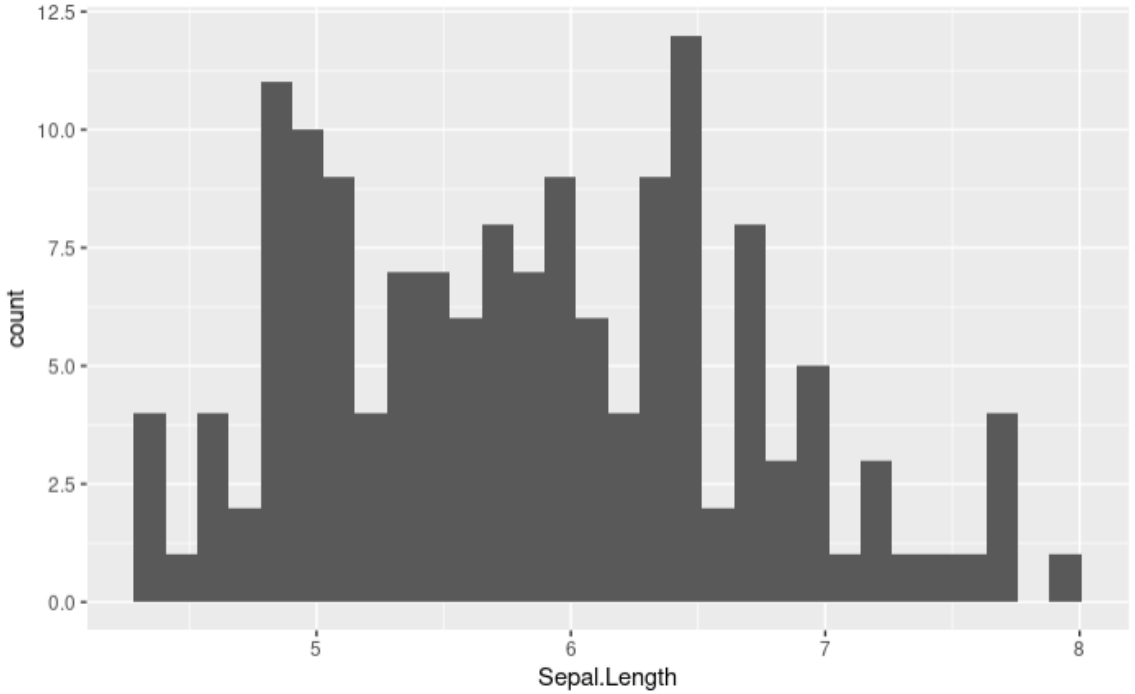
aesthetic mapping

geometric object

Geometric Objects (geom)

- Last, add geom, which identifies the marks

```
ggplot(iris, aes(x = Sepal.Length) +
   geom_histogram()
```



```
ggplot(data, aes()) +
  geom_*
```

