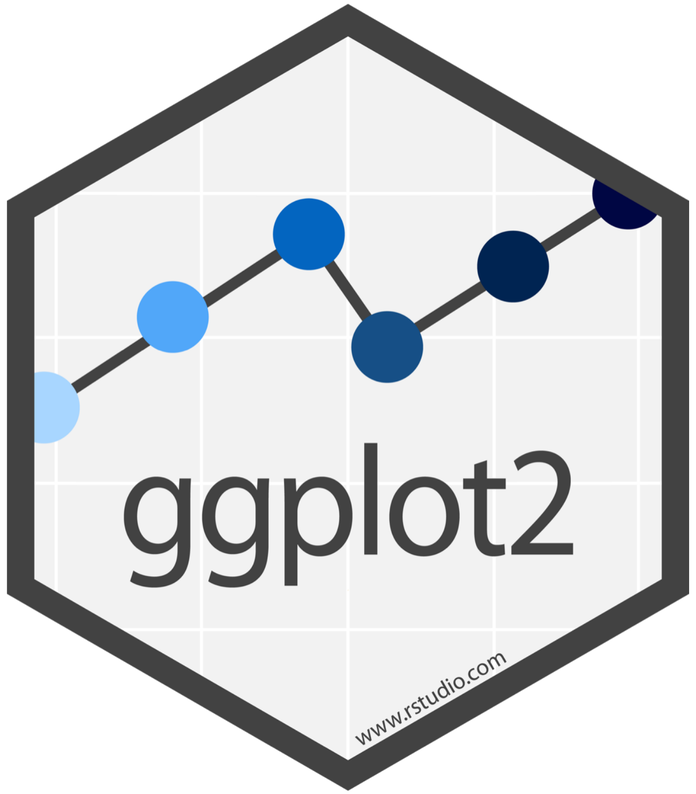
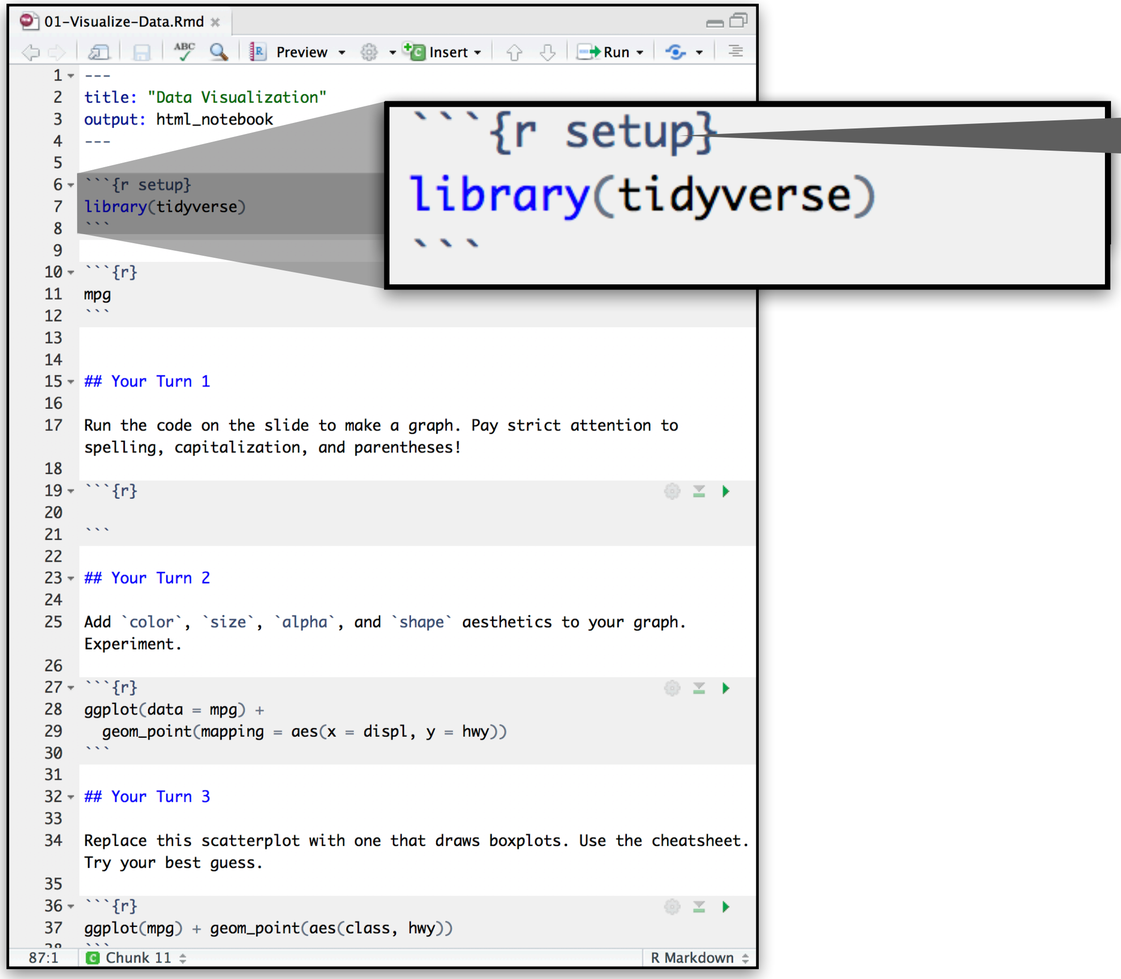
Visualize Data with



Setup

The setup chunk is always run once before anything else



chunk labels are optional, the setup label is special

warwick

Rock sample data from the Wentworth area, Nova Scotia

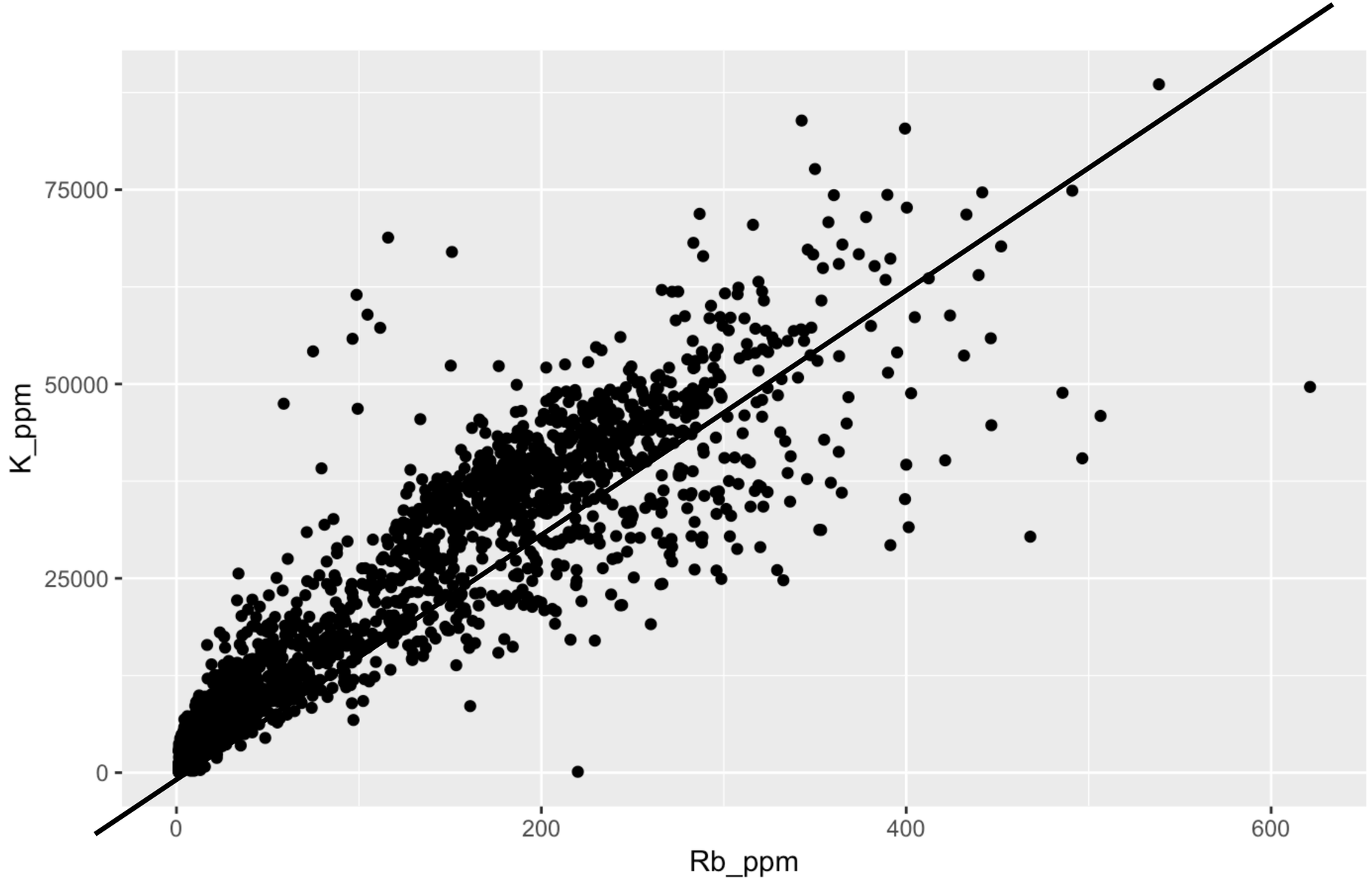
```
warwick
```

Your Turn 1

Run this code in your notebook to make a graph.

Pay strict attention to spelling, capitalization, and parentheses!

```
ggplot(data = warwick) +
  geom_point(mapping = aes(x = K_ppm, y = Rb_ppm))
```



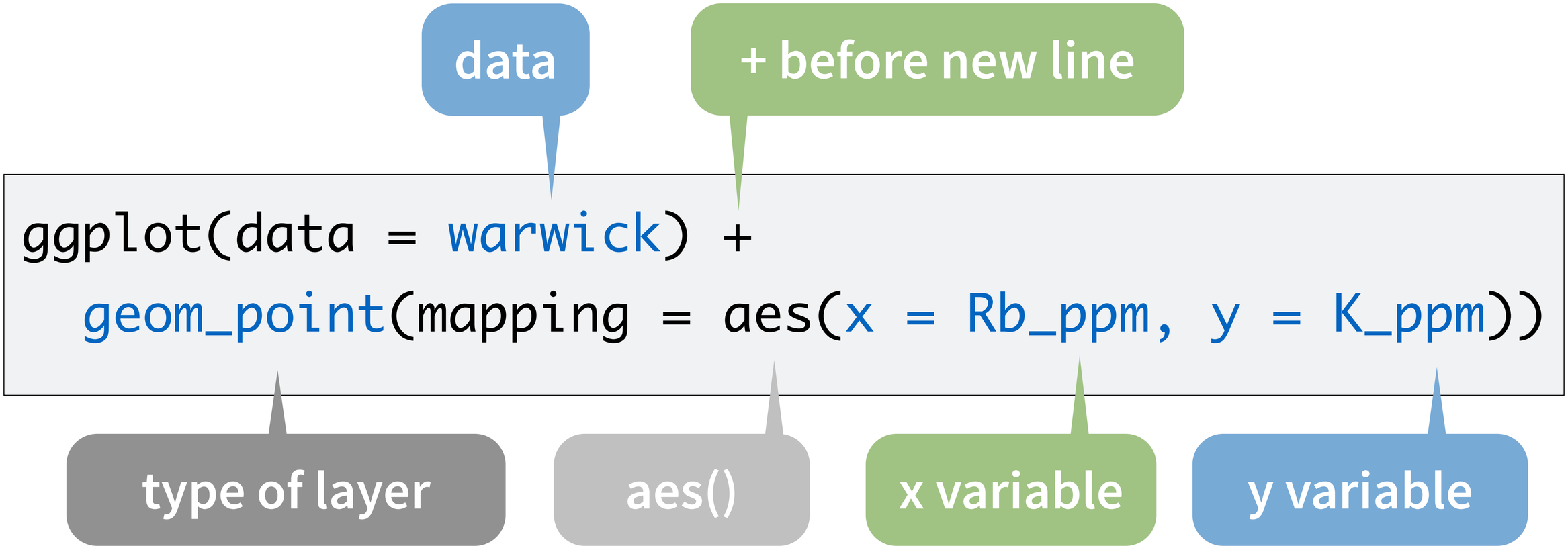

```
ggplot(data = warwick) +
  geom_point(mapping = aes(x = Rb_ppm, y = K_ppm))
```

1. "Initialize" a plot with ggplot()
2. Add layers with geom_ functions

```
ggplot(data = warwick) +
  geom_point(mapping = aes(x = Rb_ppm, y = K_ppm))
```

Pro tip: Always put the + at the end of a line, Never at the start

```
ggplot(data = warwick) +
  geom_point(mapping = aes(x = Rb_ppm, y = K_ppm))
```


data
+ before new line
ggplot(data = warwick) +
geom_point(mapping = aes(x = Rb_ppm, y = K_ppm))
type of layer
aes()
x variable
y variable

A template

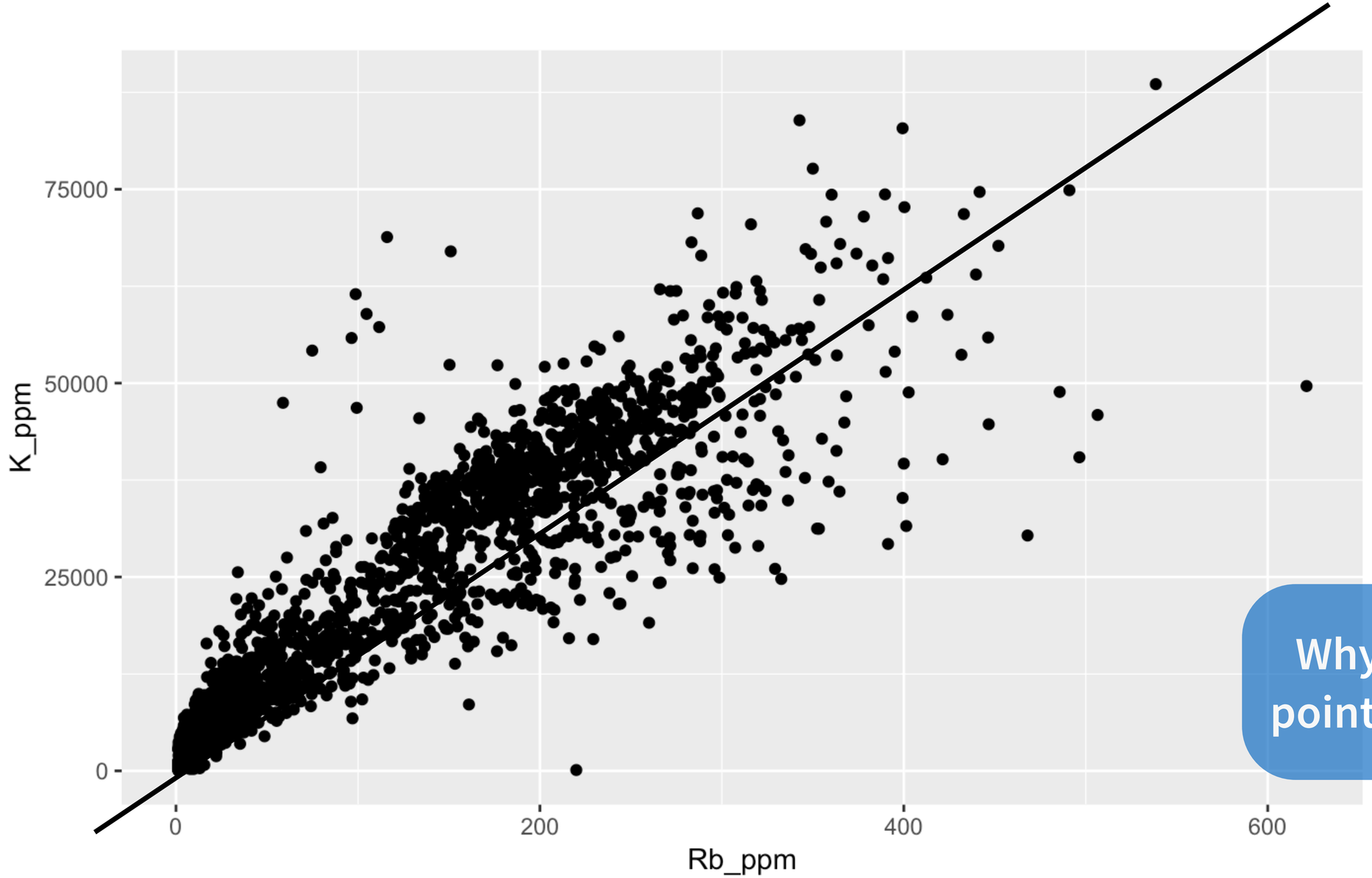
```
ggplot(data = <DATA>) +
  <GEOM_FUNCTION>(mapping = aes(<MAPPINGS>))
```

```
  geom_point(mapping = aes(x = Rb_ppm, y = K_ppm))
```

