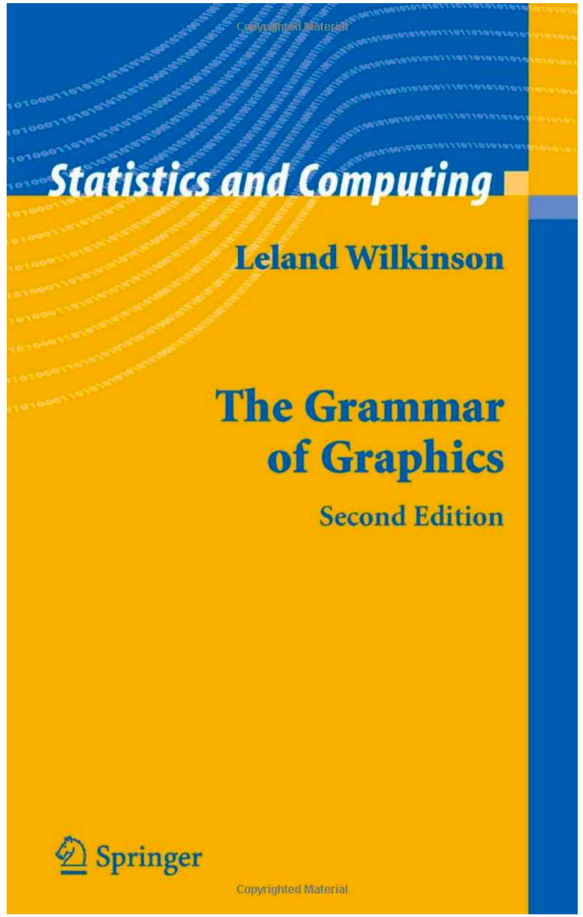
Grammar of Graphics

Overview, Layers

`slides/02_datalayer1.pdf`

Leland Wilkinson



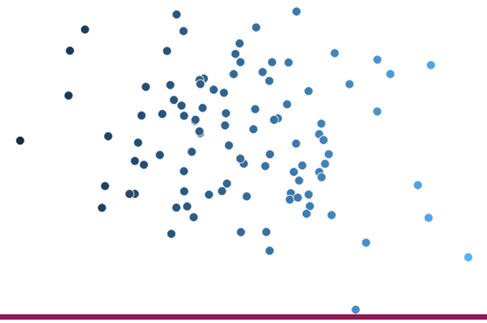
1944-2021

Grammar of graphics

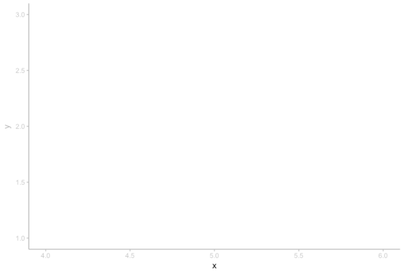
- presents a theory not a specific language / software
- takes us from "limited set of charts" to "an almost unlimited world of graphical forms"
- based on object oriented design: modular, reusuable
- other implementations exists besides ggplot2
- we will focus on the language/syntax of the ggplot2 implementation which differs slightly from the book

Building blocks

Layer(s)



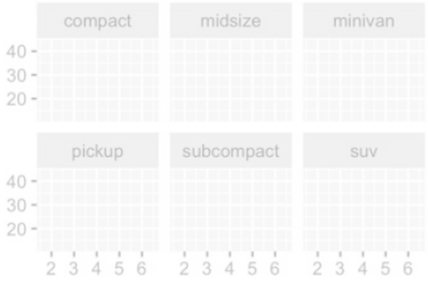
Scale(s)



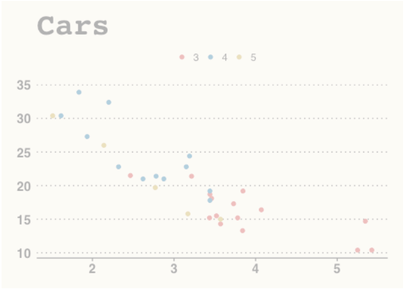
Coord

For now we will focus only on layers.

Facet

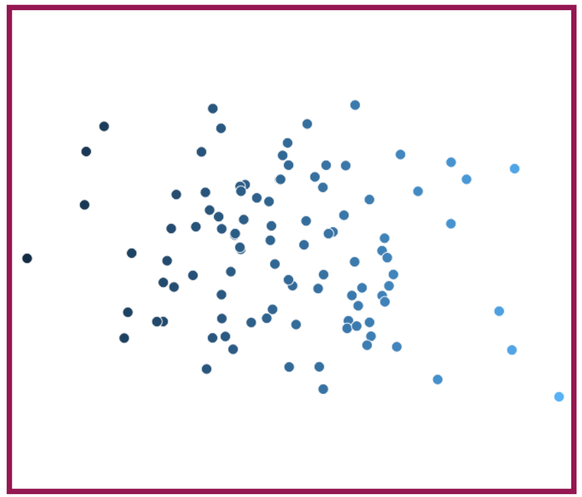


Theme



Layers

Each layer consists of:



1. GEOM
2. AESTHETIC MAPPING
3. DATA
4. STAT
5. POSITION

Layers

1. GEOM

point
bar
col
boxplot
line
histogram
density

geometric object

2. AESTHETIC MAPPING

x
y
color
fill
group
xmin
xmax
etc.

visual properties

→

variables

3. DATA

A	B	C

data frame

4. STAT

bin
boxplot
identity
density

statistical transformation

5. POSITION

identity
jitter
dodge
stack

shift

Layers

1. GEOM

point
bar
col
boxplot
line
histogram
density

2. AESTHETIC MAPPING

x
y
color
fill
group
xmin
xmax
etc.