data

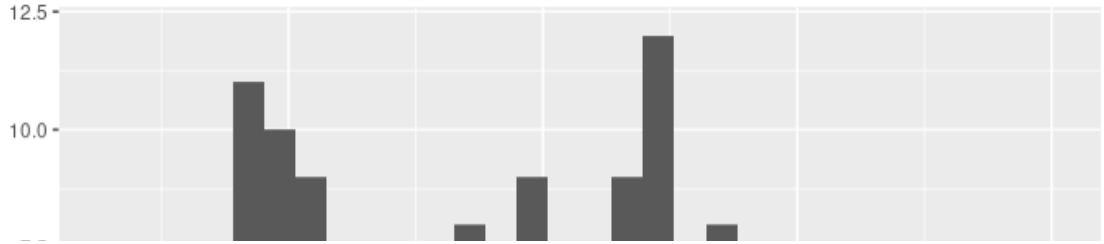
aesthetic mapping

geometric object

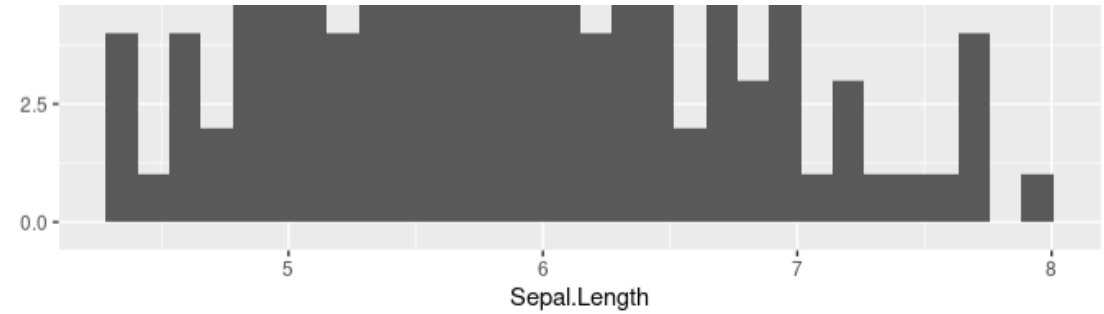
Geometric Objects (geom)

- Last, add geom, which identifies the marks

```
ggplot(iris, aes(x = Sepal.Length) +
   geom_histogram()
```




```
[38;5;232m`stat_bin()` using `bins = 30`. Pick better value with `binwidth`. [39m
```



```
ggplot(data, aes()) +
  geom_*
```

data

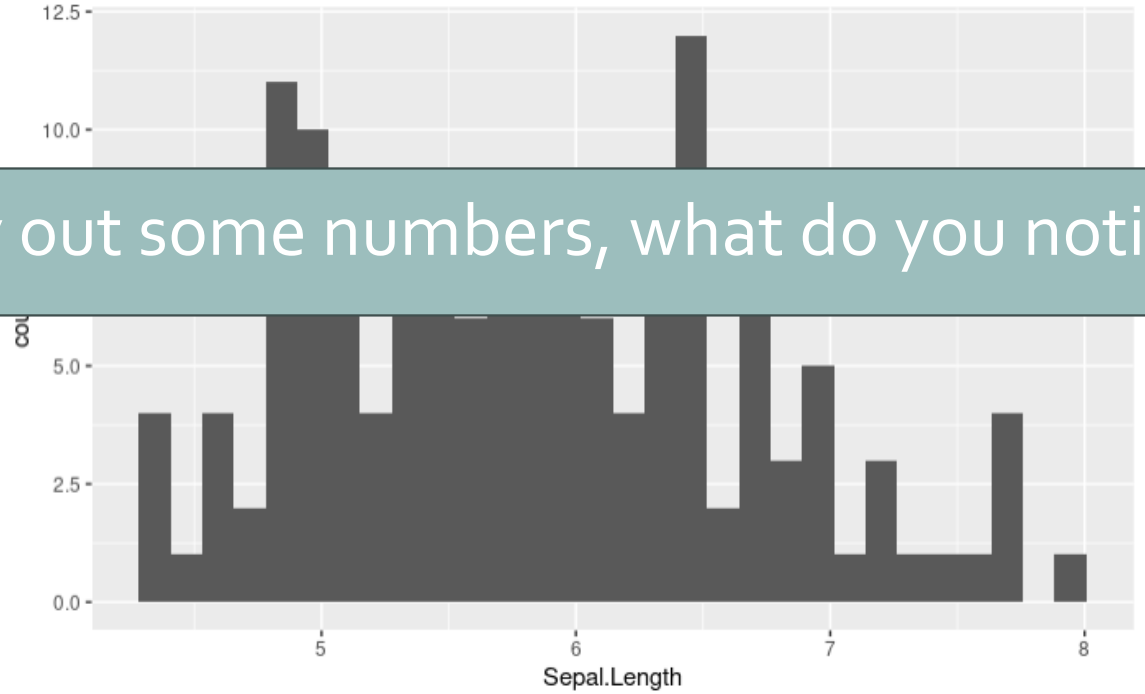
aesthetic mapping

geometric object

Geometric Objects (geom)

- Last, add geom, which identifies the marks

```
ggplot(iris, aes(x = Sepal.Length) +
   geom_histogram(binwidth = )
```



Try out some numbers, what do you notice?