A template

```
ggplot(data = <DATA>) +
  <GEOM_FUNCTION>(mapping = aes(<MAPPINGS>))
```

Mappings

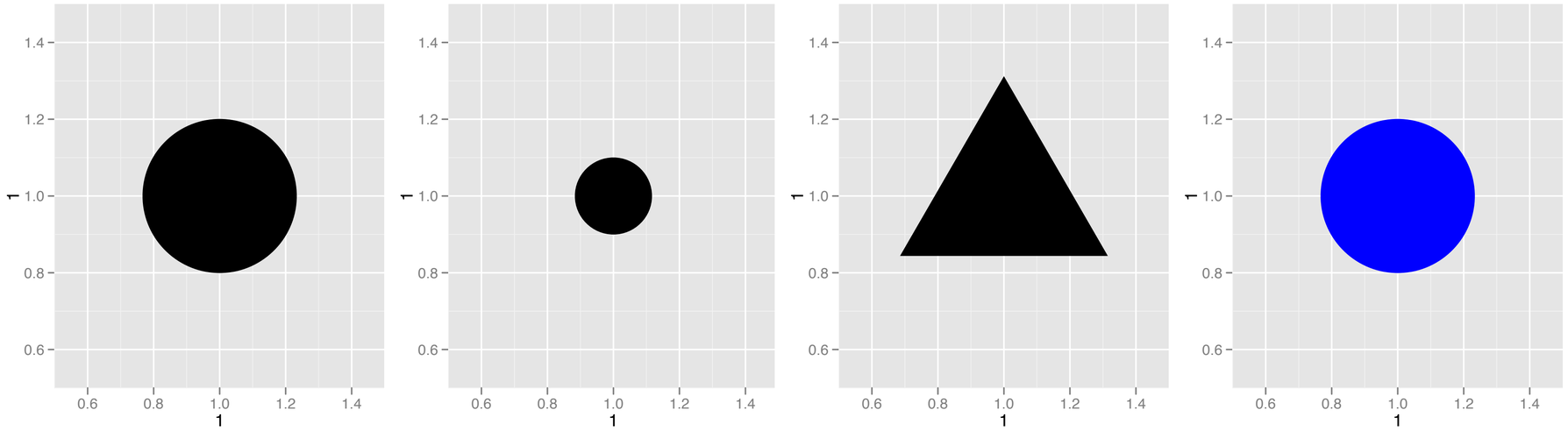


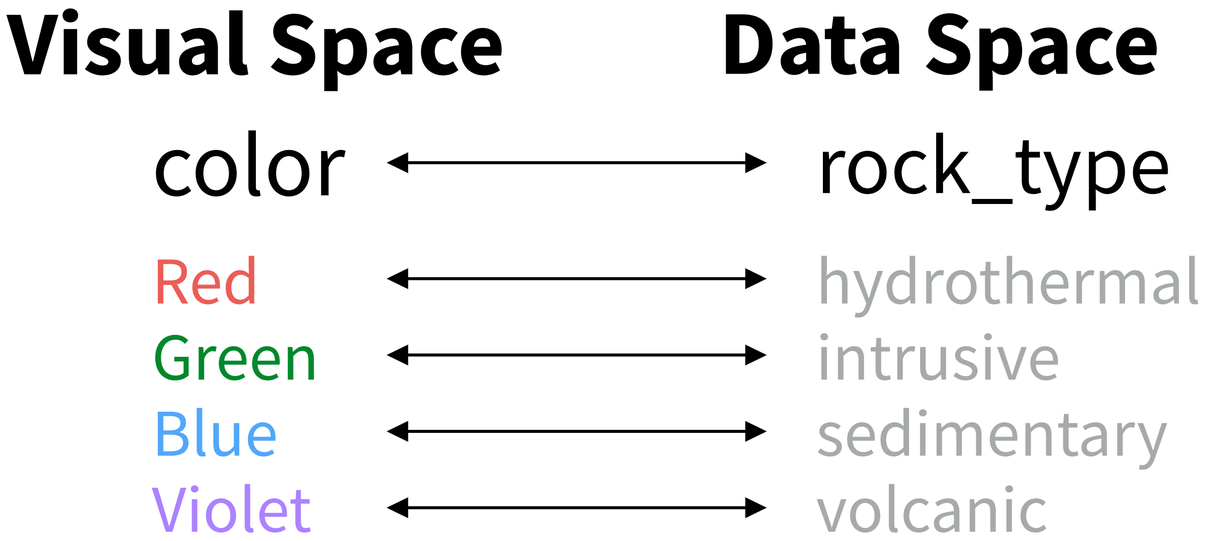
Why are some points different?

```
ggplot(data = warwick) +
  geom_point(mapping = aes(x = Rb_ppm, y = K_ppm))
```

ggplot2

Aesthetics



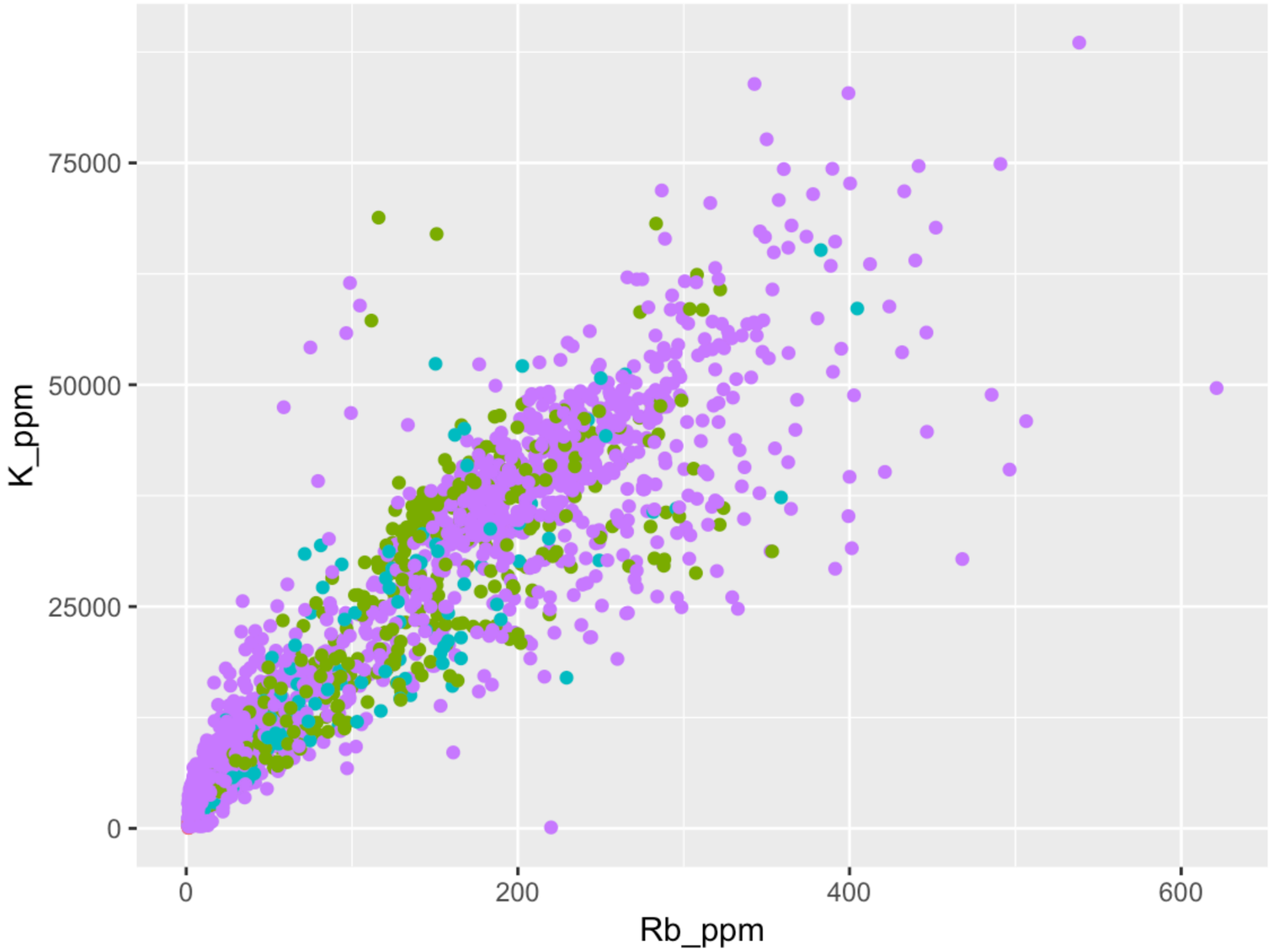

Visual Space
Data Space
color
rock_type
Red
hydrothermal
Green
intrusive
Blue
sedimentary
Violet
volcanic

Aesthetics

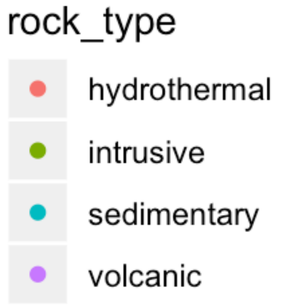
aesthetic property

Variable to map it to

```
ggplot(mpg) + geom_point(aes(x = displ, y = hwy, color = rock_type))
ggplot(mpg) + geom_point(aes(x = displ, y = hwy, size = rock_type))
ggplot(mpg) + geom_point(aes(x = displ, y = hwy, shape = rock_type))
ggplot(mpg) + geom_point(aes(x = displ, y = hwy, alpha = rock_type))
```



Legend added automatically



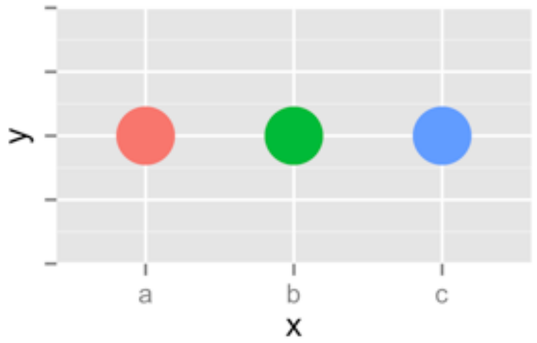
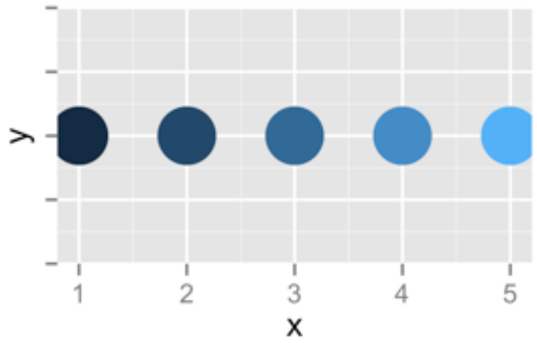
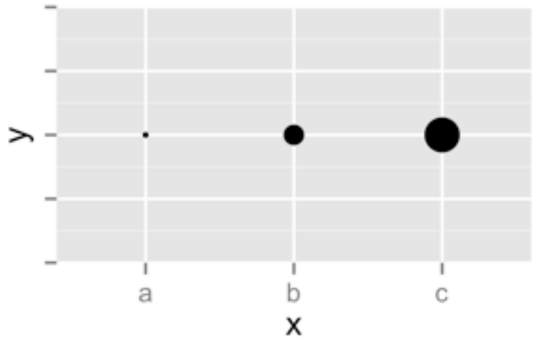
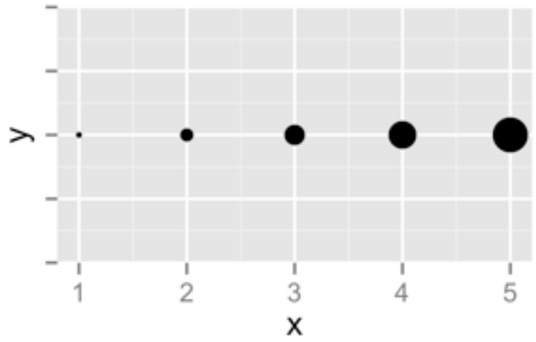
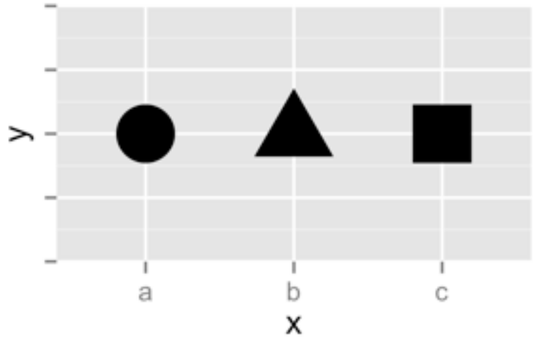

```
ggplot(data = warwick) +
  geom_point(mapping = aes(x = Rb_ppm, y = K_ppm, color = rock_type))
```

Your Turn 2

In the next chunk, add color, size, alpha, and shape aesthetics to your graph. Experiment.

Do different things happen when you map aesthetics to discrete and continuous variables?

