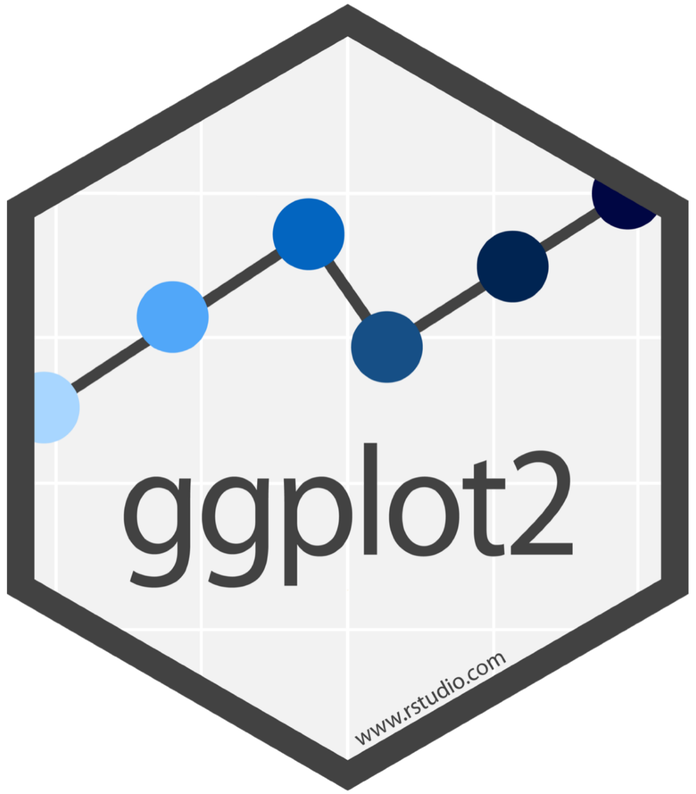
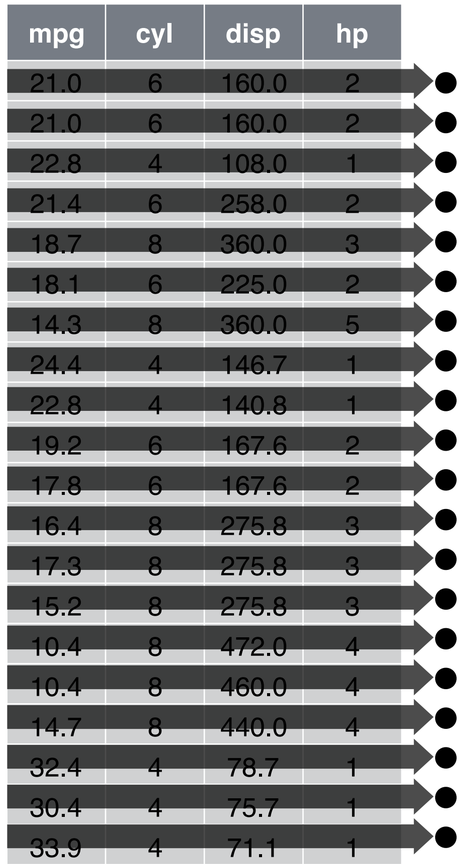
Visualize (More) Data with



To make a graph



mpg	cyl	disp	hp
21.0	6	160.0	2
21.0	6	160.0	2
22.8	4	108.0	1
21.4	6	258.0	2
18.7	8	360.0	3
18.1	6	225.0	2
14.3	8	360.0	5
24.4	4	146.7	1
22.8	4	140.8	1
19.2	6	167.6	2
17.8	6	167.6	2
16.4	8	275.8	3
17.3	8	275.8	3
15.2	8	275.8	3
10.4	8	472.0	4
10.4	8	460.0	4
14.7	8	440.0	4
32.4	4	78.7	1
30.4	4	75.7	1
33.9	4	71.1	1