Themes

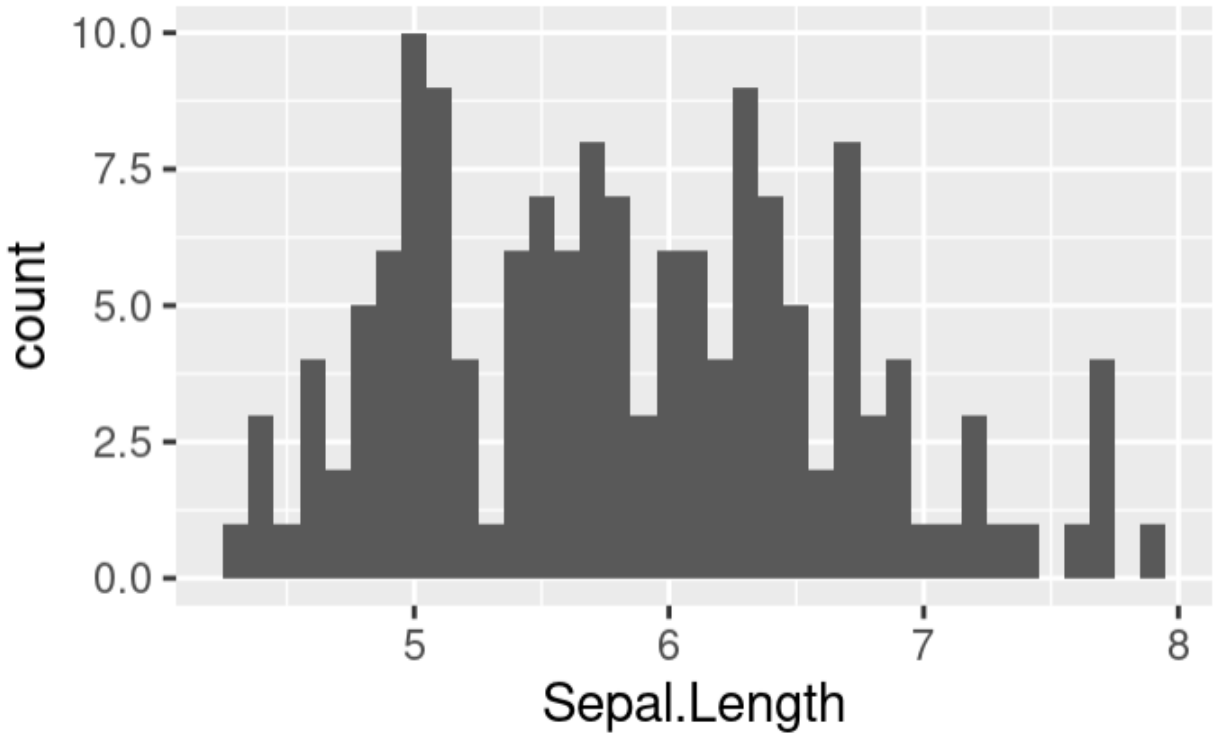
```
ggplot(data, aes()) +
  geom_*
```

data

aesthetic mapping

geometric object

```
ggplot(iris, aes(x = Sepal.Length) +
   geom_histogram(binwidth = 0.1) +
   theme_gray(base_size = 22)
```