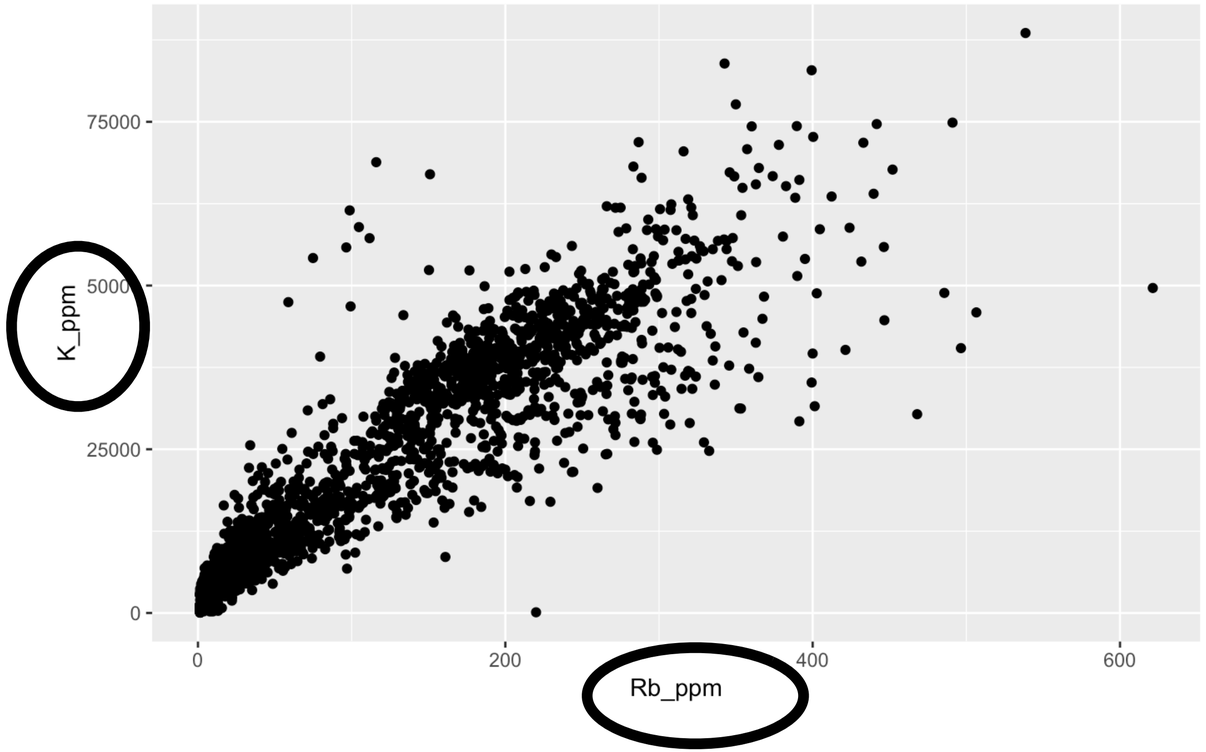
What happens when you use more than one aesthetic?

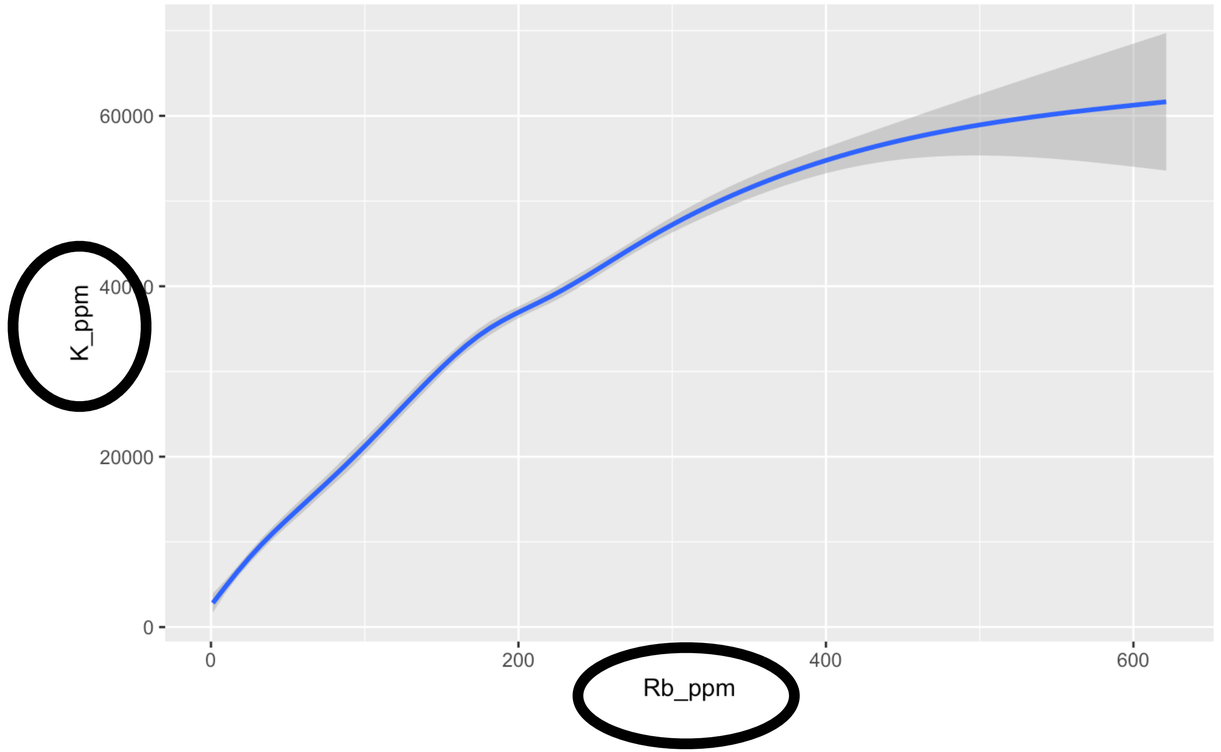
	Discrete	Continuous
Color		
Size		
Shape		

Geoms

How are these plots similar?

Same: x var , y var , data

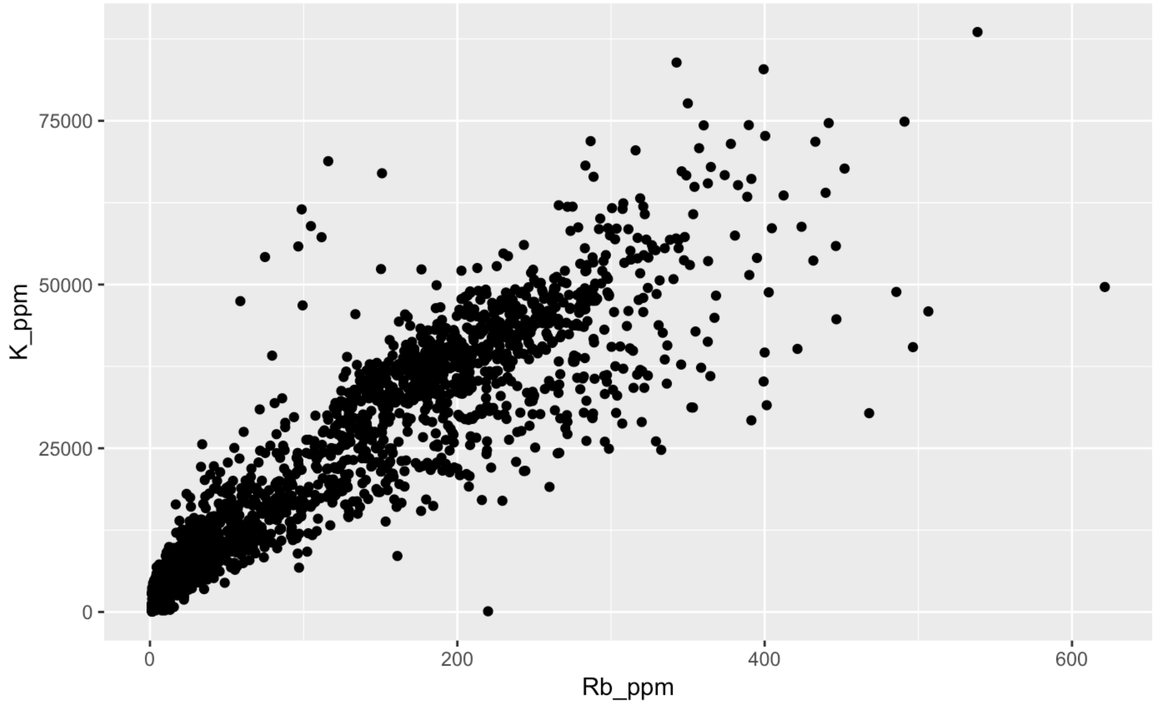


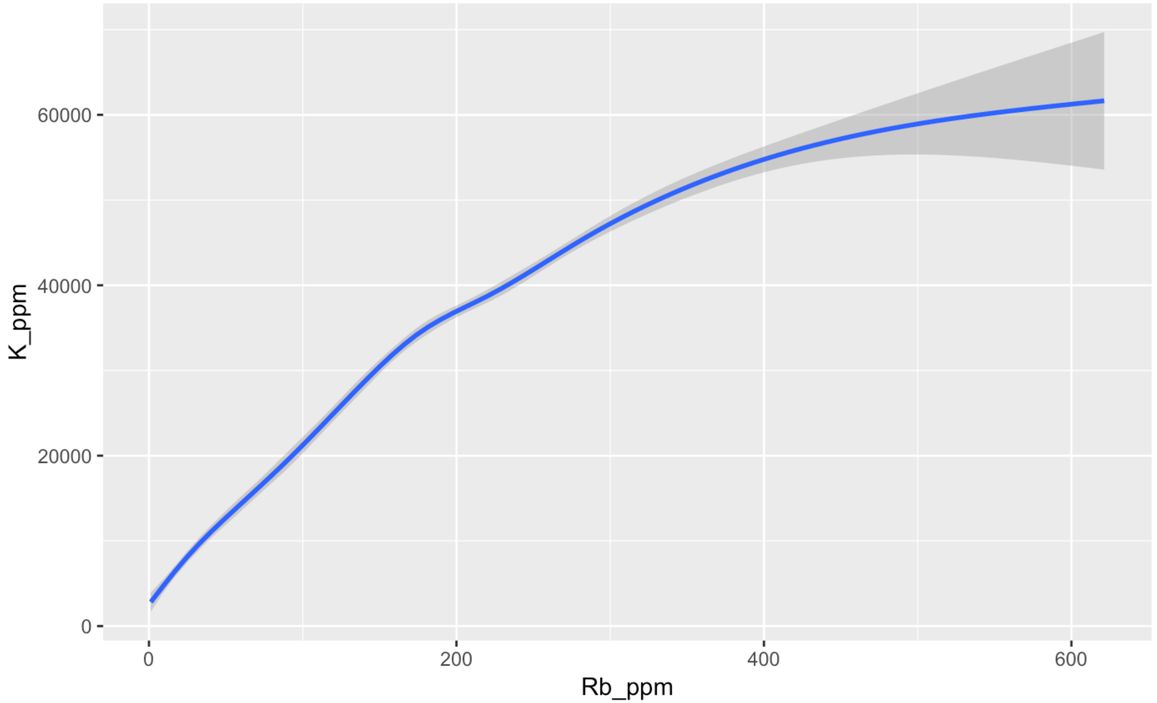


How are these plots different?

Different: geometric object (geom),

e.g. the visual object used to represent the data



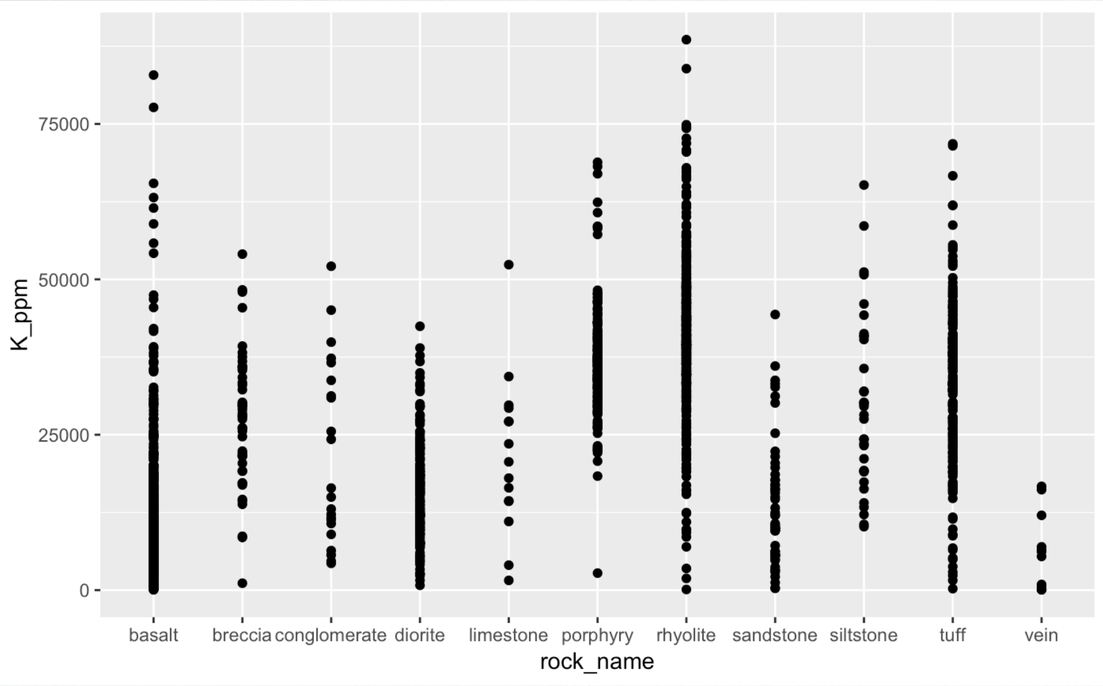


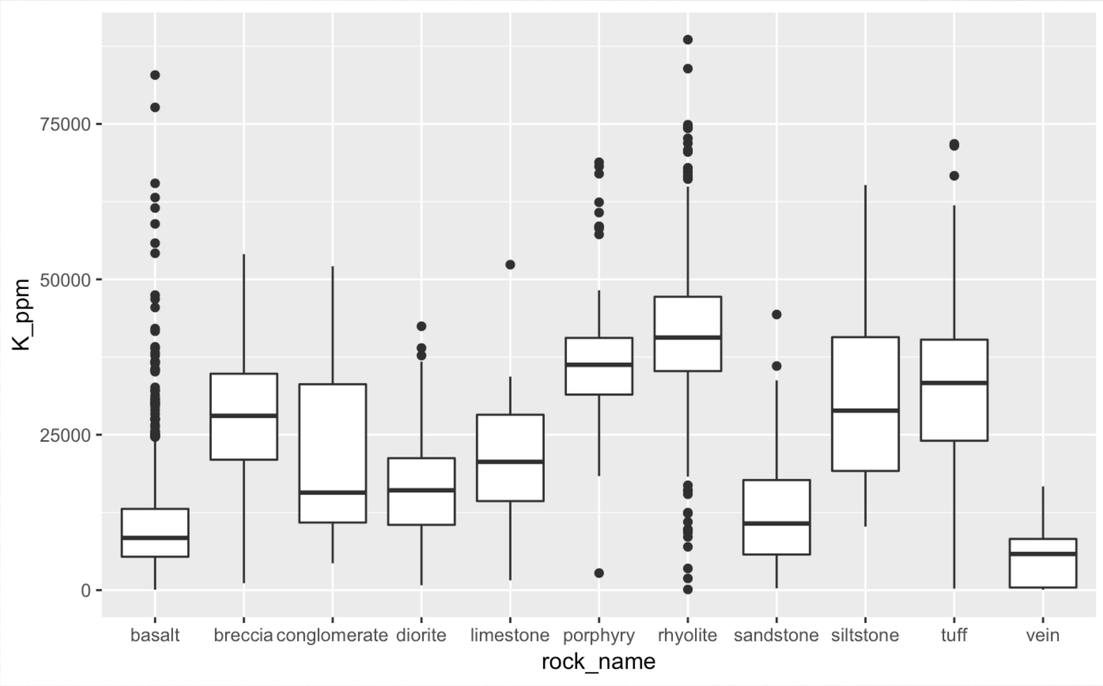
geoms

```
ggplot(data = <DATA>) +
  <GEOM_FUNCTION>(mapping = aes(<MAPPINGS>))
```