data **geom**

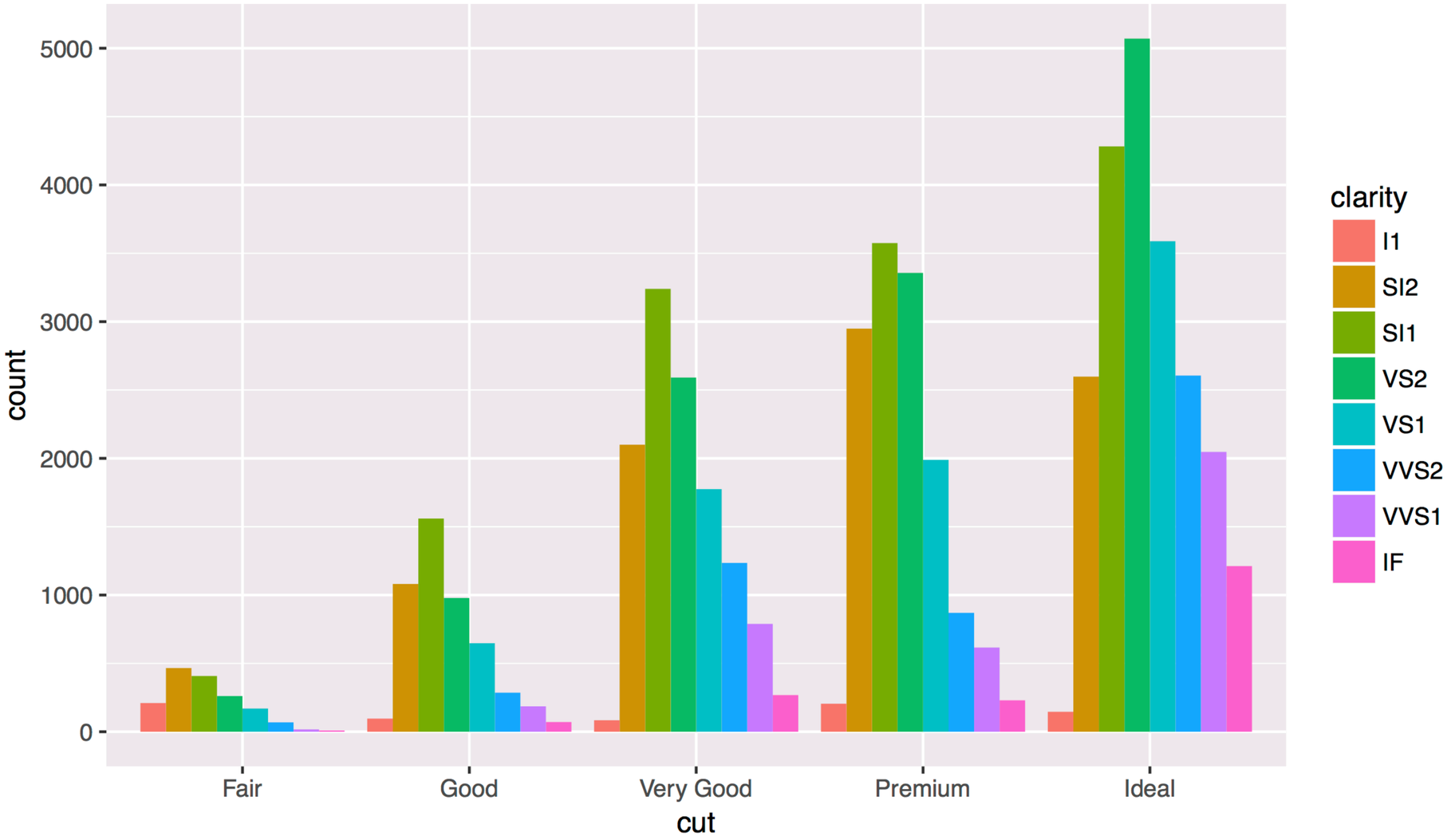
1. Pick a **data** set

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

2. Choose a **geom** to display cases

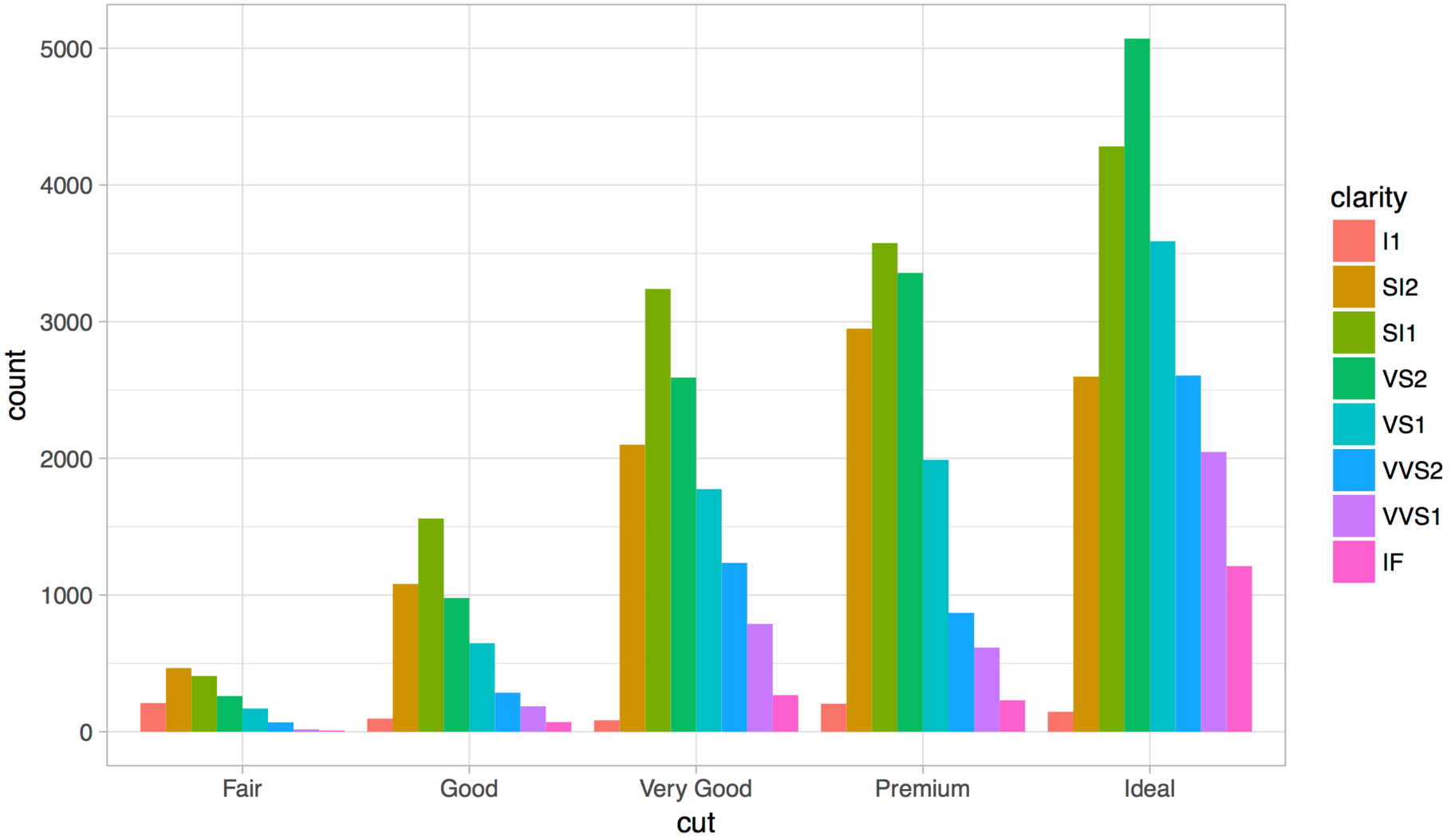
Themes

Visual appearance of non-data elements