3. DATA

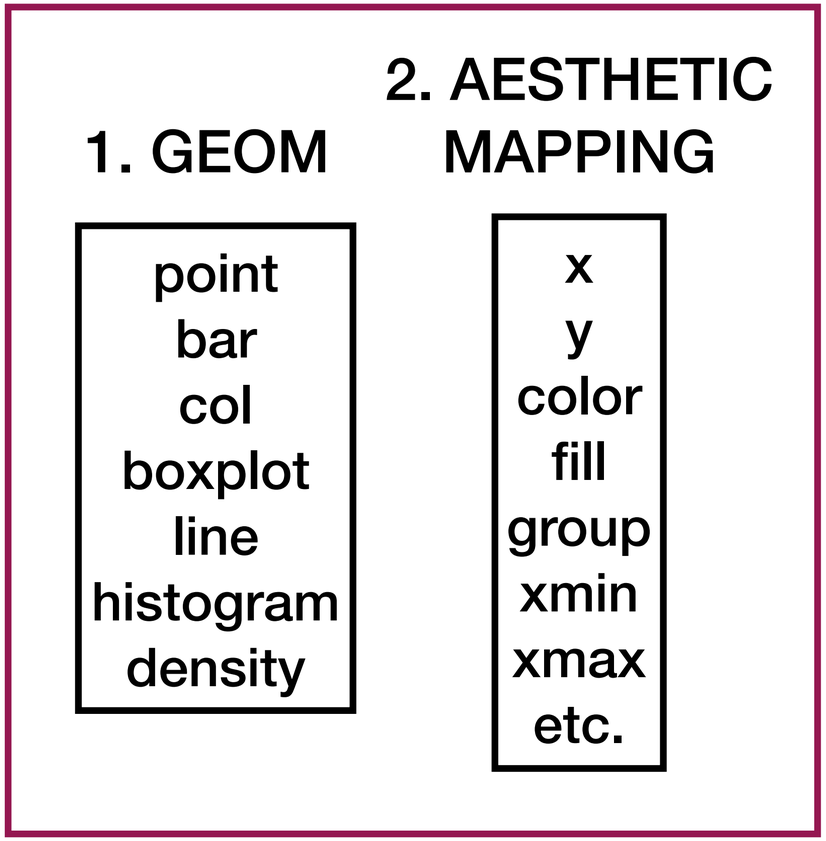
A	B	C

required

Most of the time you can use the default settings
for stat and position

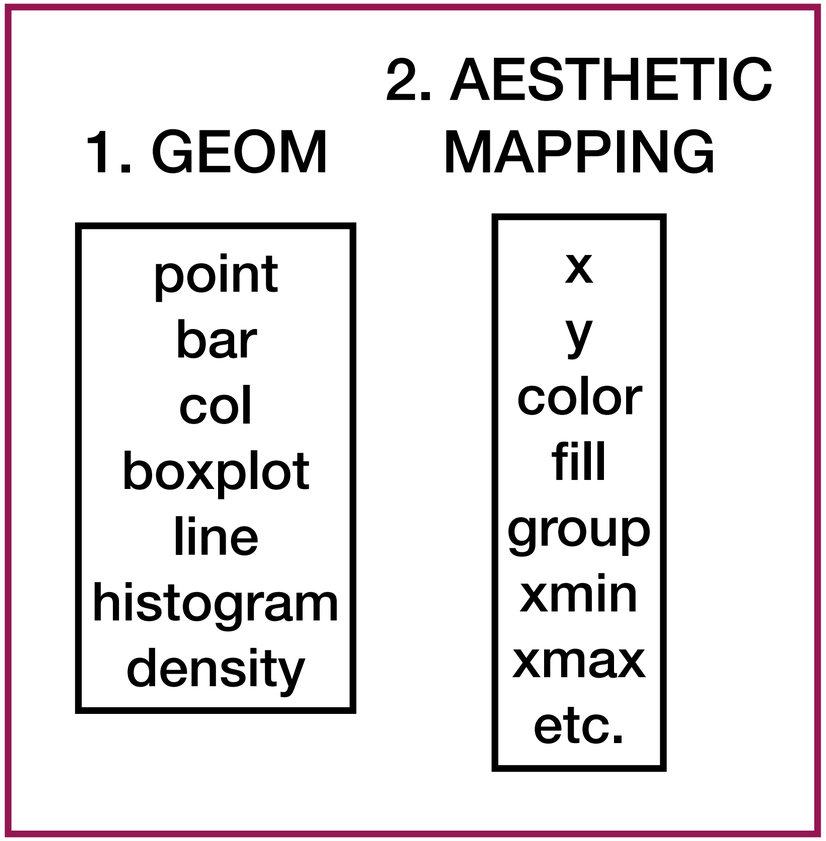
GEOMs and mappings

- Think about a plot as a collection of GEOMs
- Each GEOM has required mappings
- For example `geom_histogram()` requires **x** (or **y**)
- Required mappings are sometimes indicated in bold in the help files (though not on the **posit** cheatsheet)



GEOMs and mappings

- Sometimes the mapping must be **continuous** or **discrete**, sometimes it can be either
- **continuous** = `numeric`
- **discrete** = `factor, character`
- Many mistakes are caused by data in the wrong form, for example, numeric classified as character data



Continuous vs. Discrete

- Continuous = any *numeric* type:

  `numeric, double, integer (num, dbl, int)`

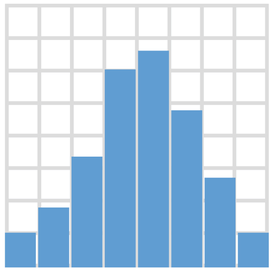
- Discrete = any *non-numeric* type:

  `factor, character, logical (fct, chr, lgl)`

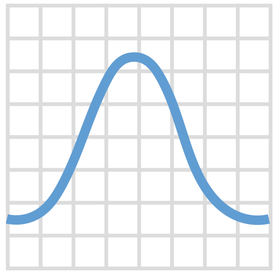
Continuous data / one mapping

```
geom_histogram()
geom_density()
```

GEOMS for continuous data, one mapping

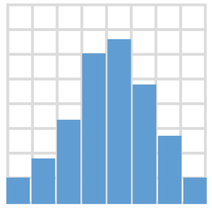


`geom_histogram()`

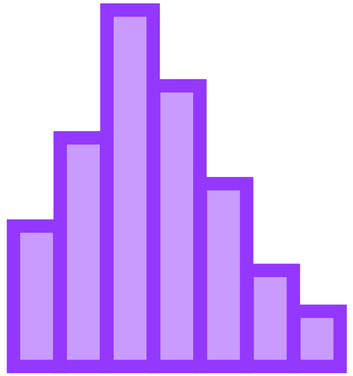


`geom_density()`

`geom_histogram()`

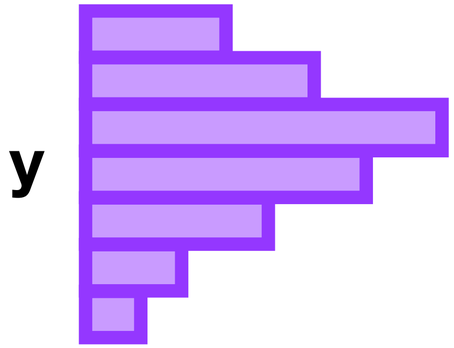


- Shows the distribution of a continuous variable (unbinned = no count column)
- Requires an **x** (vertical bars) *or* **y** (horizontal bars, rare)
- No spaces between bars