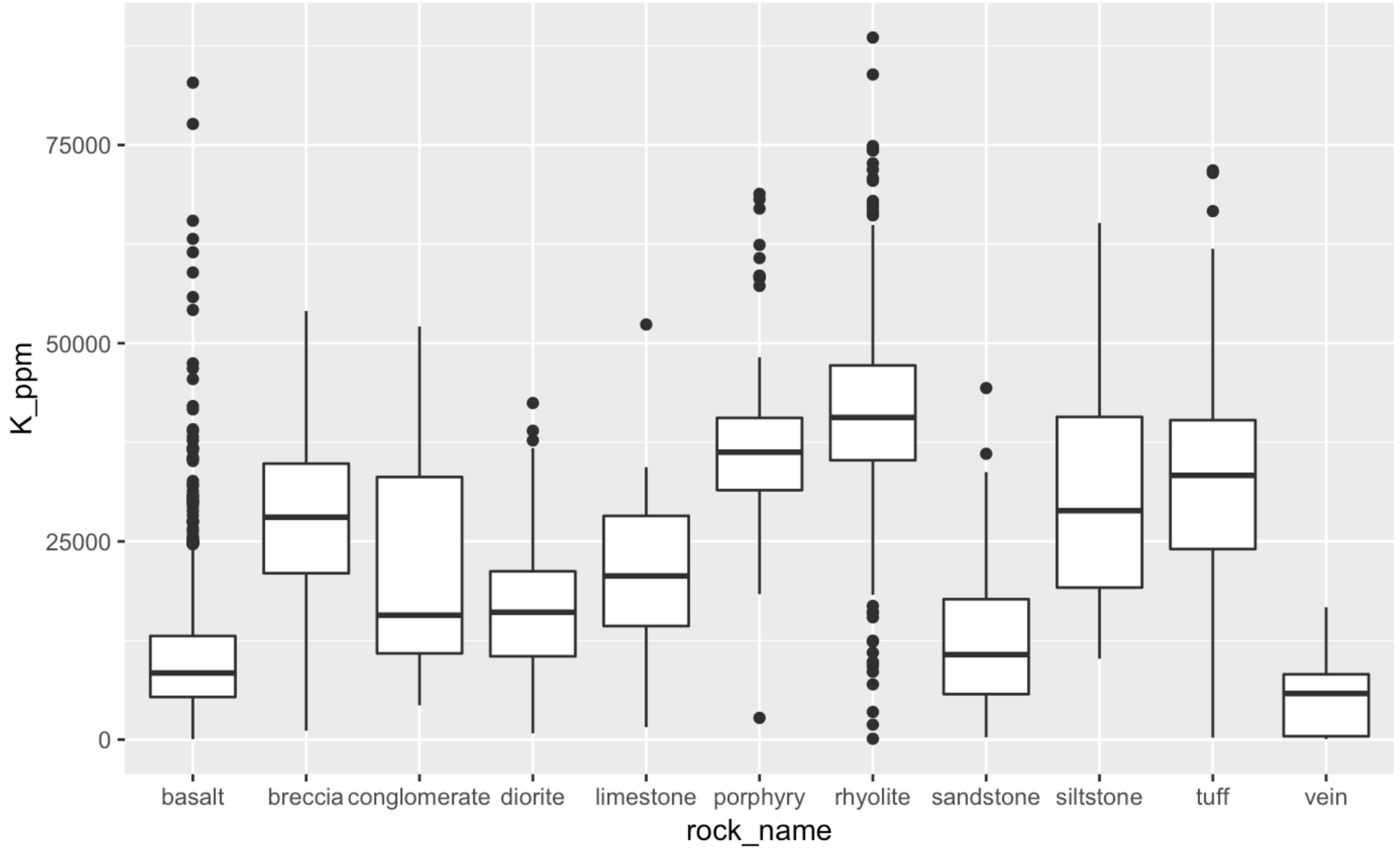
Your Turn 3

Decide how to replace this scatterplot with one that draws boxplots. Use the cheatsheet. Try your best guess.





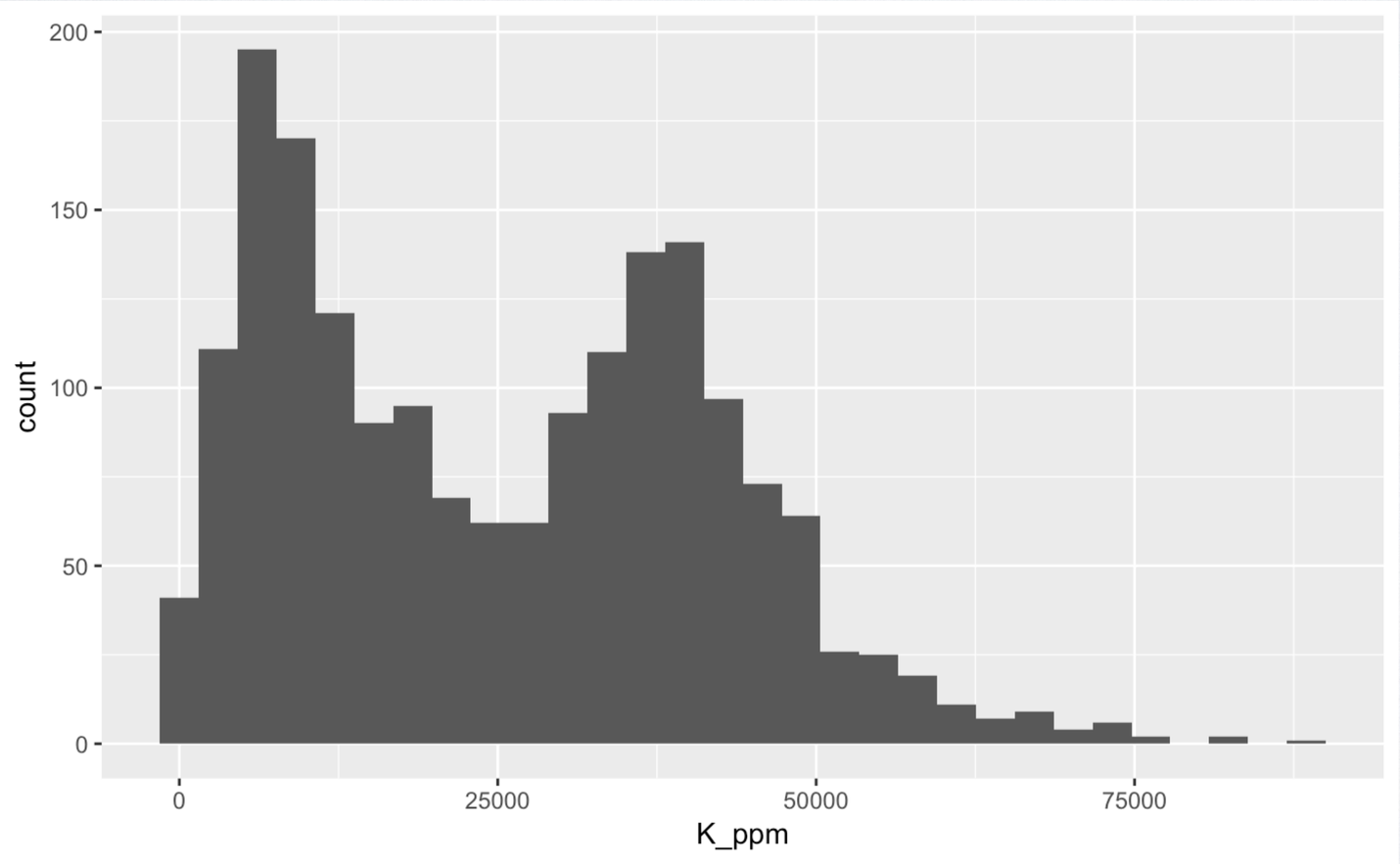

```
ggplot(warwick) + geom_point(aes(rock_name, K_ppm))
```

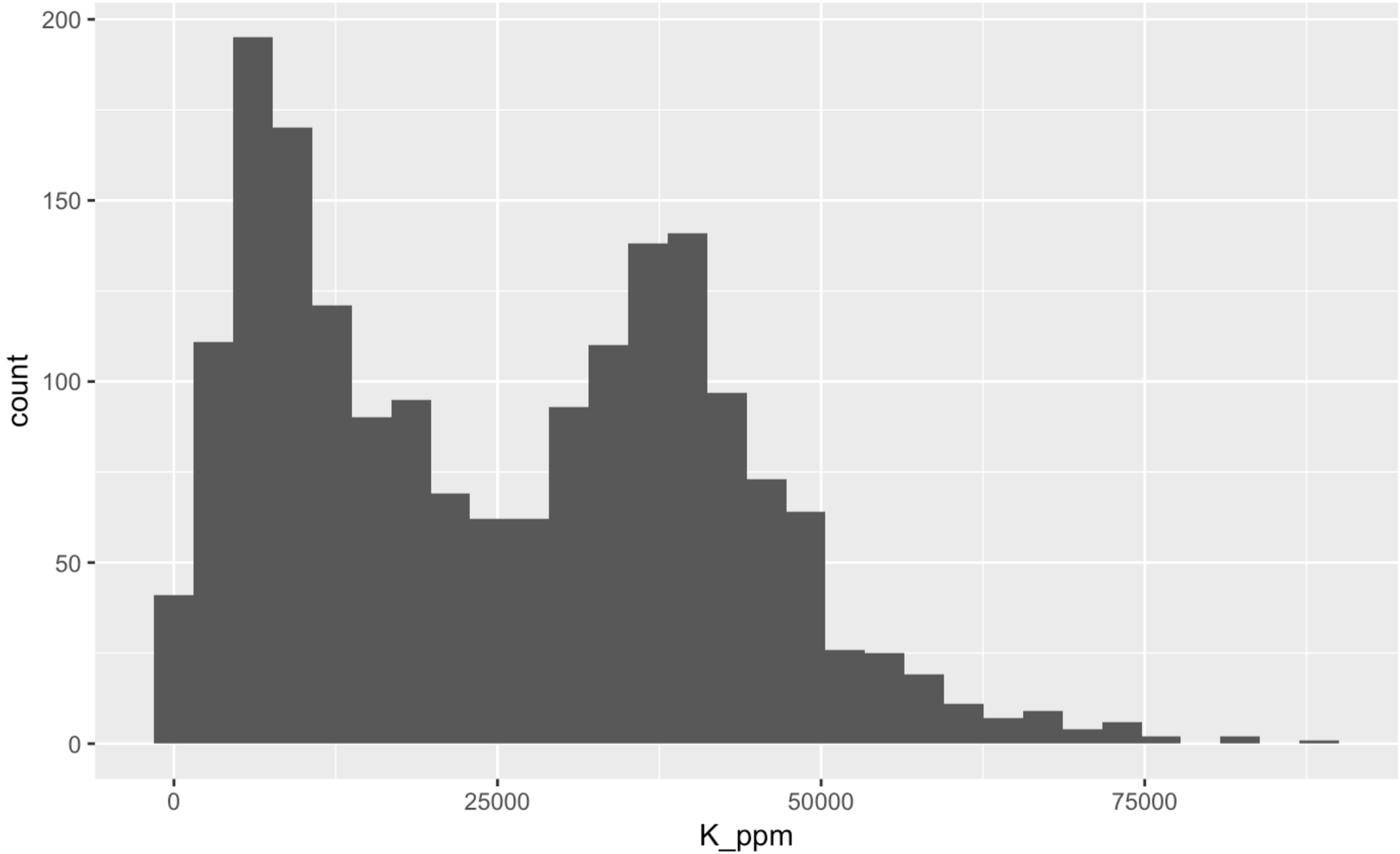



```
ggplot(data = warwick) +
  geom_boxplot(mapping = aes(x = rock_name, y = K_ppm))
```

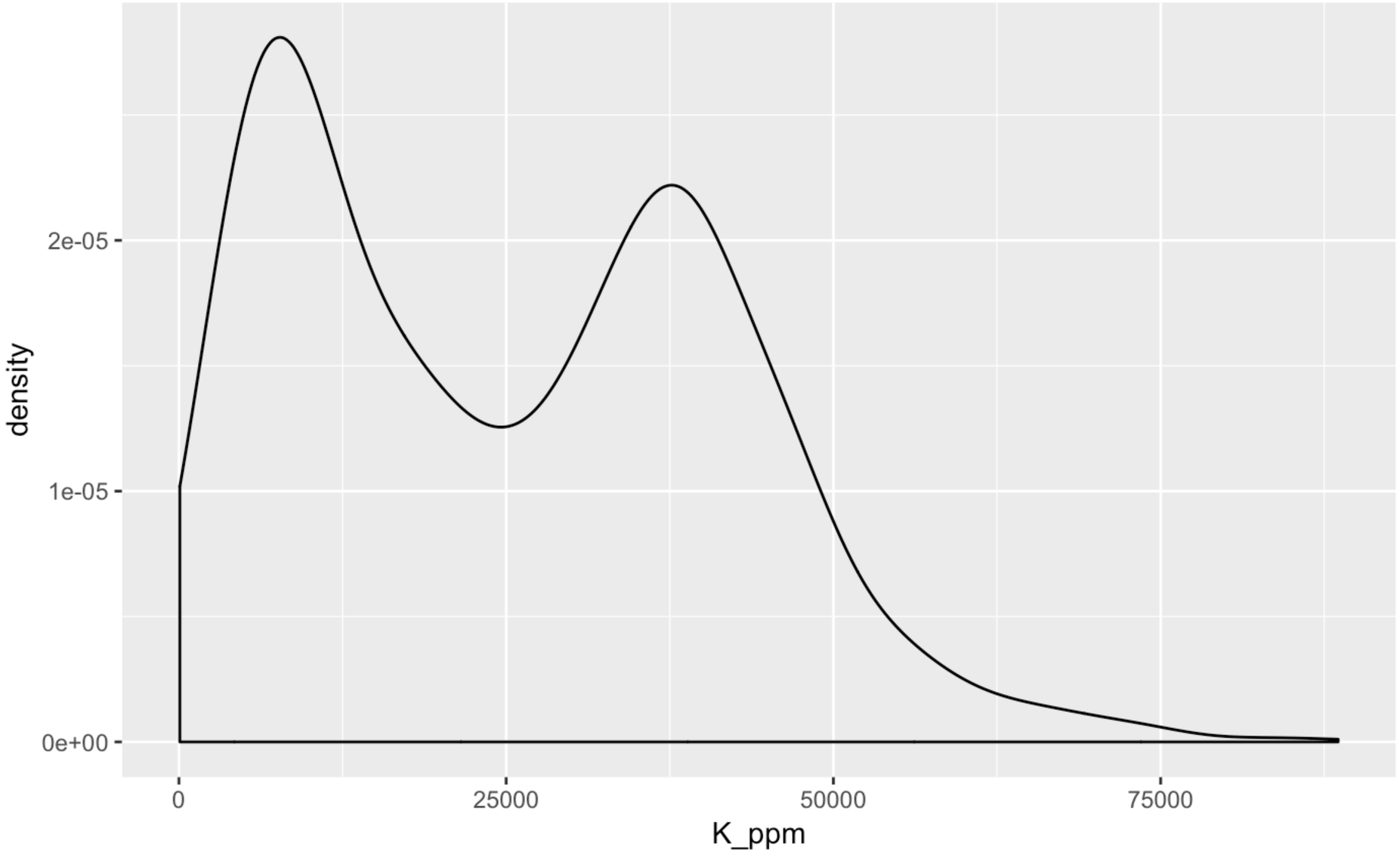
Your Turn 4

With your partner, make the histogram of **K_ppm** below. Use the cheatsheet. Hint: do not supply a **y** variable.