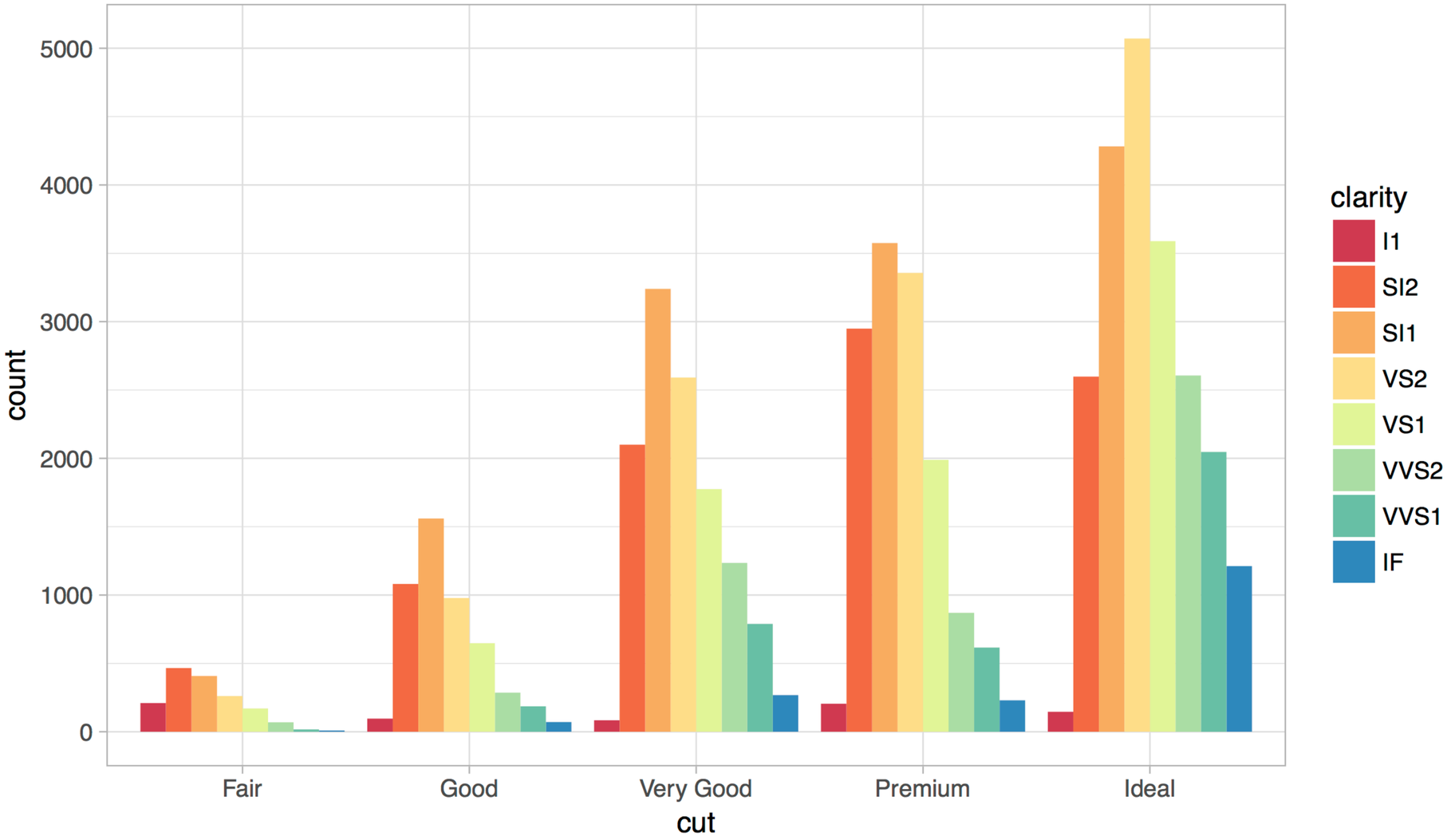
Themes

Visual appearance of non-data elements



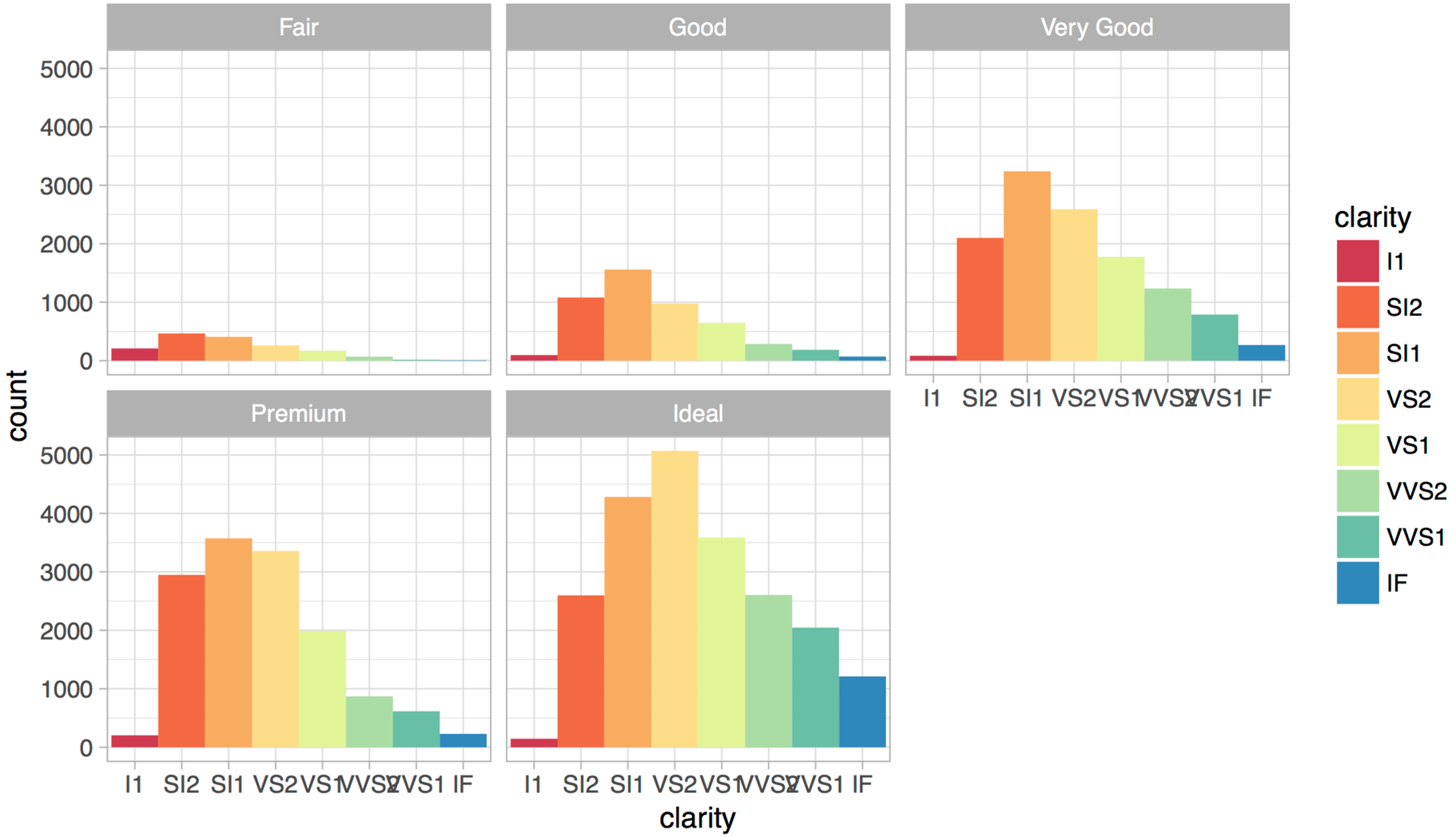
Scales

Customize color scales, other mappings

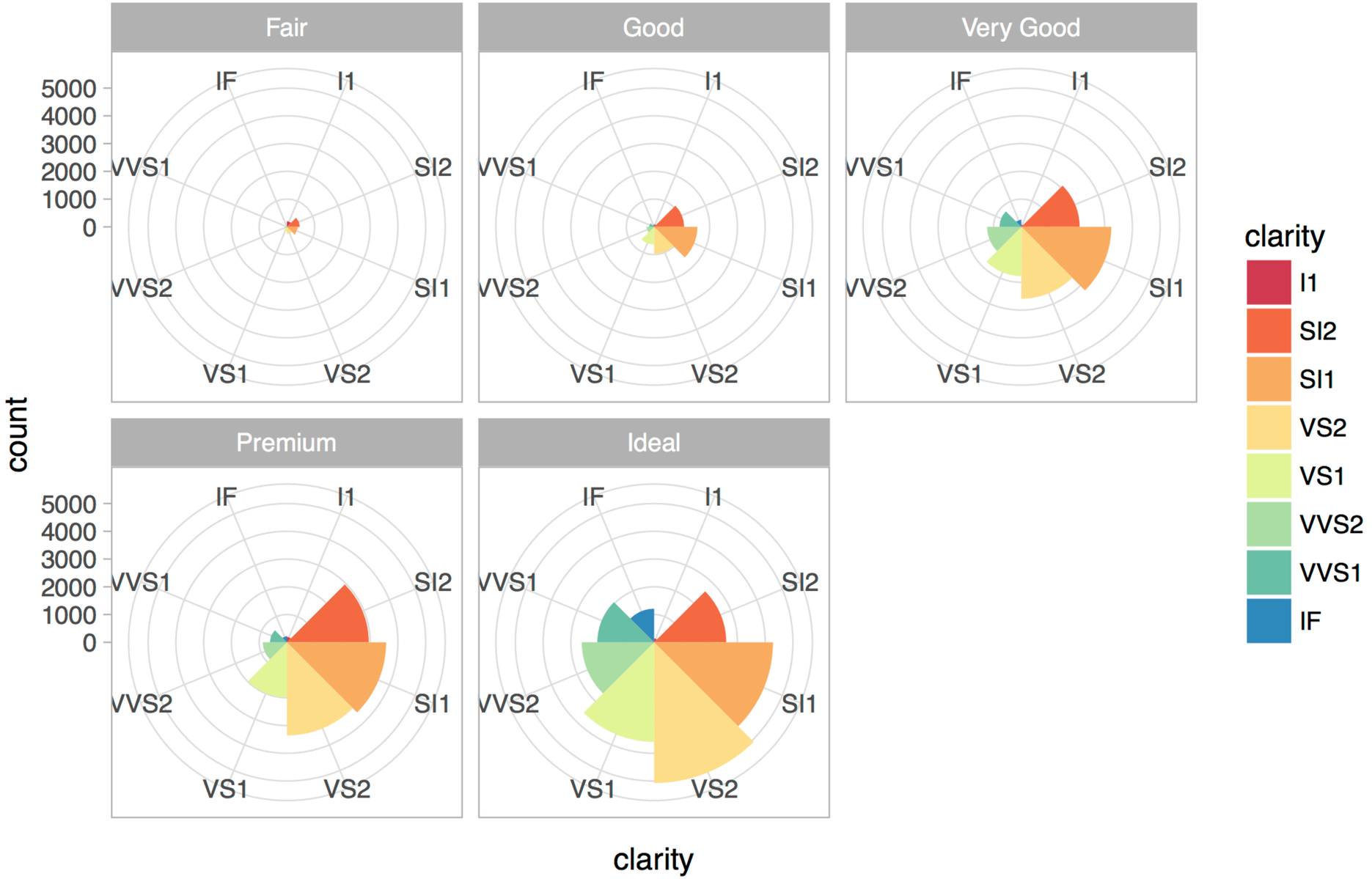


Facets