Themes

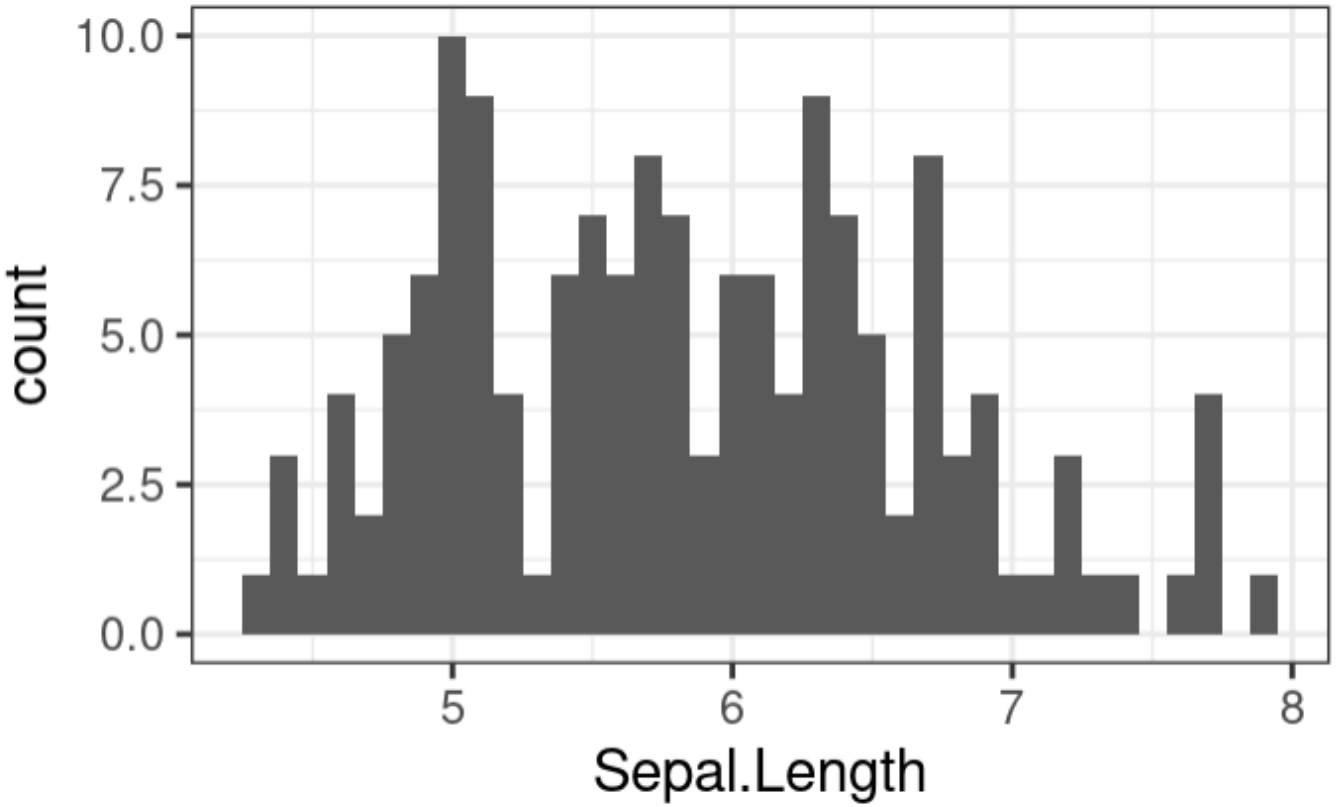
```
ggplot(data, aes()) +
   geom_*
```

data

aesthetic mapping

geometric object

```
ggplot(iris, aes(x = Sepal.Length) +
   geom_histogram(binwidth = 0.1) +
   theme_bw(base_size = 22)
```



```
ggplot(data, aes()) +
  geom_*
```