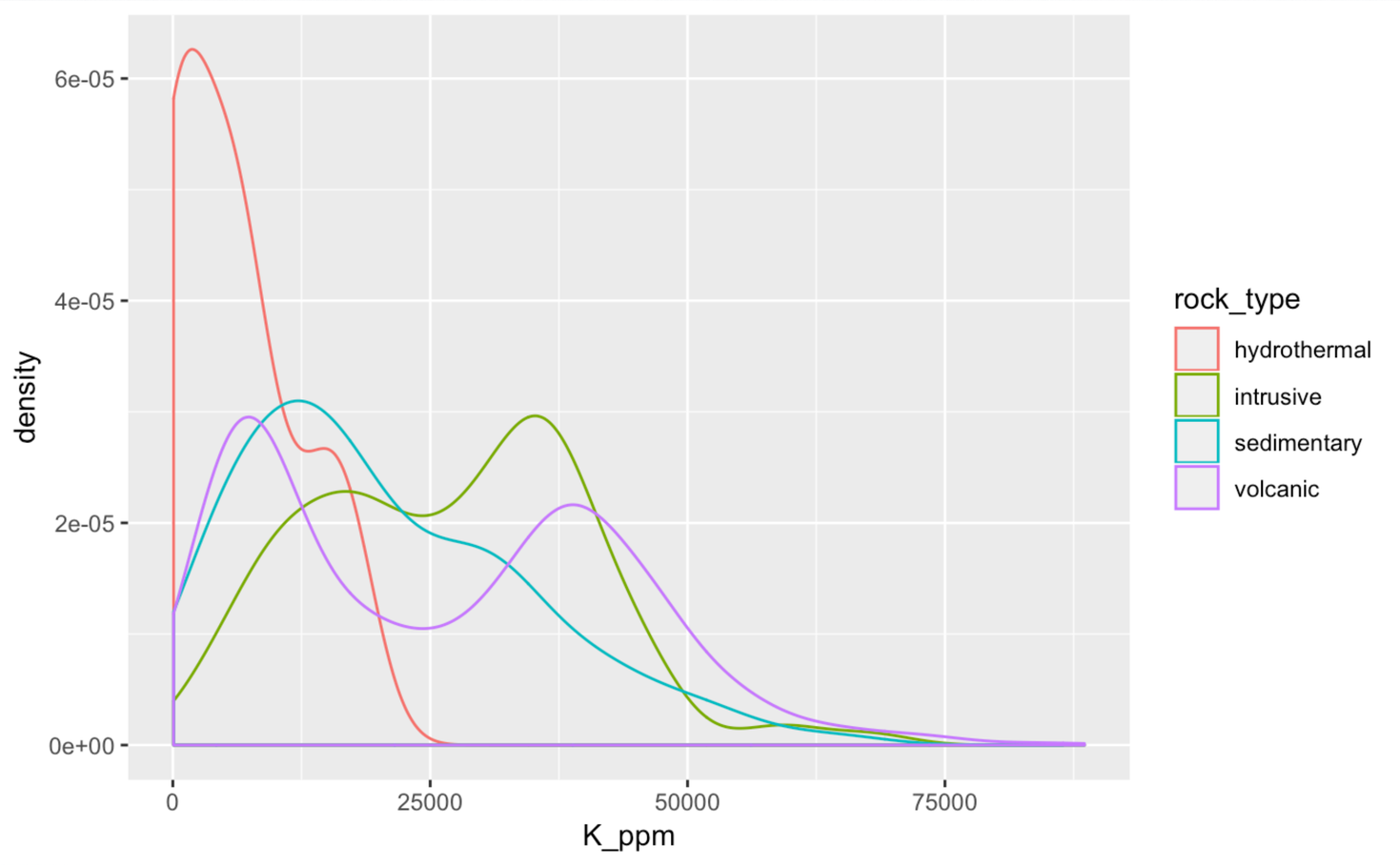

```
ggplot(data = warwick) +
  geom_histogram(mapping = aes(x = K_ppm))
```



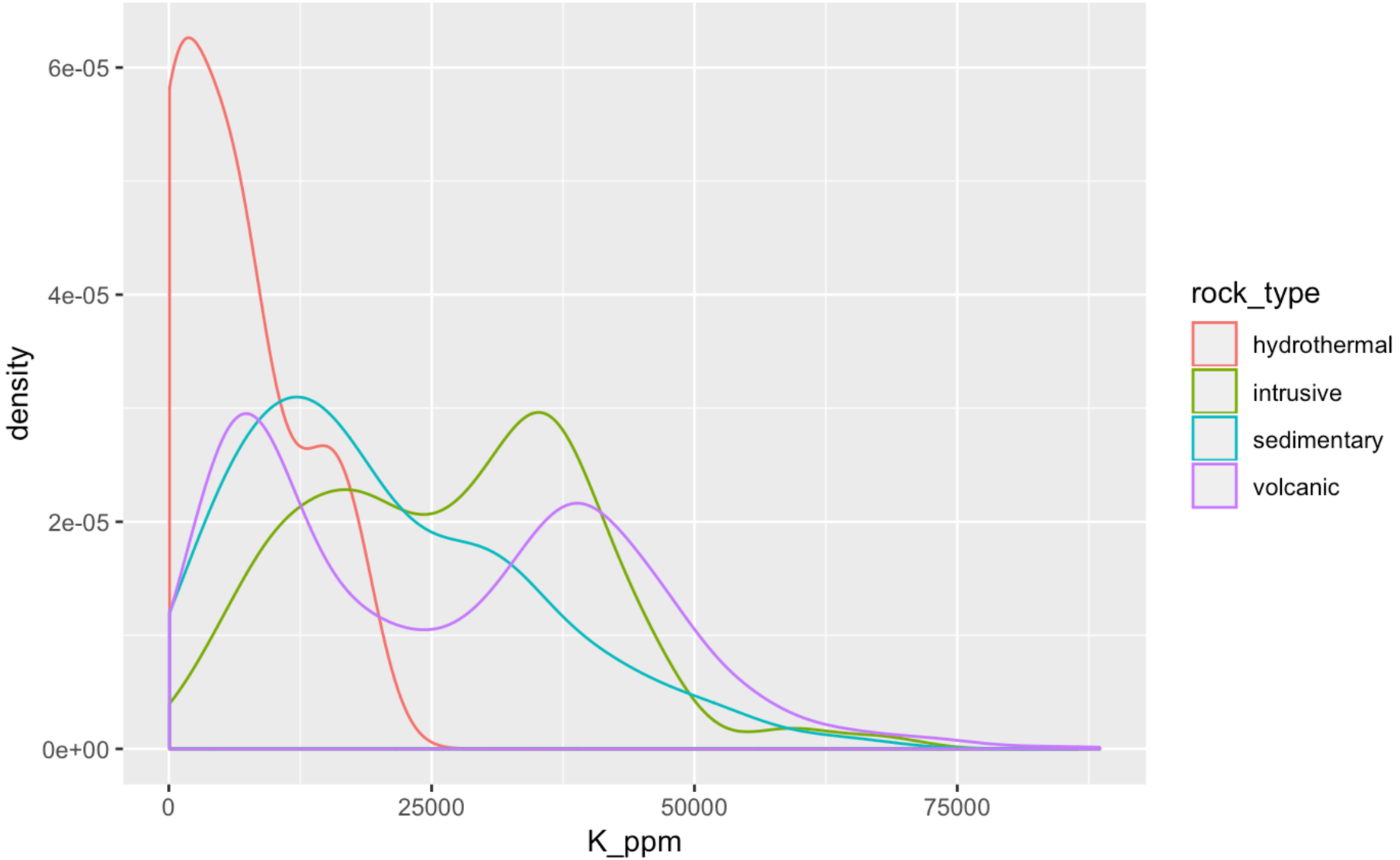

```
ggplot(data = warwick) +
  geom_density(mapping = aes(x = K_ppm))
```

Your Turn 5

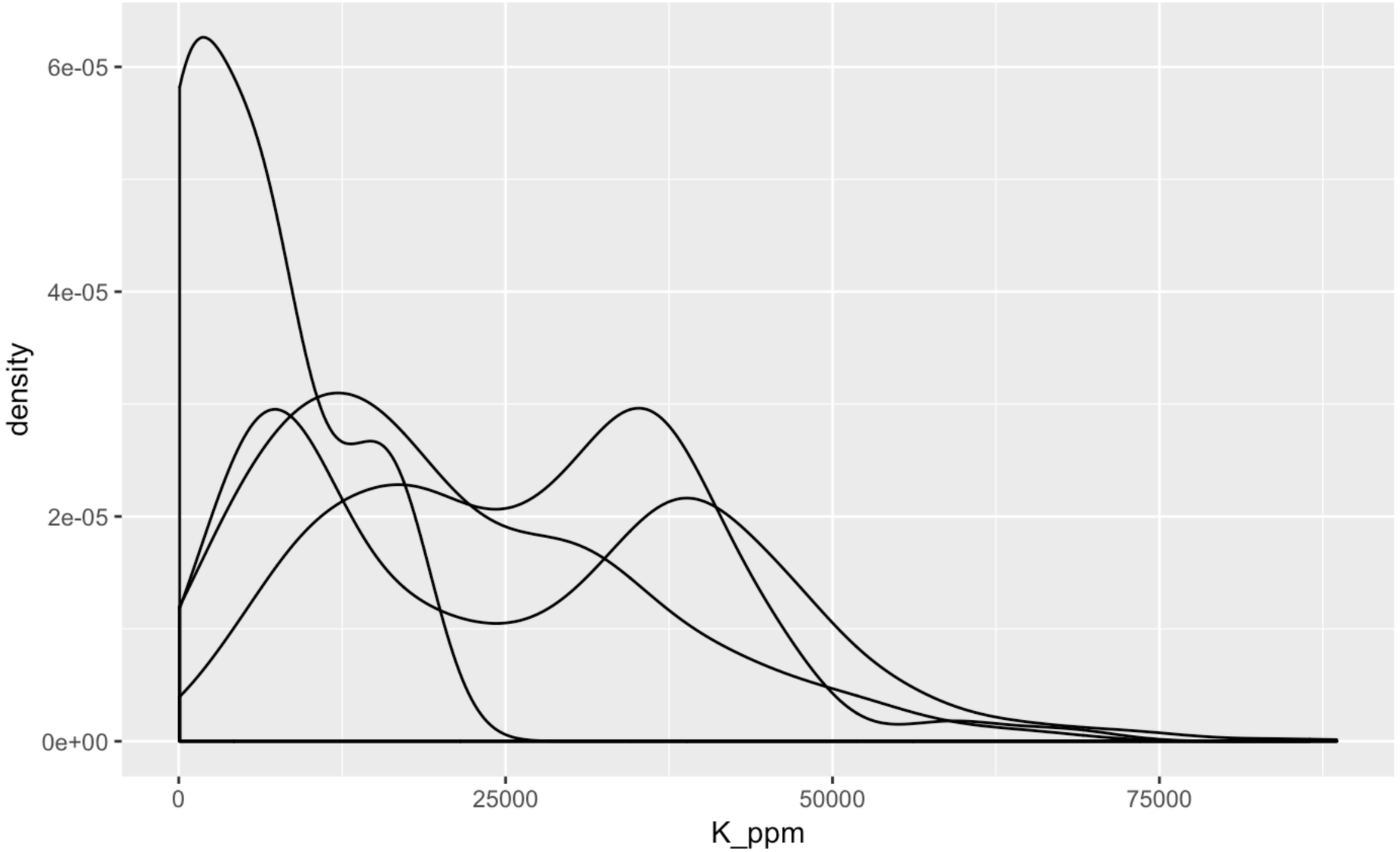
Make the density plot of **K_ppm** colored by **rock_type** below.
Use the cheatsheet. Try your best guess.



02:00




```
ggplot(data = warwick) +
  geom_density(mapping = aes(x = K_ppm, color = rock_type))
```



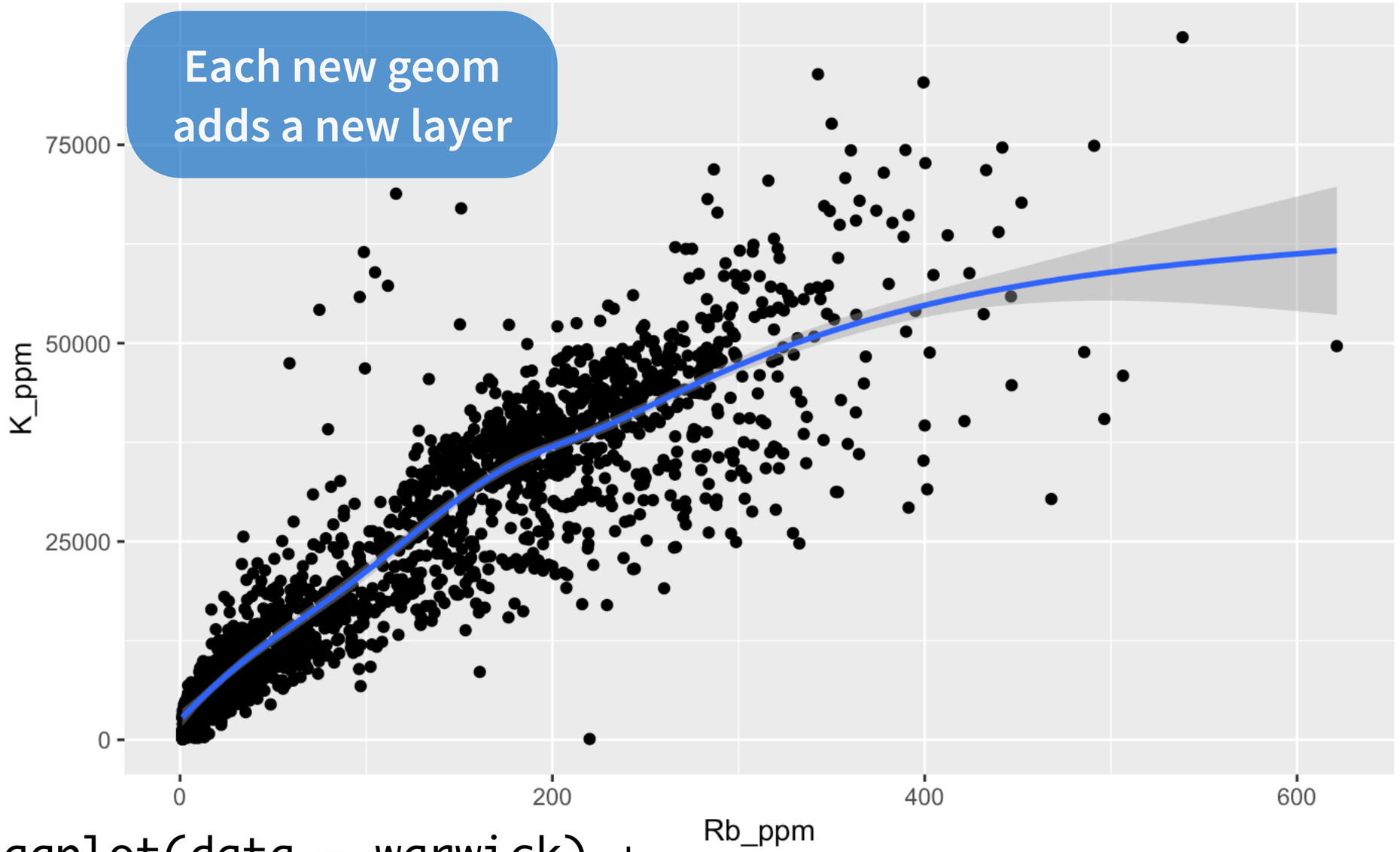

```
ggplot(data = warwick) +
  geom_density(mapping = aes(x = K_ppm, group = rock_type))
```