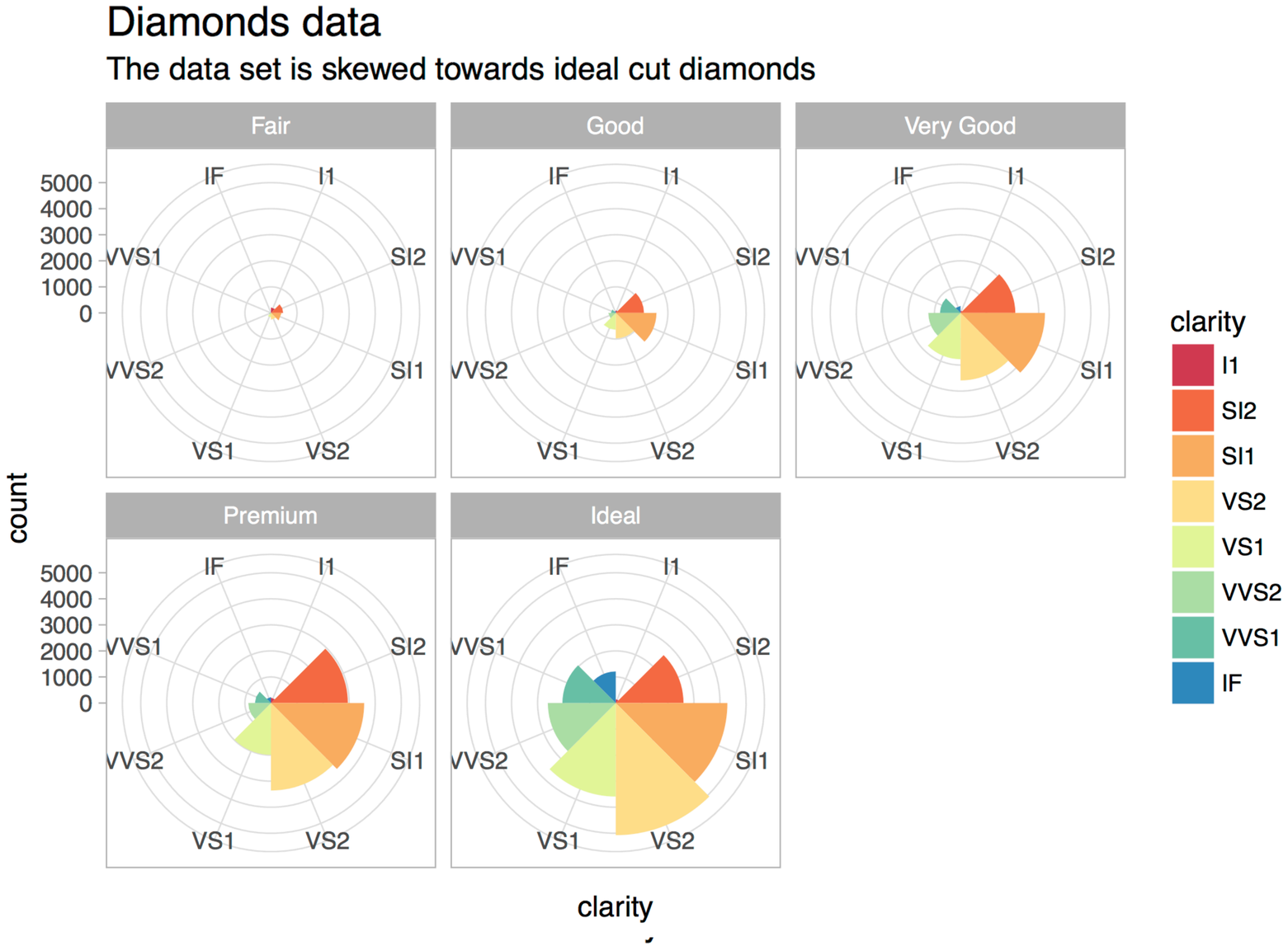
Subplots that display subsets of the data.



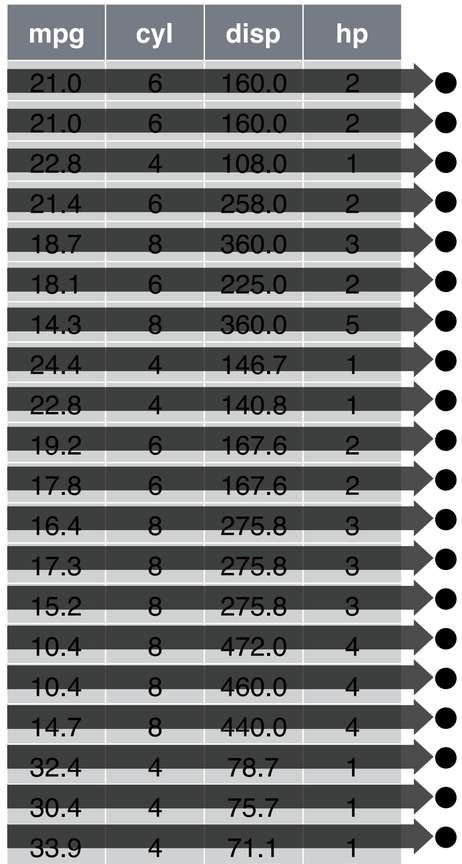
Coordinate systems



Titles and captions



To make a graph



mpg	cyl	disp	hp
21.0	6	160.0	2
21.0	6	160.0	2
22.8	4	108.0	1
21.4	6	258.0	2
18.7	8	360.0	3
18.1	6	225.0	2
14.3	8	360.0	5
24.4	4	146.7	1
22.8	4	140.8	1
19.2	6	167.6	2
17.8	6	167.6	2
16.4	8	275.8	3
17.3	8	275.8	3
15.2	8	275.8	3
10.4	8	472.0	4
10.4	8	460.0	4
14.7	8	440.0	4
32.4	4	78.7	1
30.4	4	75.7	1
33.9	4	71.1	1