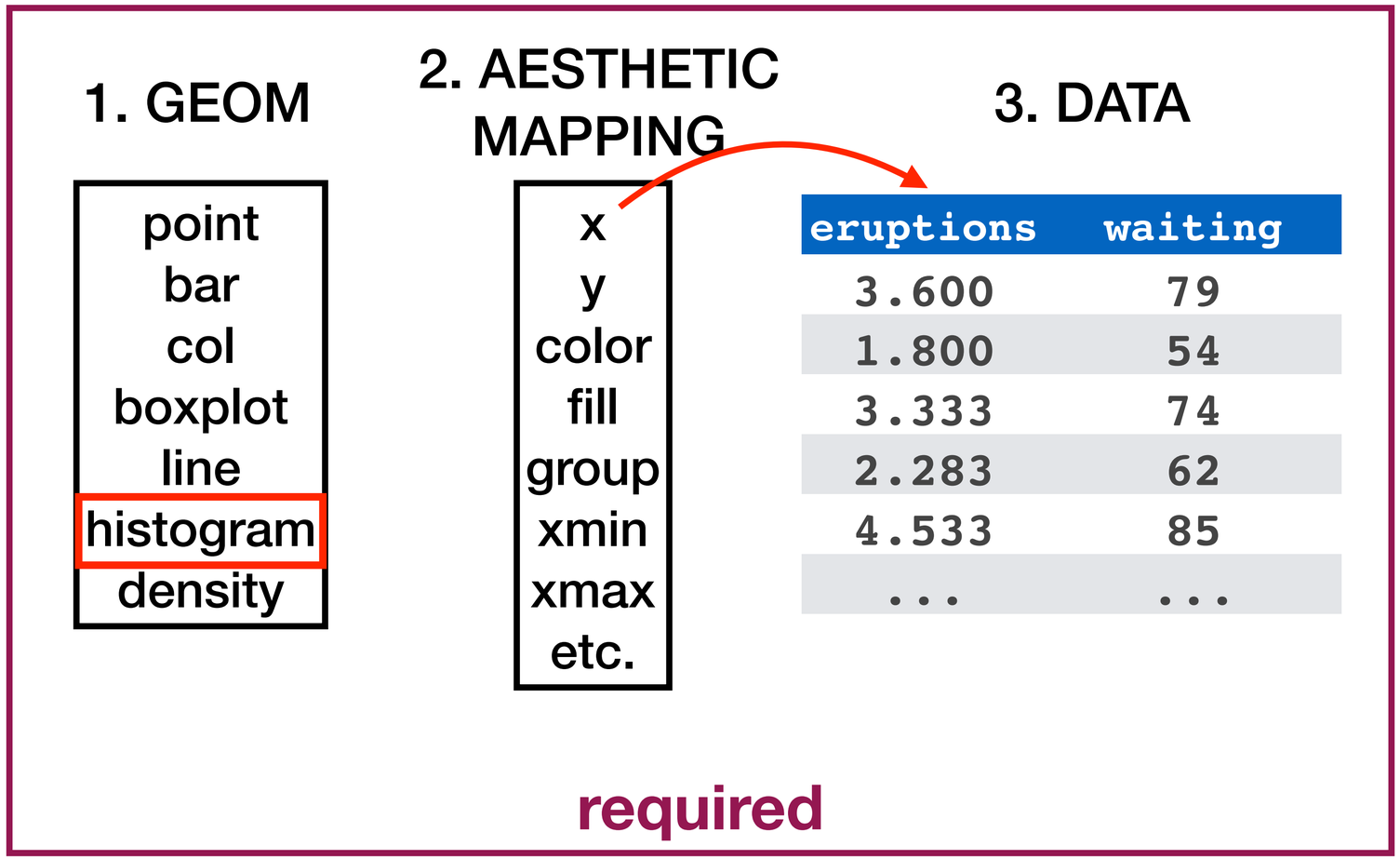
x

continuous

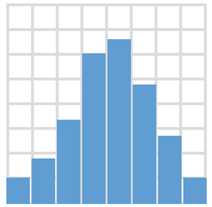


continuous

Putting it all together: start with the GEOM



Look at the data

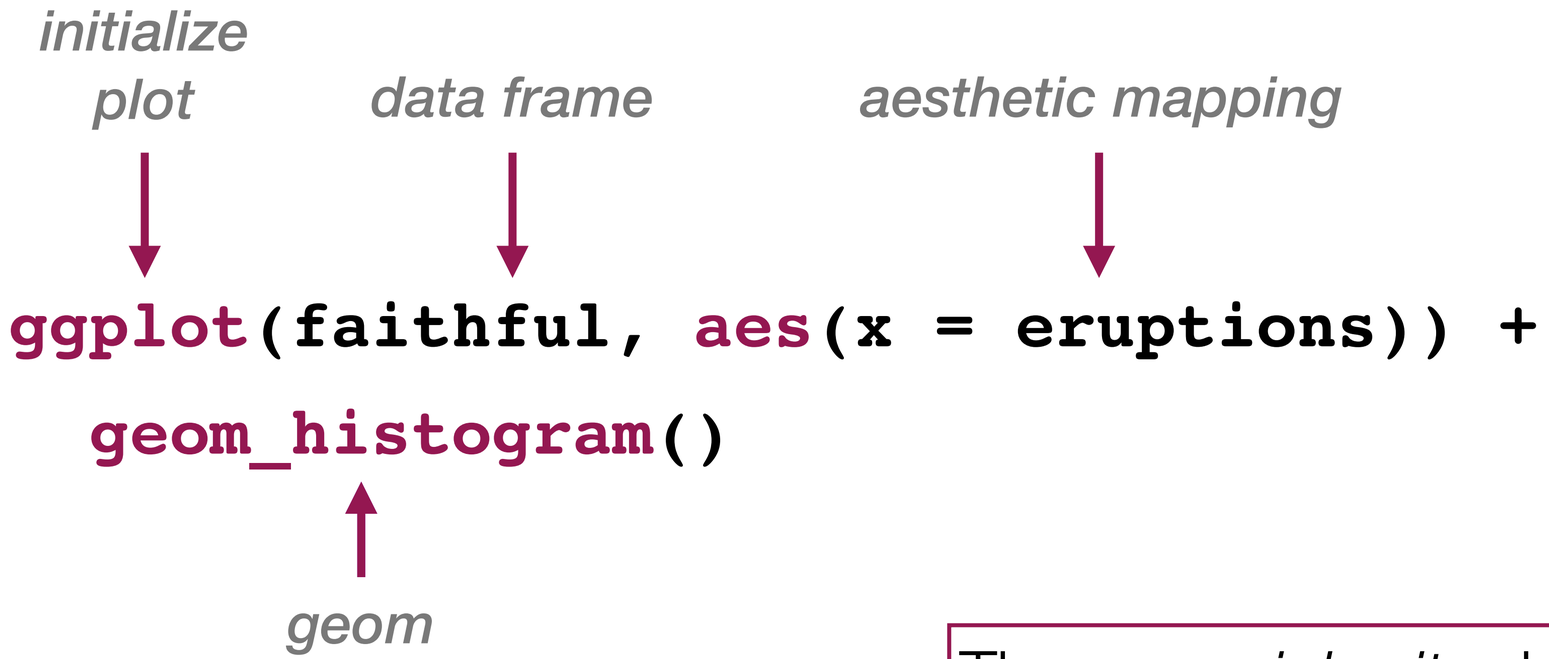


```
 1  str(faithful)
```

```
'data.frame':   272 obs. of  2 variables:
 $ eruptions: num  3.6 1.8 3.33 2.28 4.53 ...
 $ waiting  : num  79 54 74 62 85 55 88 85 51 85 ...
```

Remember: data must be continuous (numeric)!

Ready to code



The geom *inherits* data and mappings from the call to `ggplot()`

Do not start a new line with "+"

initialize plot

data frame

aesthetic mapping