Your Turn 6

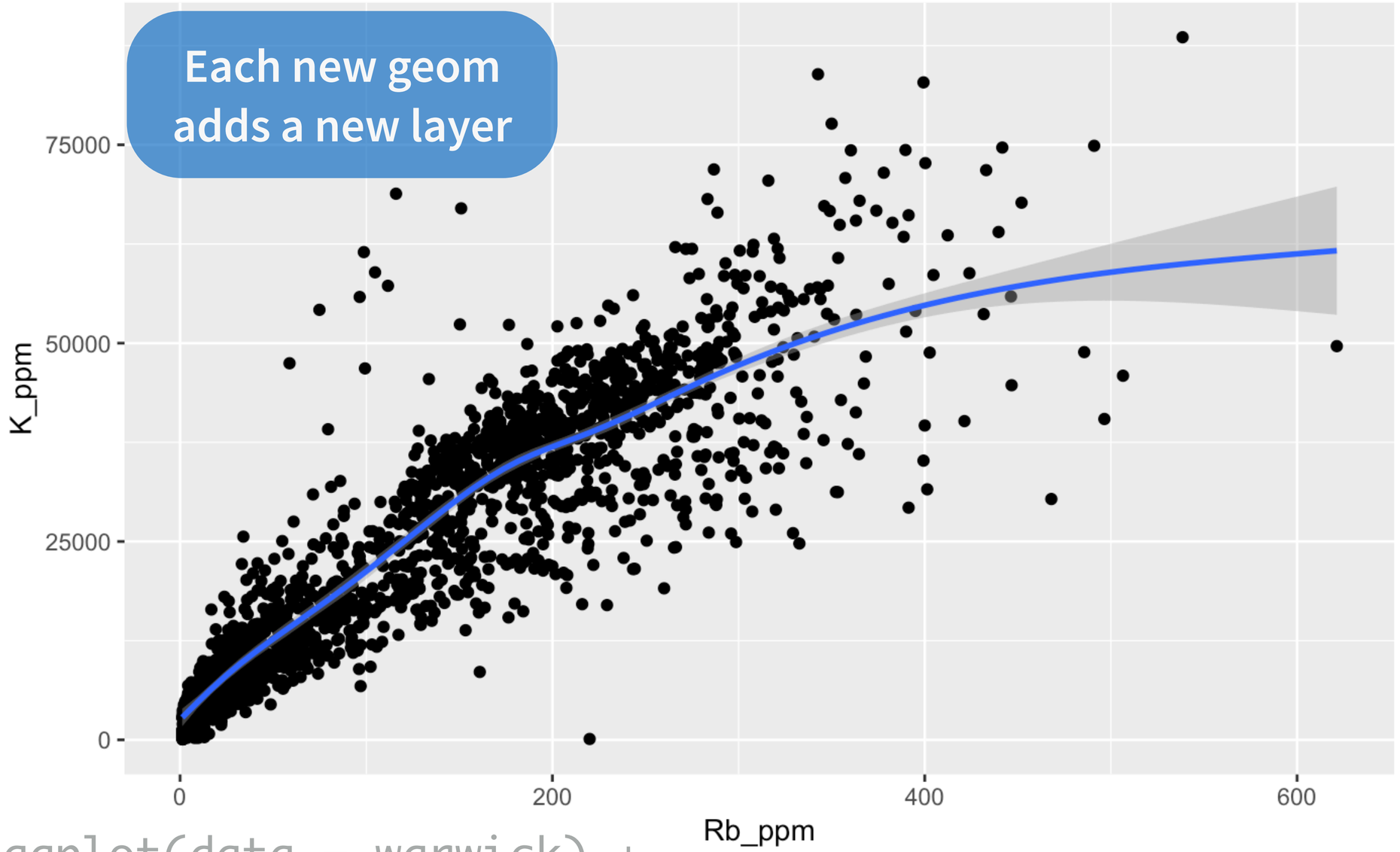
Predict what this code will do.

Then run it.

```
ggplot(warwick) +
  geom_point(aes(Rb_ppm, K_ppm)) +
  geom_smooth(aes(Rb_ppm, K_ppm))
```

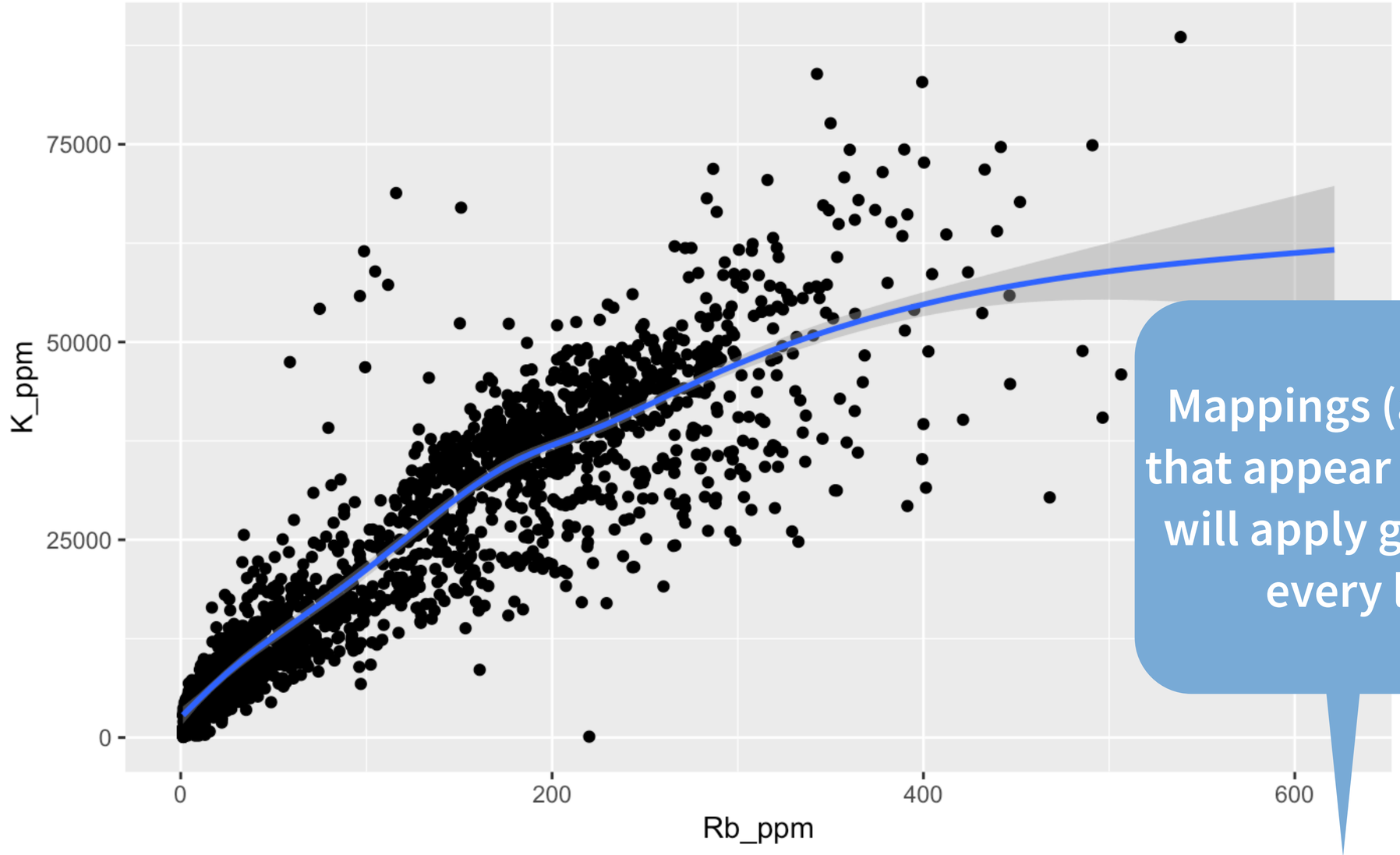



```
ggplot(data = warwick) +
  geom_point(mapping = aes(x = Rb_ppm, y = K_ppm)) +
  geom_smooth(mapping = aes(x = Rb_ppm, y = K_ppm))
```



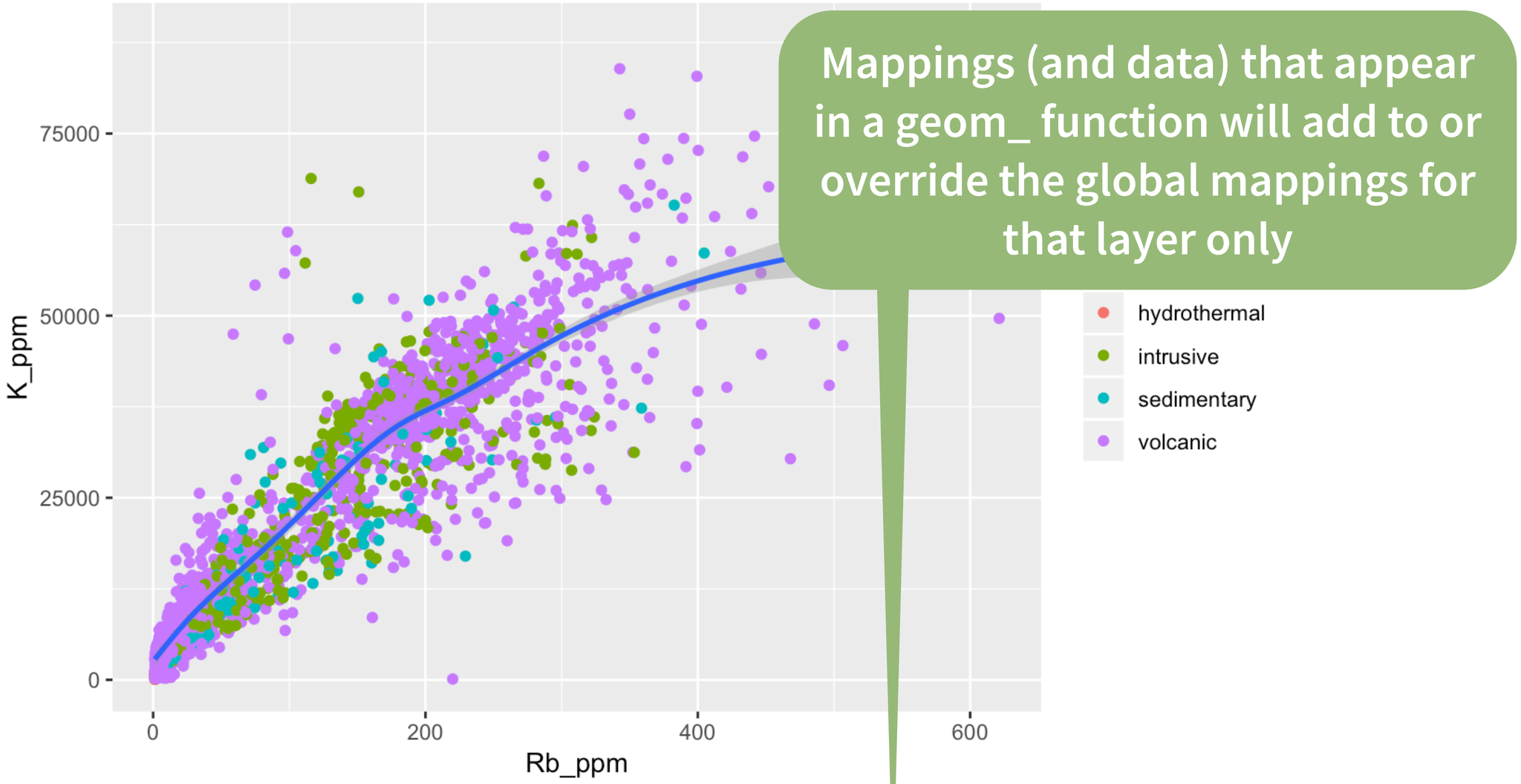

```
ggplot(data = warwick) +
  geom_point(mapping = aes(x = Rb_ppm, y = K_ppm)) +
  geom_smooth(mapping = aes(x = Rb_ppm, y = K_ppm))
```

global vs. local

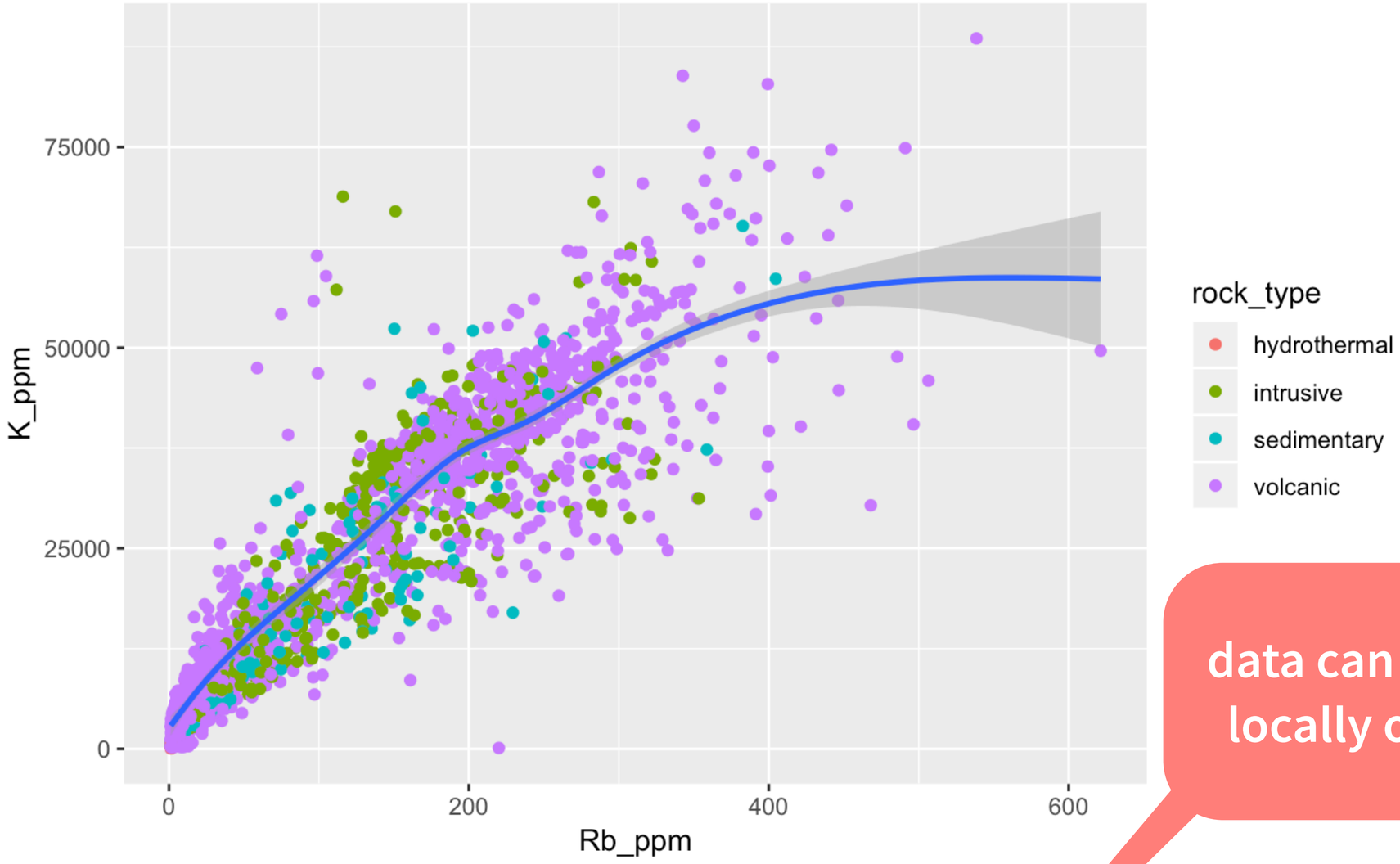


Mappings (and data) that appear in ggplot() will apply globally to every layer

```
ggplot(data = warwick, mapping = aes(x = Rb_ppm, y = K_ppm)) +
  geom_point() +
  geom_smooth()
```