data **geom**

1. Pick a **data** set

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

2. Choose a **geom** to display cases

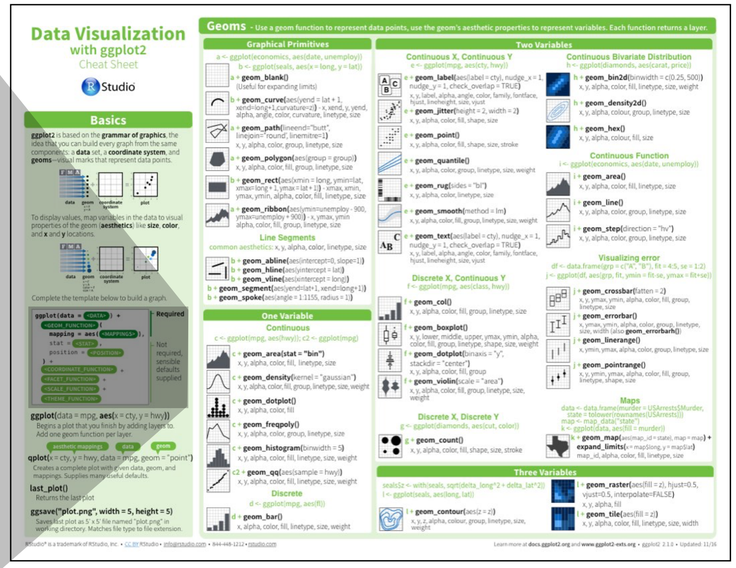
A ggplot2 template