```
ggplot(faithful, aes(x = eruptions)) +
  geom_histogram()
```

geom

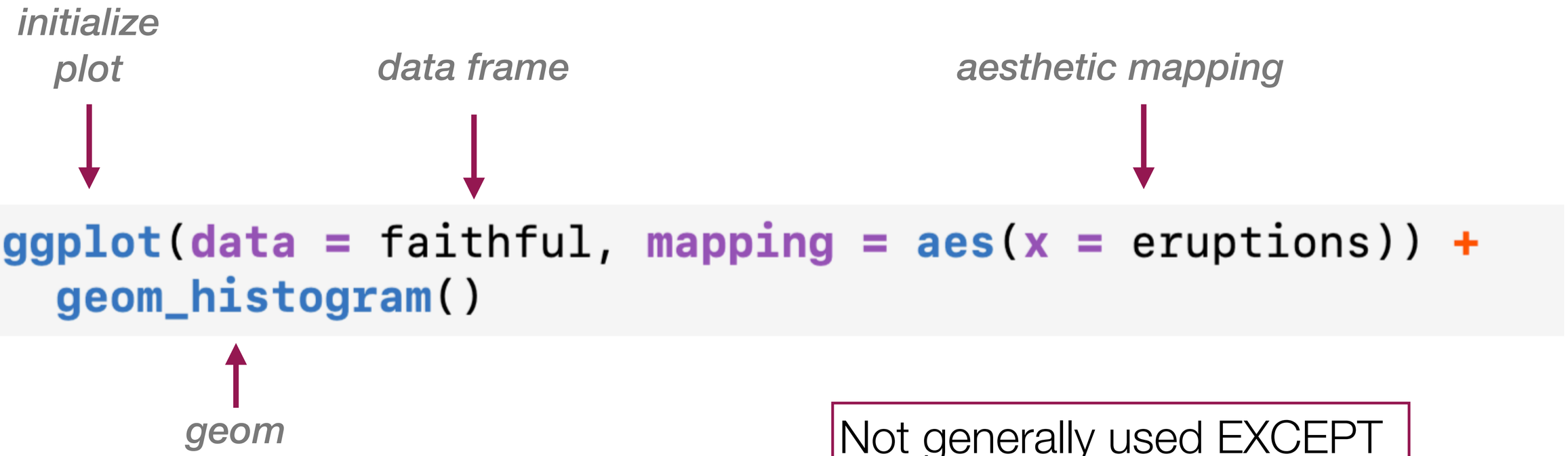
Good practice: end lines with "+"

If you do...


```
ggplot(faithful, aes(x = eruptions))
  + geom_histogram()
```

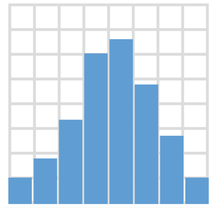
```
Error:
! Cannot use `+` with a single argument
ℹ Did you accidentally put `+` on a new line?
```

With parameter names

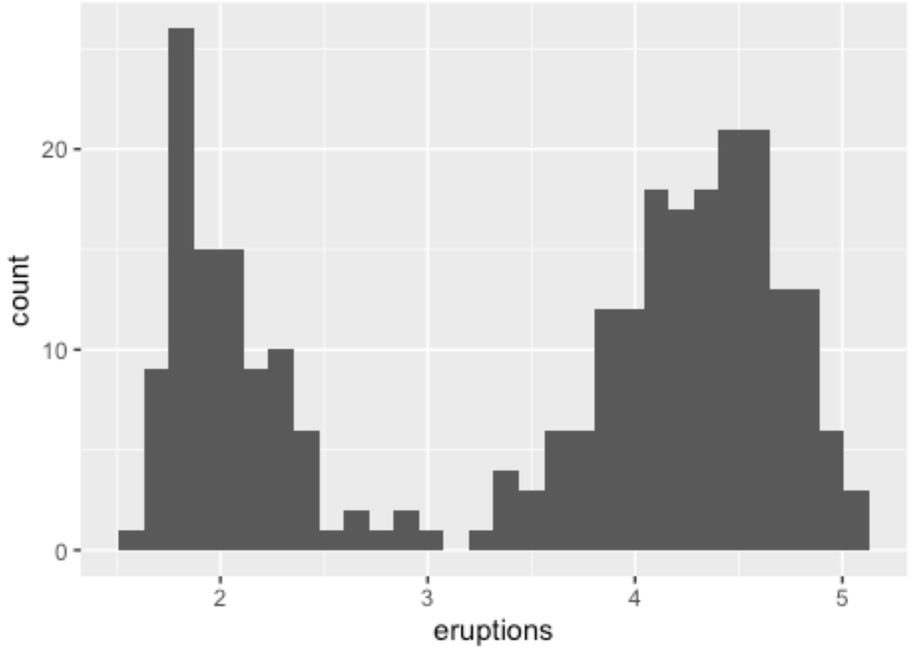


Not generally used EXCEPT for "`data =`" in GEOMS -- will discuss later

geom_histogram( )



```
ggplot(faithful, aes(x = eruptions)) +
  geom_histogram()
```