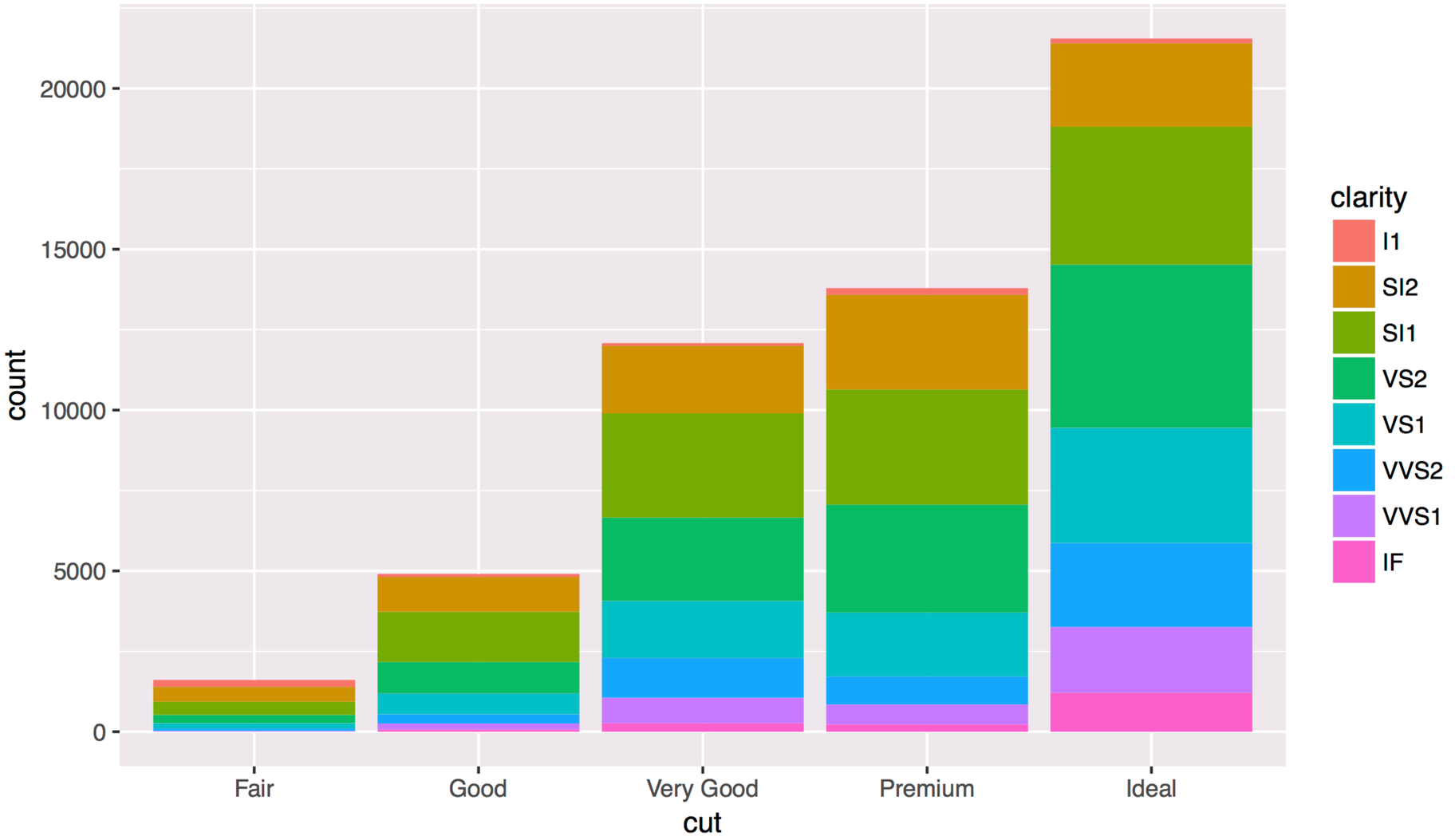

```
ggplot(data = warwick, mapping = aes(x = Rb_ppm, y = K_ppm)) +
  geom_point(mapping = aes(color = rock_type)) +
  geom_smooth()
```



data can also be set locally or globally

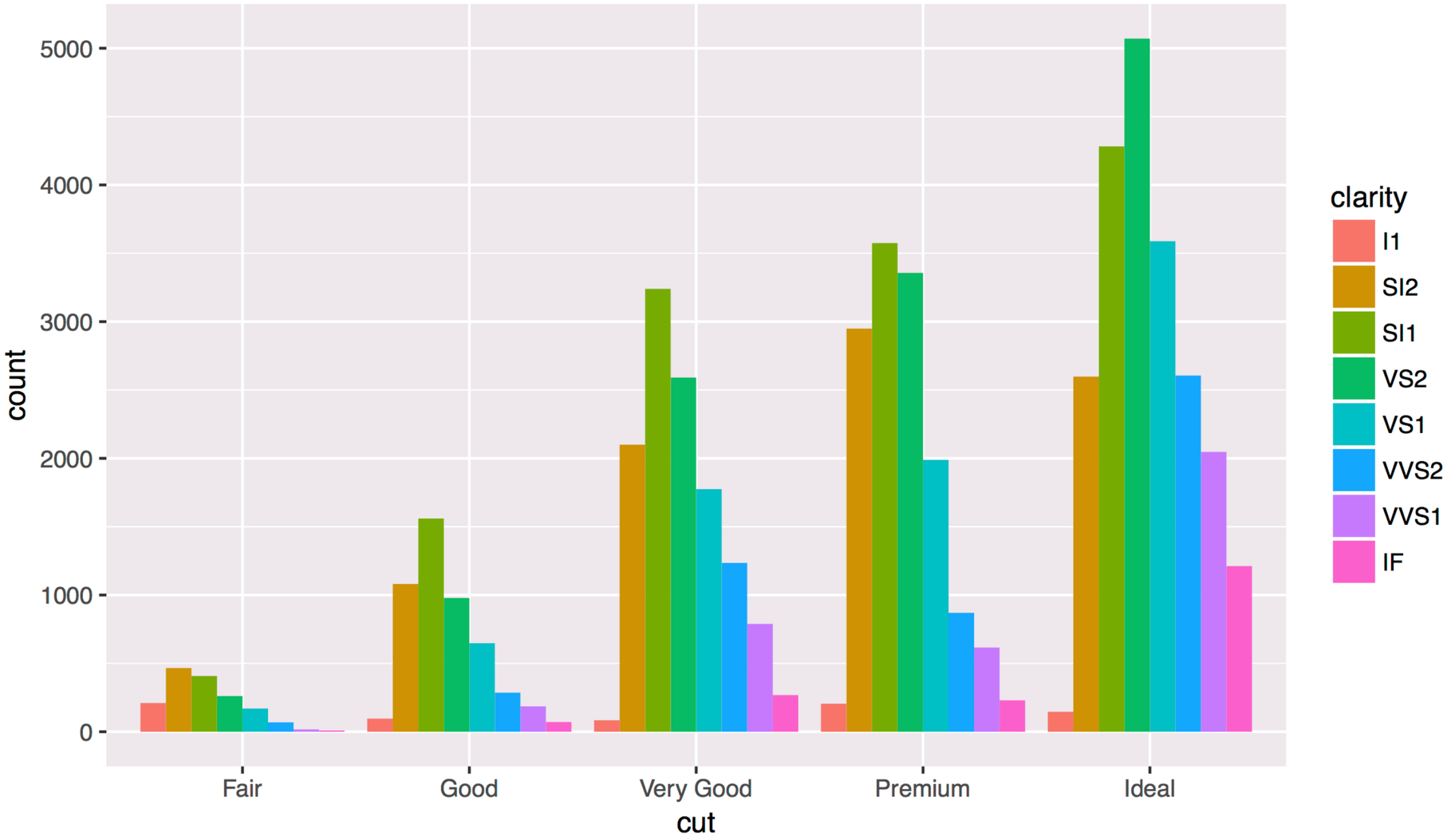
```
ggplot(data = warwick, mapping = aes(x = Rb_ppm, y = K_ppm)) +
  geom_point(mapping = aes(color = rock_type)) +
  geom_smooth(data = filter(warwick, rock_type == "volcanic"))
```

What else?


20000
15000
10000
5000
0
count
Fair
Good
Very Good
Premium
Ideal
cut
clarity
I1
SI2
SI1
VS2
VS1
VVS2
VVS1
IF