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

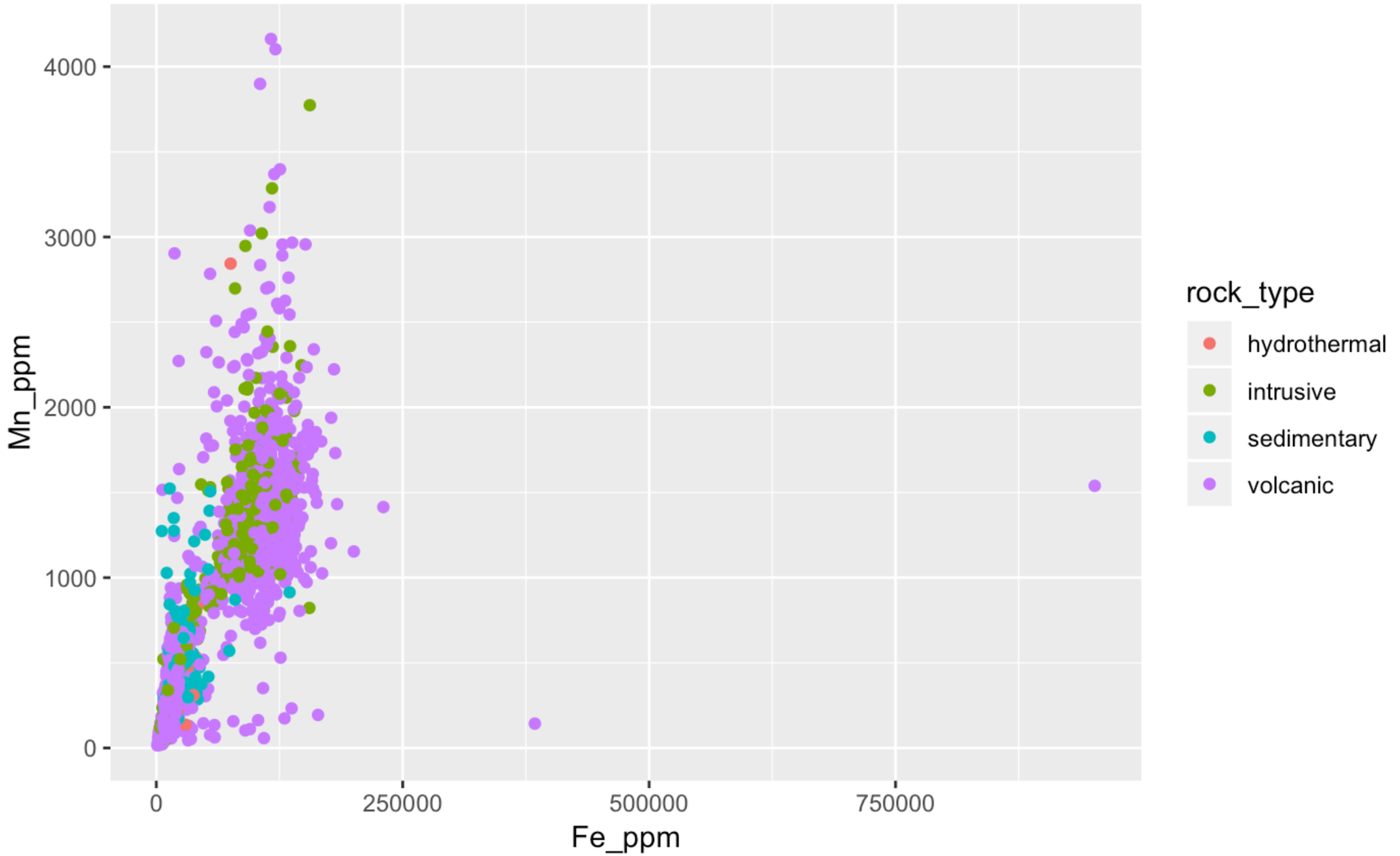


warwick

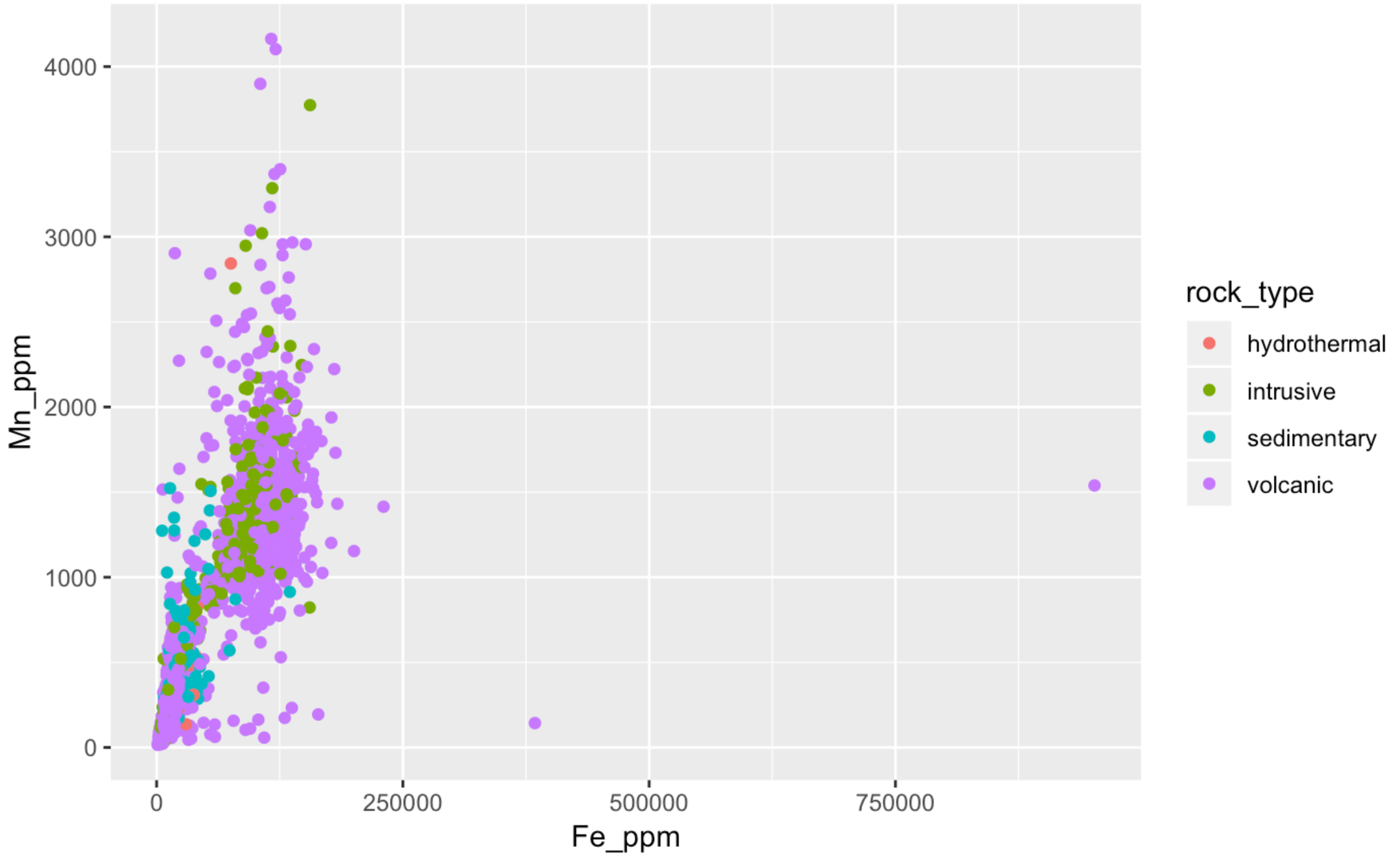
Rock sample data from the Wentworth area, Nova Scotia

```
warwick
```