Mappings vs. settings

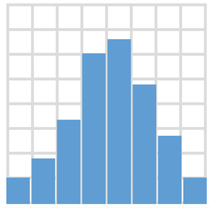
- *mappings* connect variables to aesthetics:

```
aes(x = eruptions)
```

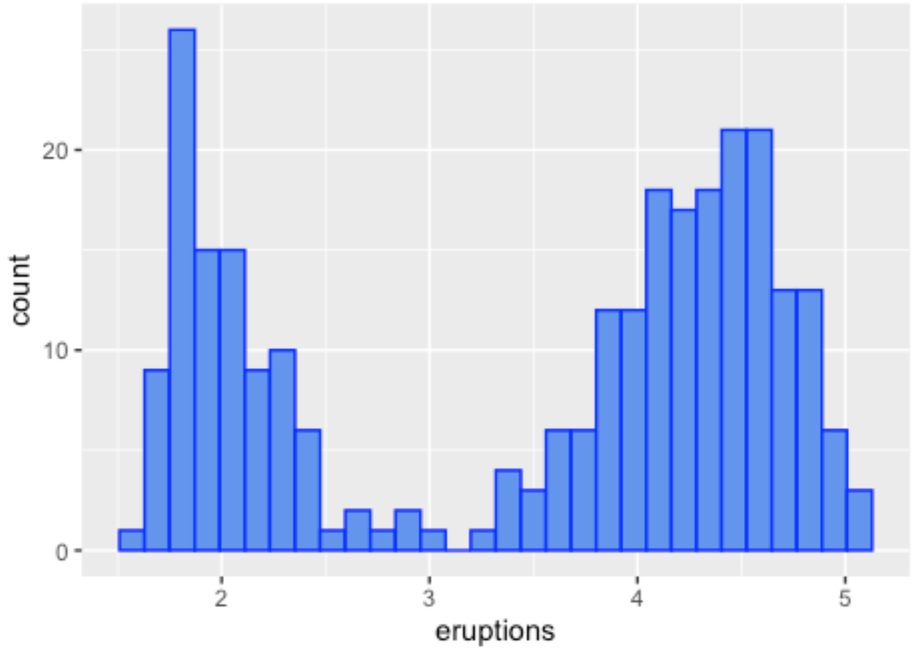
- *settings* specify constant values:

```
geom_histogram(color = "blue", fill = "cornflowerblue")
```

Change the color and fill



```
ggplot(faithful, aes(x = eruptions)) +
  geom_histogram(color = "blue", fill = "cornflowerblue")
```



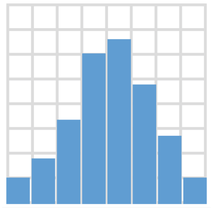
Color and fill

- Use **`color =`** for 0 or 1 dimensions (points, lines)
- Use **`fill =`** for 2 dimensions (area)
- Base R graphics users:

`border =` → **`color =`**

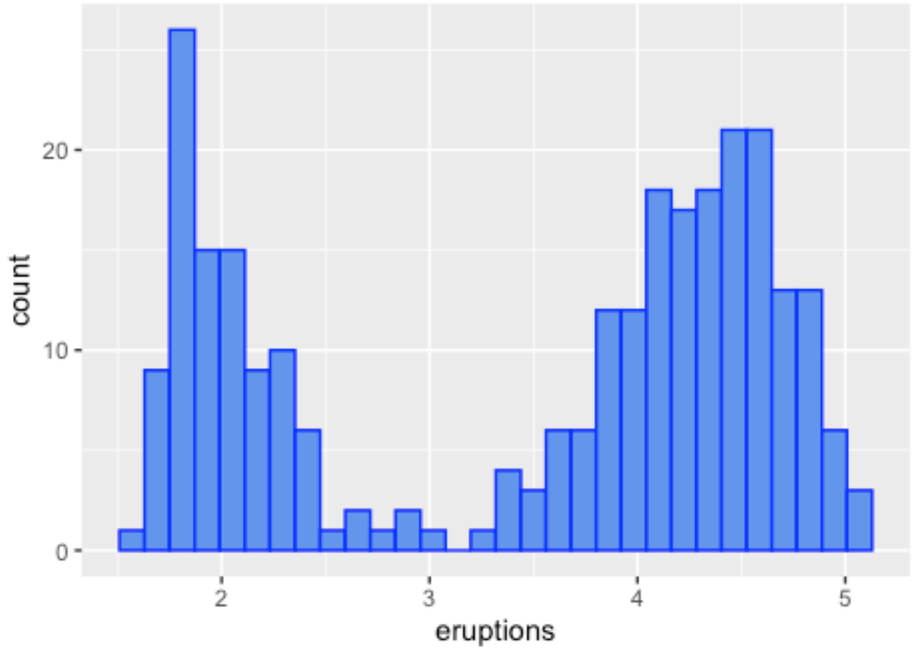
`col =` → **`fill =`**

Read the message!