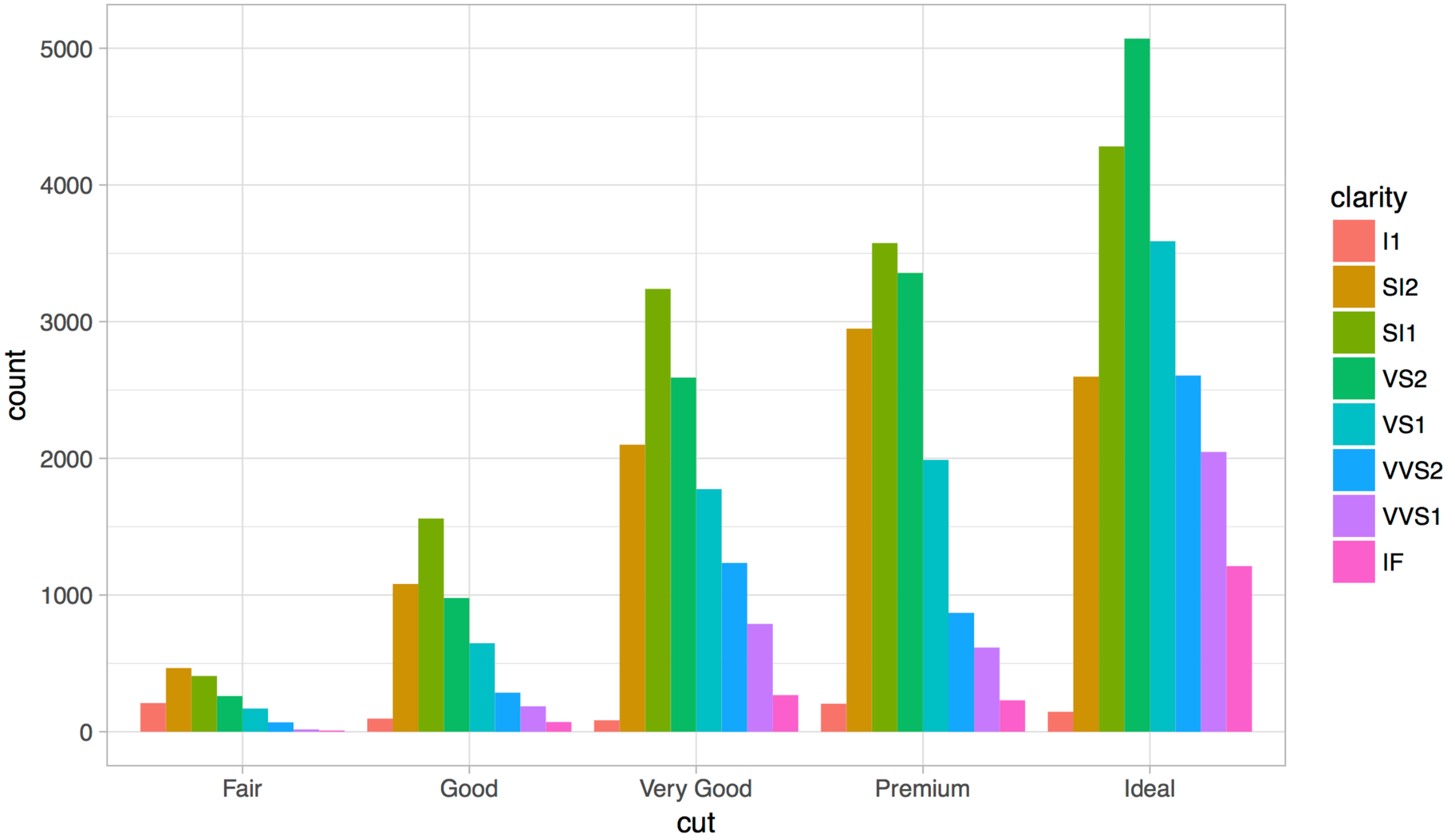
Position Adjustments

How overlapping objects are arranged



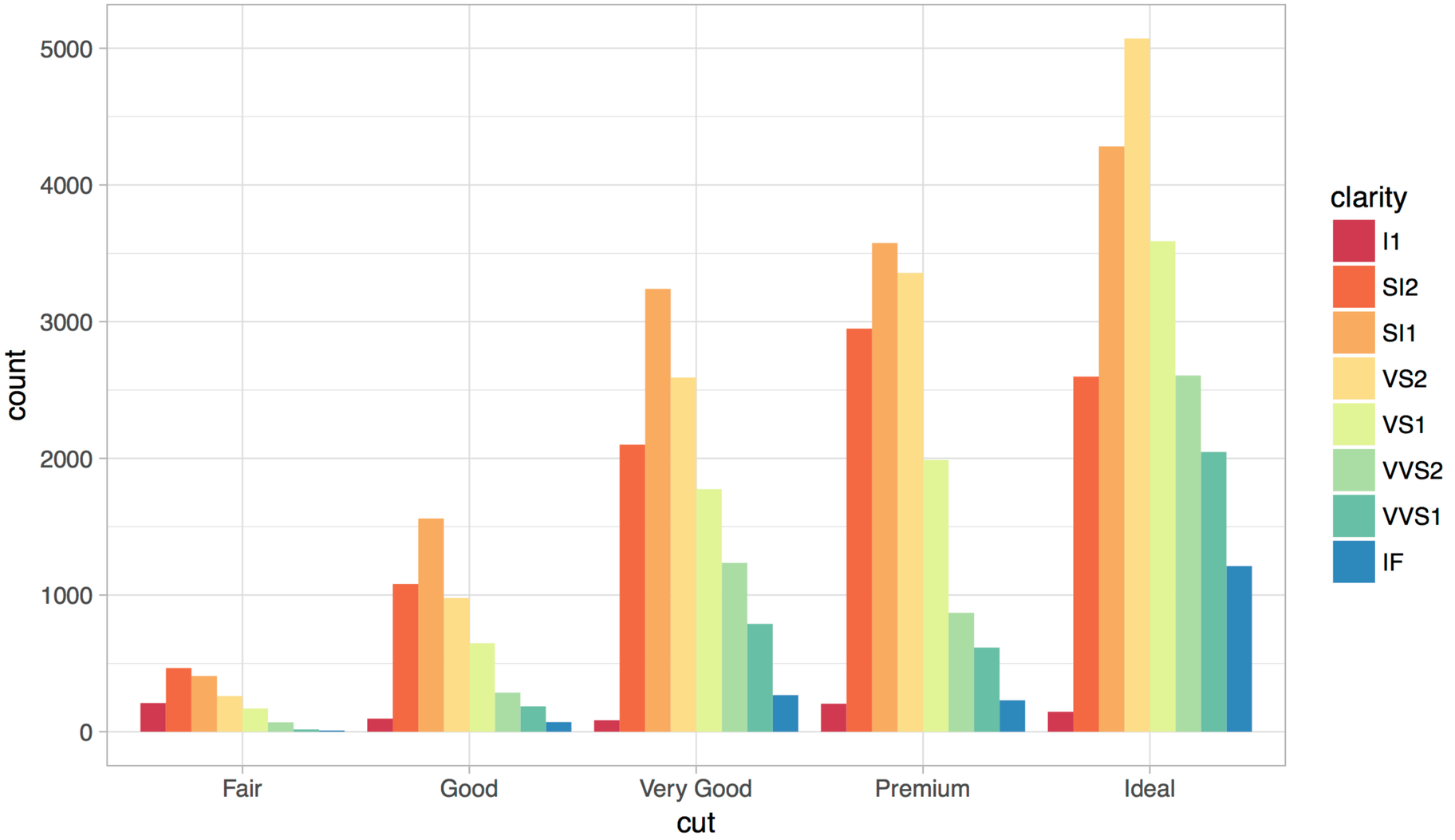
Themes

Visual appearance of non-data elements



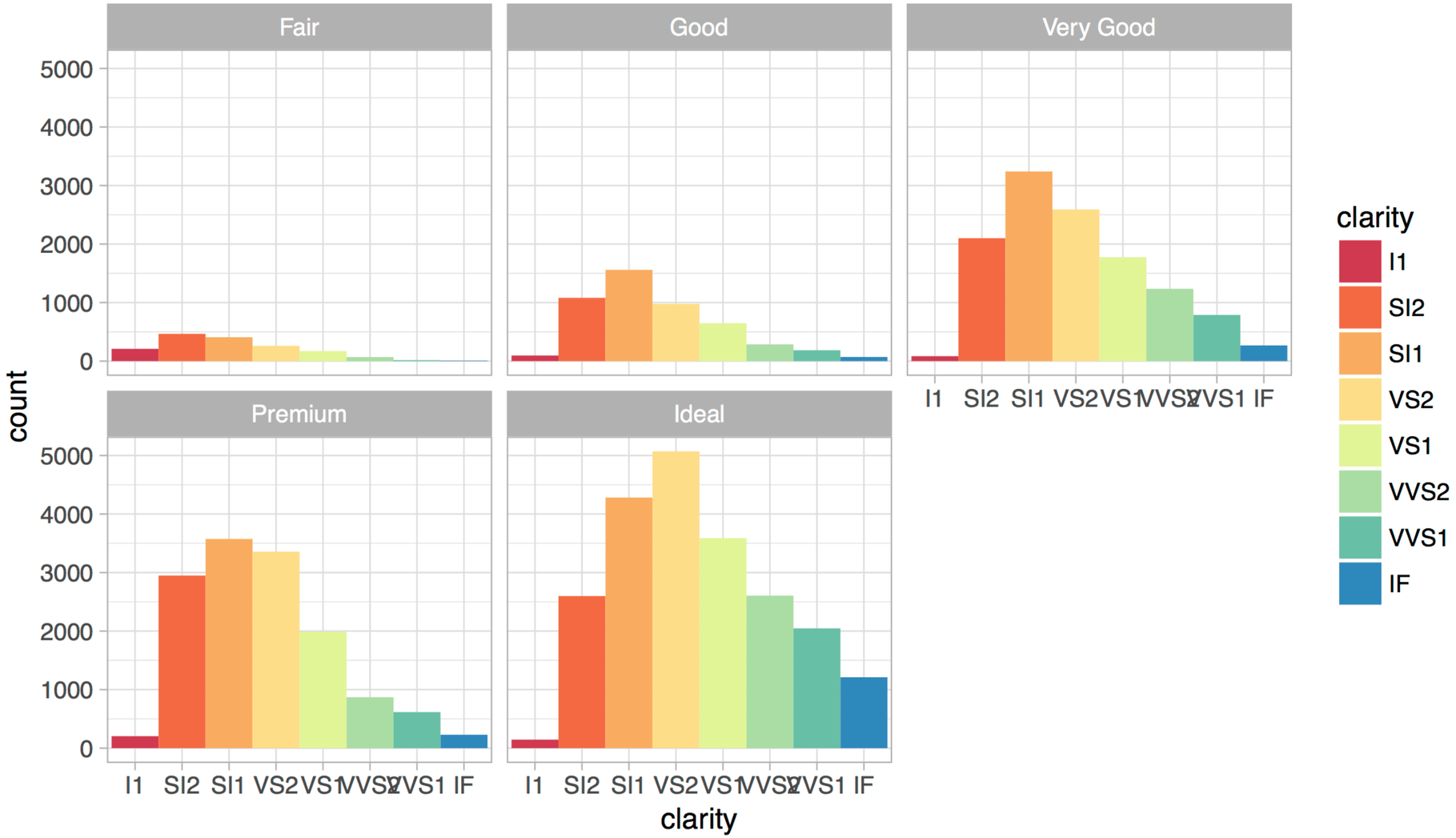
Scales

Customize color scales, other mappings

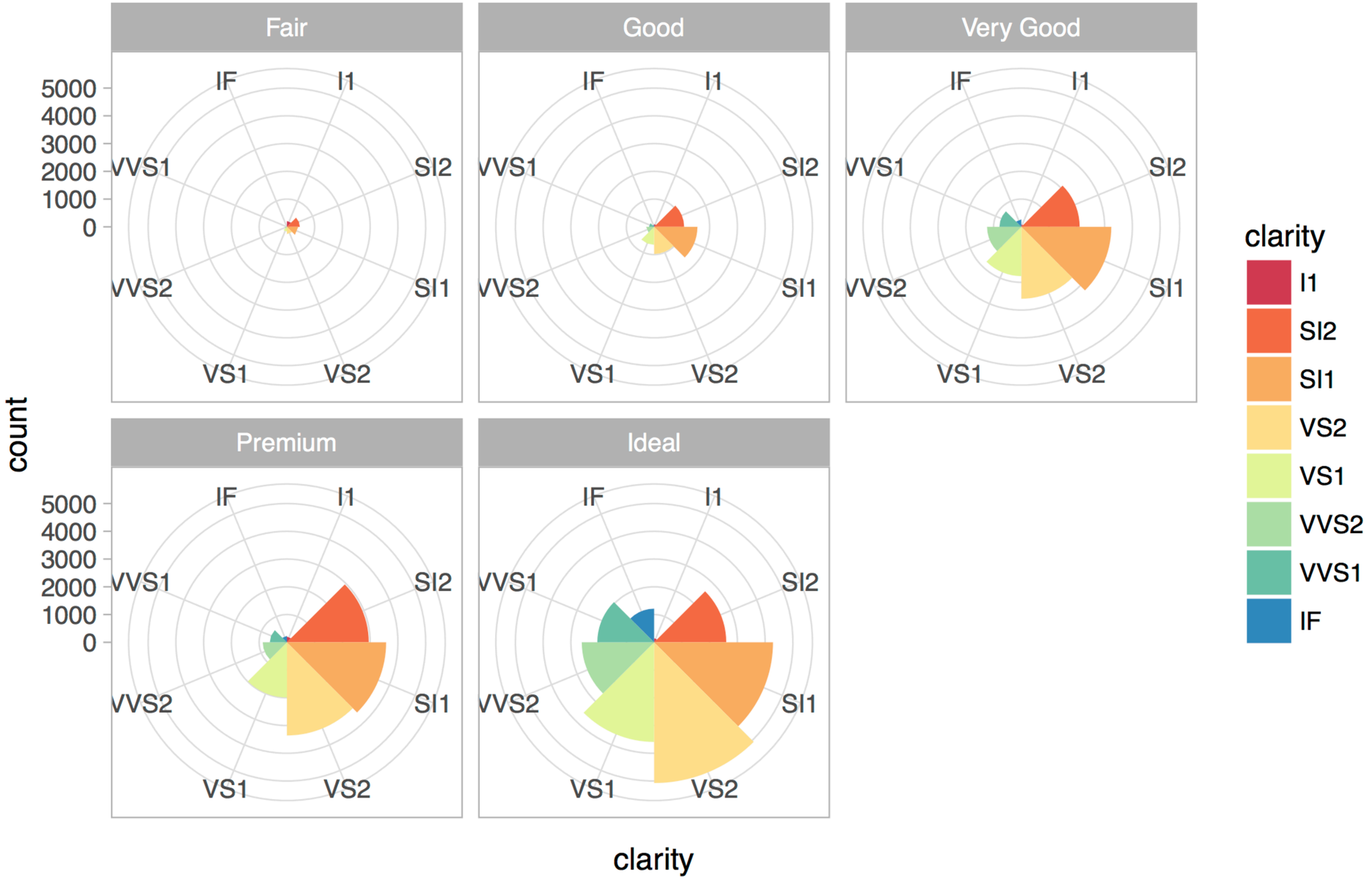


Facets

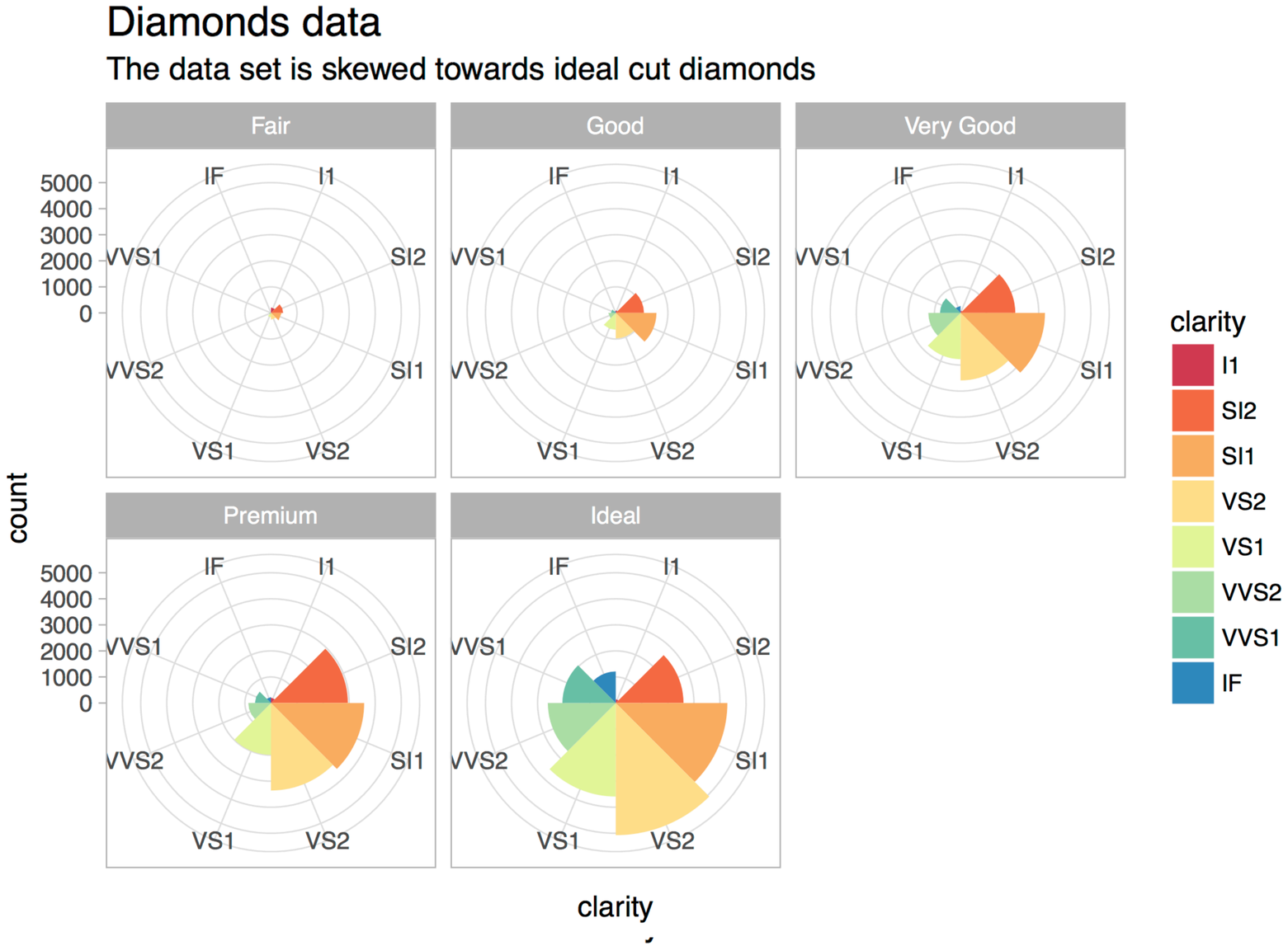
Subplots that display subsets of the data.



Coordinate systems



Titles and captions



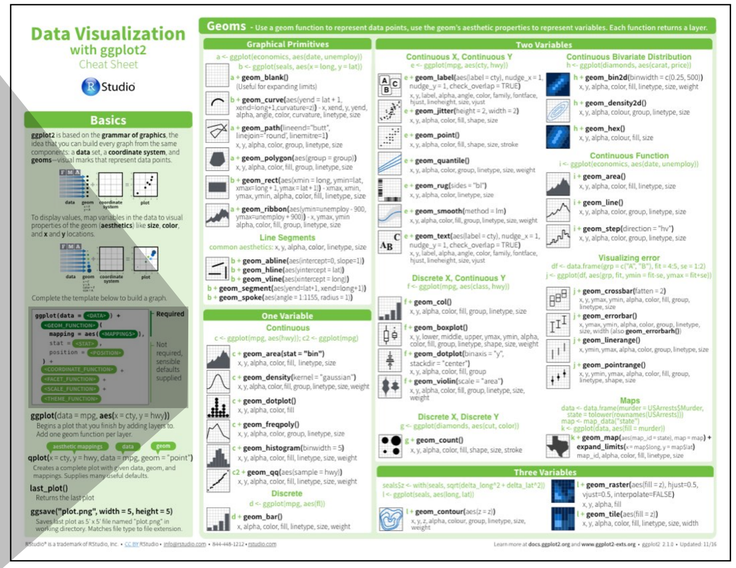
A ggplot2 template

Make any plot by filling in the parameters of this template

```
ggplot(data = <DATA>) +
  <GEOM_FUNCTION>(
    mapping = aes(<MAPPINGS>),
    stat = <STAT>,
    position = <POSITION>
  ) +
  <COORDINATE_FUNCTION> +
  <FACET_FUNCTION> +
  <SCALE_FUNCTION> +
  <THEME_FUNCTION>
```

Required

Not required, sensible defaults supplied



ggplot2

ggplot2.tidyverse.org

Create Elegant Data Visualisati ×

Garrett

ggplot2.tidyverse.org

ggplot2 part of the tidyverse

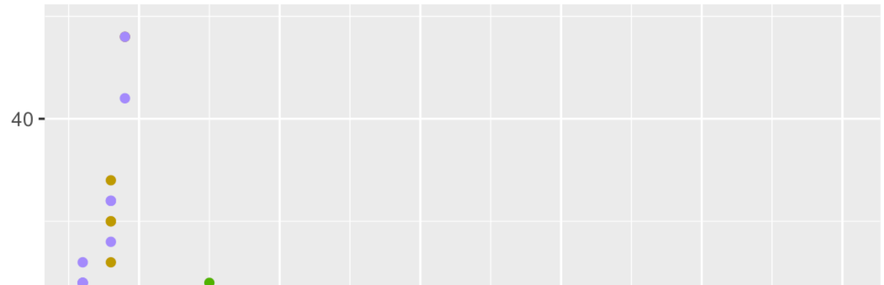
Reference Articles News

Usage

It's hard to succinctly describe how ggplot2 works because it embodies a deep philosophy of visualisation. However, in most cases you start with `ggplot()`, supply a dataset and aesthetic mapping (with `aes()`). You then add on layers (like `geom_point()` or `geom_histogram()`), scales (like `scale_colour_brewer()`), faceting specifications (like `facet_wrap()`) and coordinate systems (like `coord_flip()`).

```
library(ggplot2)

ggplot(mpg, aes(displ, hwy, colour = class)) +
  geom_point()
```



Links

Download from CRAN at https://cran.r-project.org/package=ggplot2

Browse source code at https://github.com/tidyverse/ggplot2

Report a bug at https://github.com/tidyverse/ggplot2/issues

Learn more at http://r4ds.had.co.nz/data-visualisation.html

License

GPL-2 | file LICENSE

Developers

Hadley Wickham

Author, maintainer

Visualize Data with