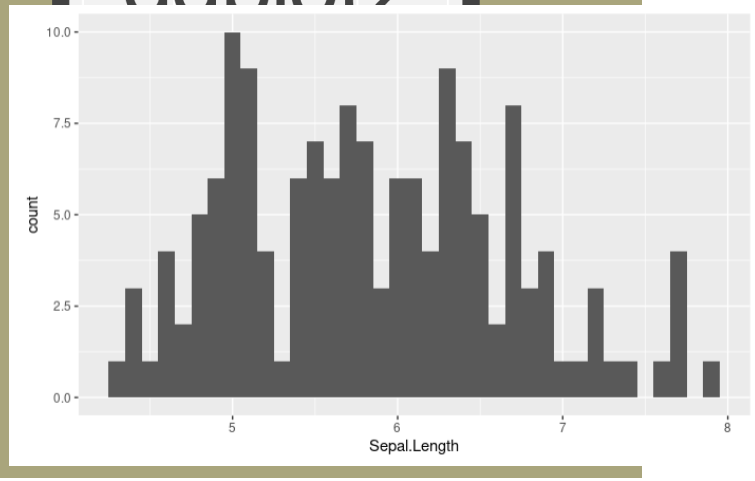
data

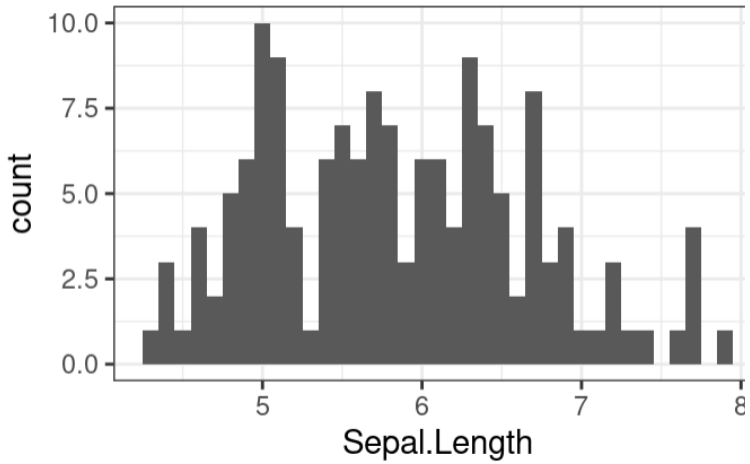
aesthetic mapping

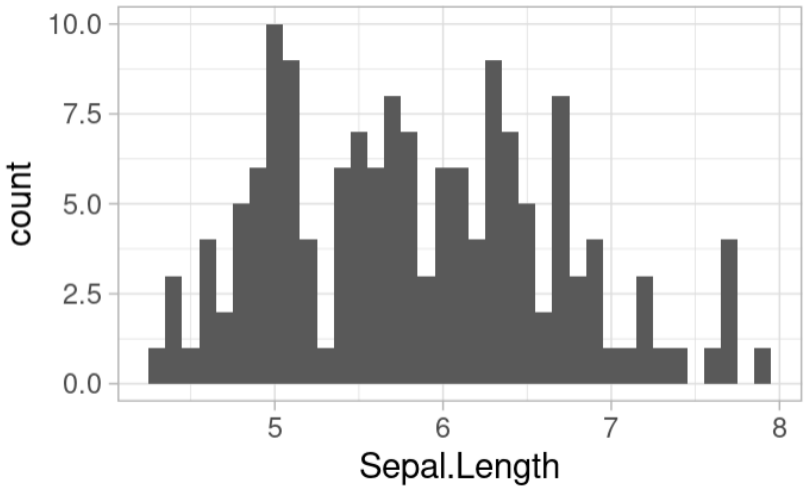
geometric object

Plots as Objects

```
plt <- ggplot(iris, aes(x = Sepal.Length) +
        geom_histogram(binwidth = 0.1)
plt_bw <- plt + theme_bw(base_size = 22)
plt_bw <- plt + theme_bw(base_size = 22)
```








```
ggplot(data, aes()) +
  geom_*
```

data

aesthetic mapping

geometric object

Your Turn!

- Upload the drinking-water.csv on the course website (under Labs) to your R Studio workspace

- Use ggplot to make a scatterplot of "# of Residential Connections" versus "Winter / Year Round Population Served"
  - Put "# of Residential Connections" on the x-axis
  - Hint: To refer to a variable with spaces in the name use ` around the name. Ex. x = ` "# of Residential Connections`

- What do you notice in the scatterplot?