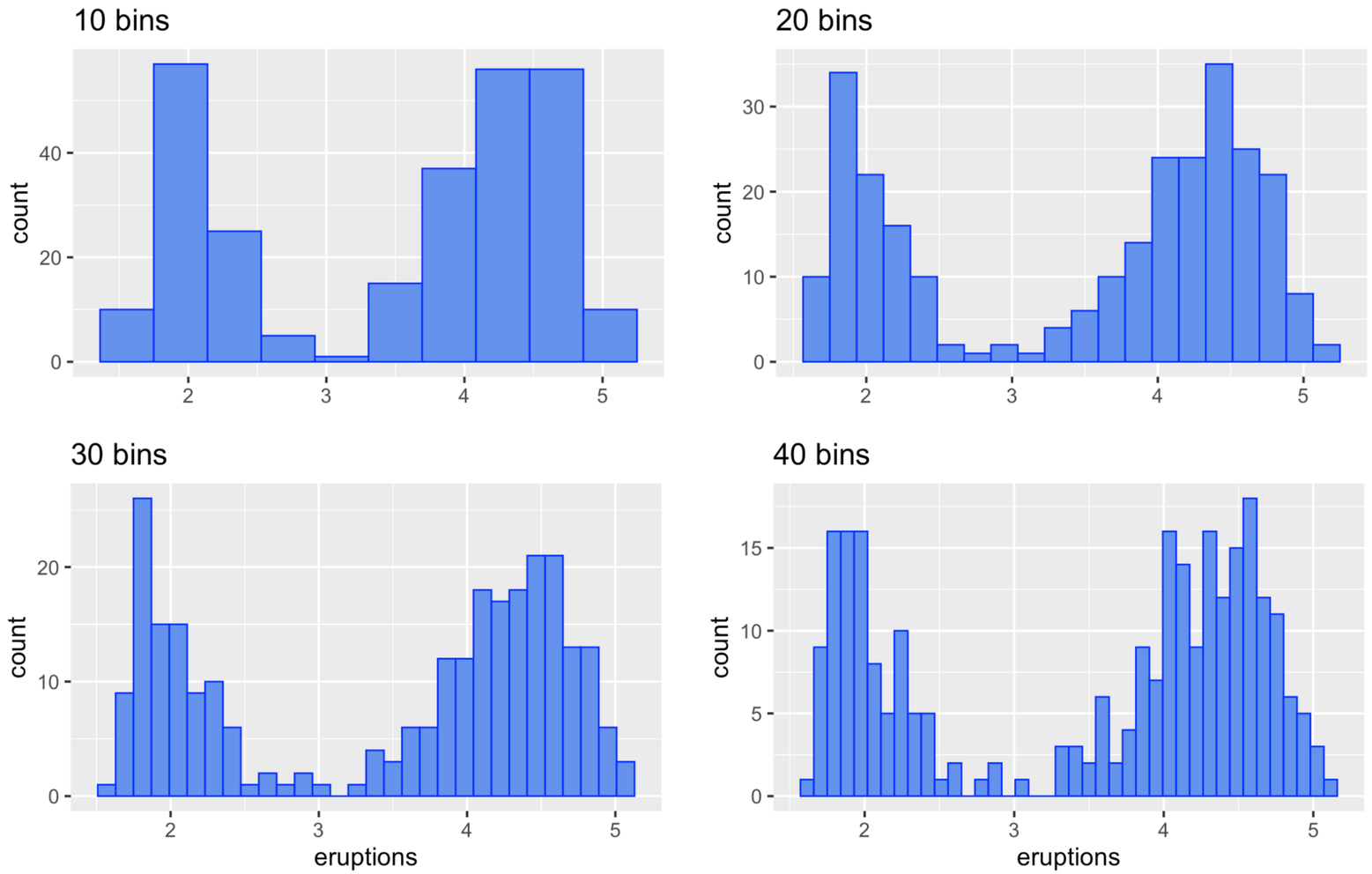


```
ggplot(faithful, aes(x = eruptions)) +
  geom_histogram()
```

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



Histogram bins / binwidth

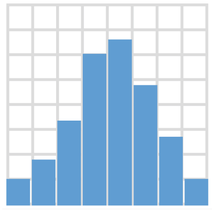


Bins / binwidth

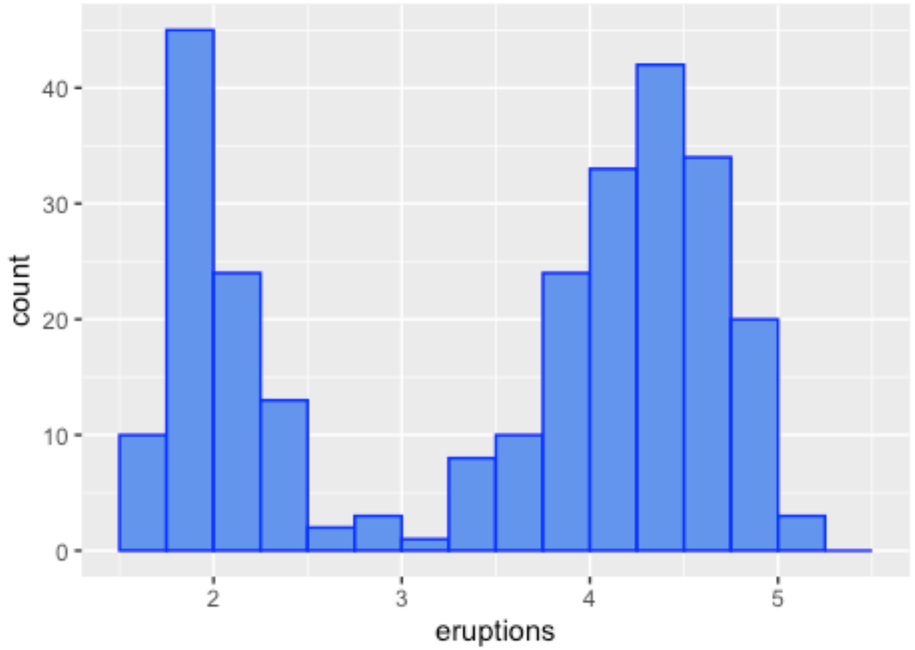
- Changing the number of bins is a *setting*
- Use trial and error to find the right "focus" in order to see the shape of the distribution
- Options: `bins = , binwidth = , breaks =`
- Examples:

```
bins = 20
binwidth = 10
breaks = c(0, 10, 20, 30)
breaks = seq(0, 1000, 100)
```

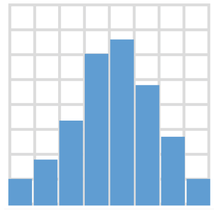
New bin boundaries



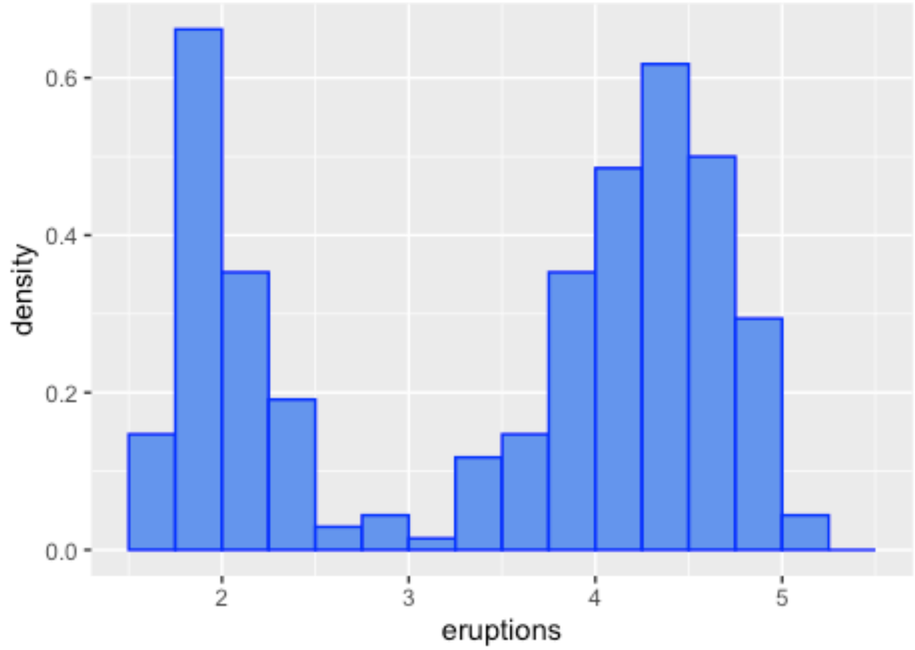
```
ggplot(faithful, aes(x = eruptions)) +
  geom_histogram(breaks = seq(1.5, 5.5, .25),
                 color = "blue",
                 fill = "cornflowerblue")
```



Density histogram



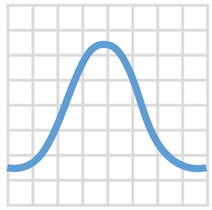
```
ggplot(faithful, aes(x = eruptions, y = after_stat(density))) +
  geom_histogram(color = "blue",
                 fill = "cornflowerblue")
```



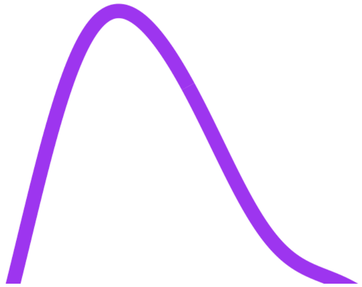
Replaces the default:
`after_stat(count)`

(not very common except for histograms)

`geom_density()`



- Like a histogram, shows the distribution of a continuous variable
- Requires an **x** *or* **y** (rare) mapping
- The other axis (usually **y**) defaults to a density scale



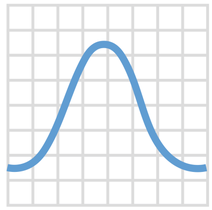
x

continuous

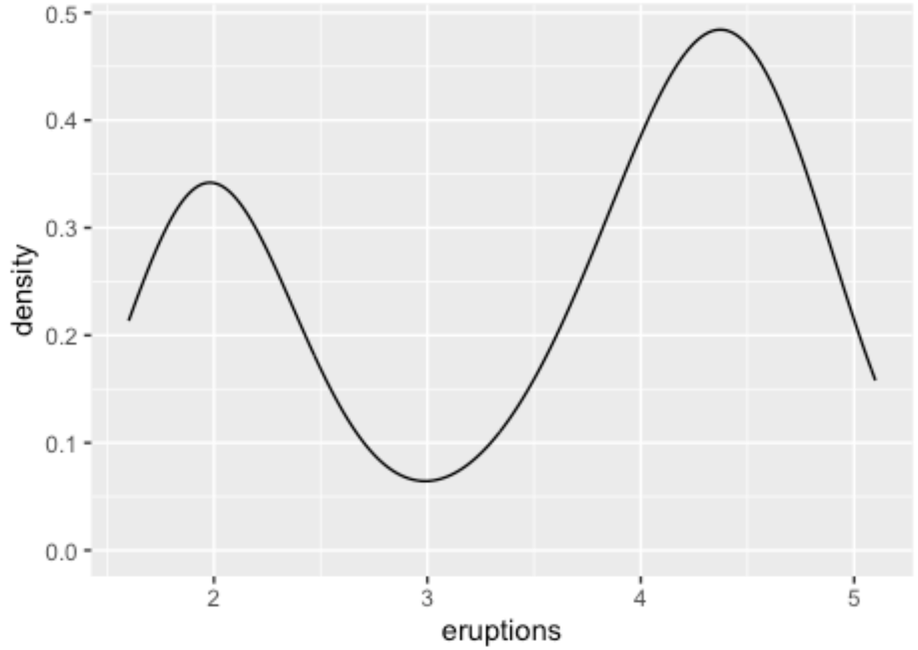
y

continuous

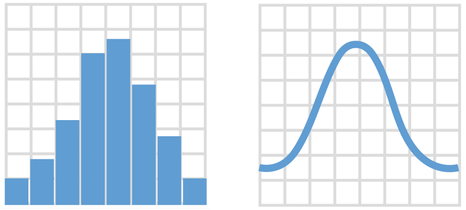
geom_density()



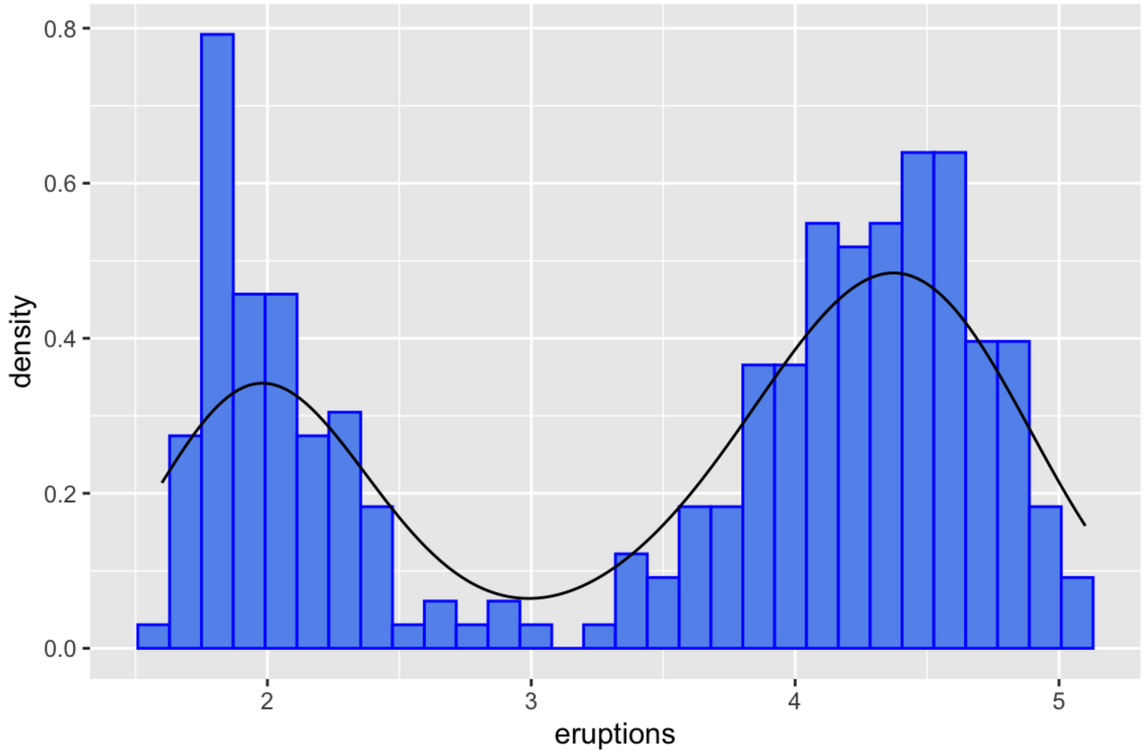
```
ggplot(faithful, aes(x = eruptions)) +
  geom_density()
```



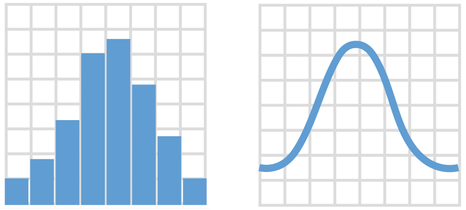
Two GEOMs



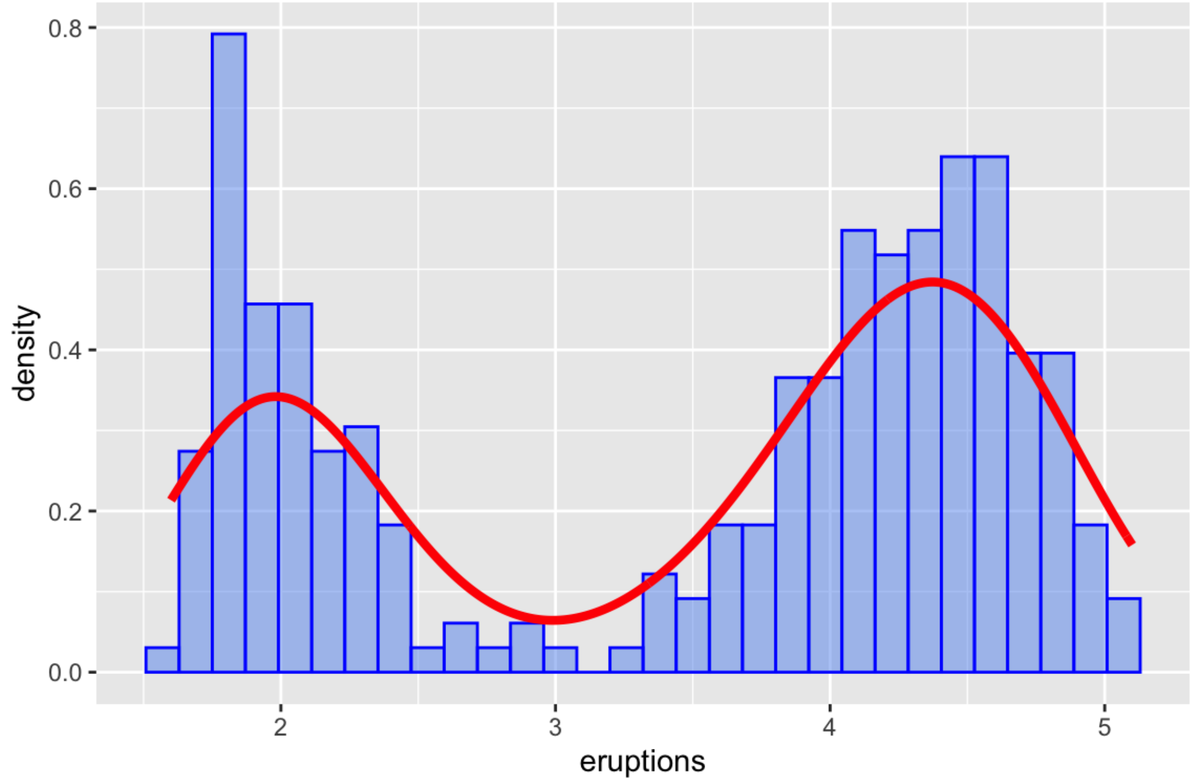
```
ggplot(faithful, aes(x = eruptions, y = after_stat(density))) +
  geom_histogram(color = "blue", fill = "cornflowerblue") +
  geom_density()
```



Change settings

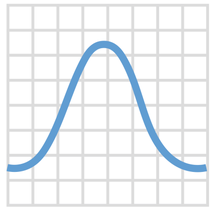


```
ggplot(faithful, aes(x = eruptions, y = after_stat(density))) +
  geom_histogram(color = "blue", fill = "cornflowerblue", alpha = .5) +
  geom_density(linewidth = 1.5, color = "red")
```



`lwd` also works for `linewidth` (default = 0.5)

Add fill and alpha transparency



```
ggplot(faithful, aes(x = eruptions)) +
  geom_density(linewidth = 1, color = "red", fill = "red", alpha = .25)
```

The scale of `alpha` is 0 to 1 (0% to 100% transparency)