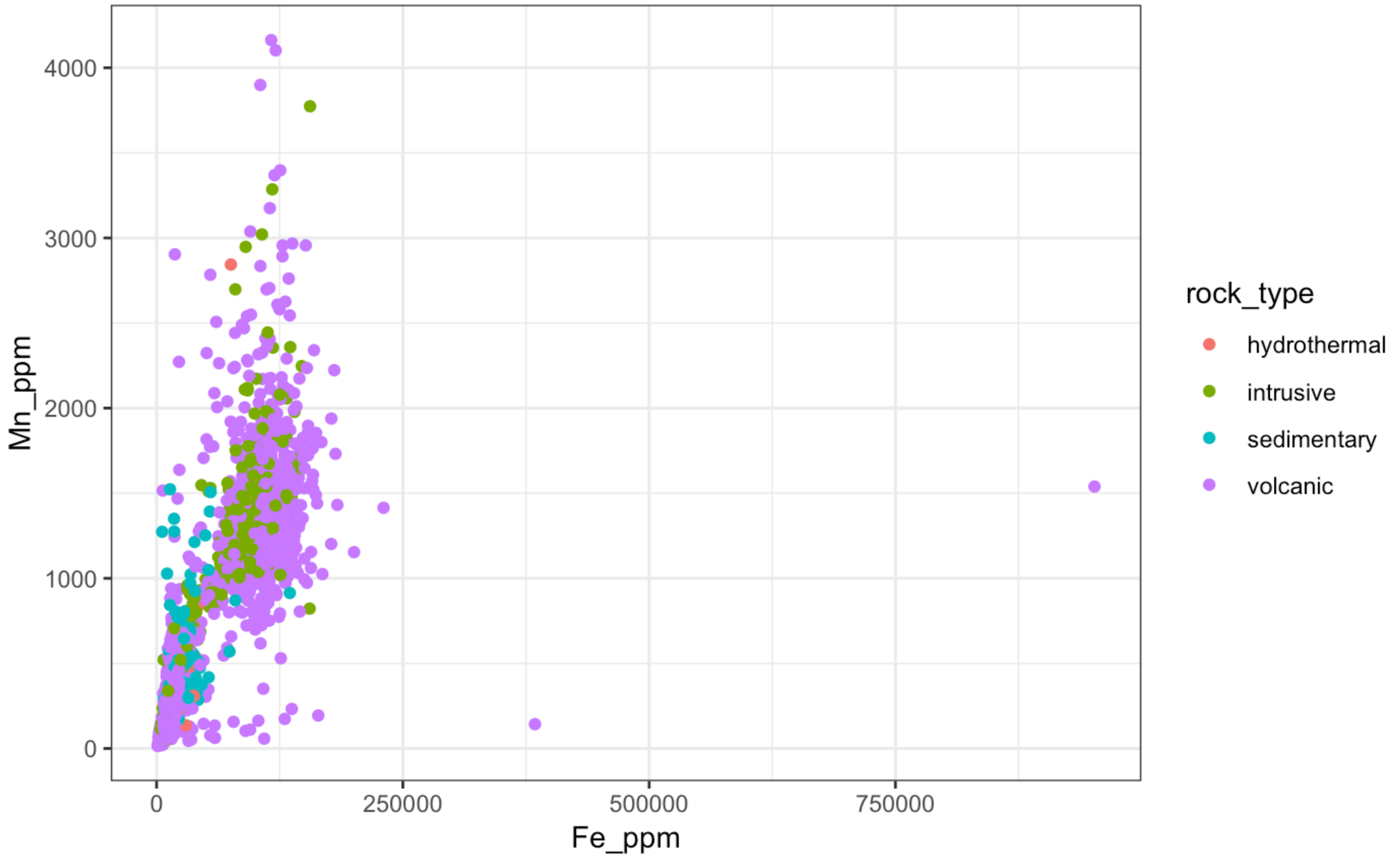
Themes



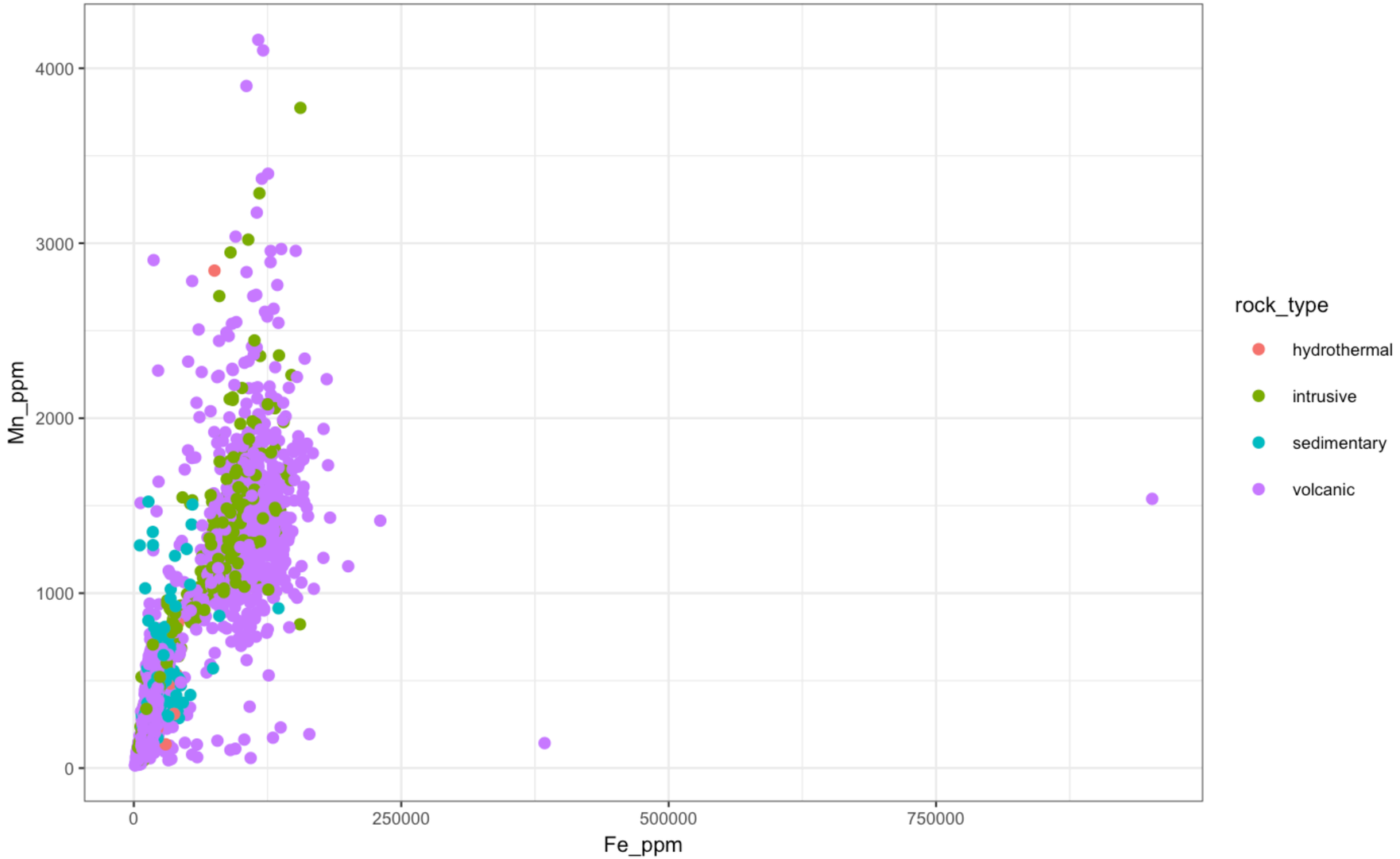

```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type))
```




```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type)) +
  theme_grey()
```




```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type)) +
  theme_bw()
```

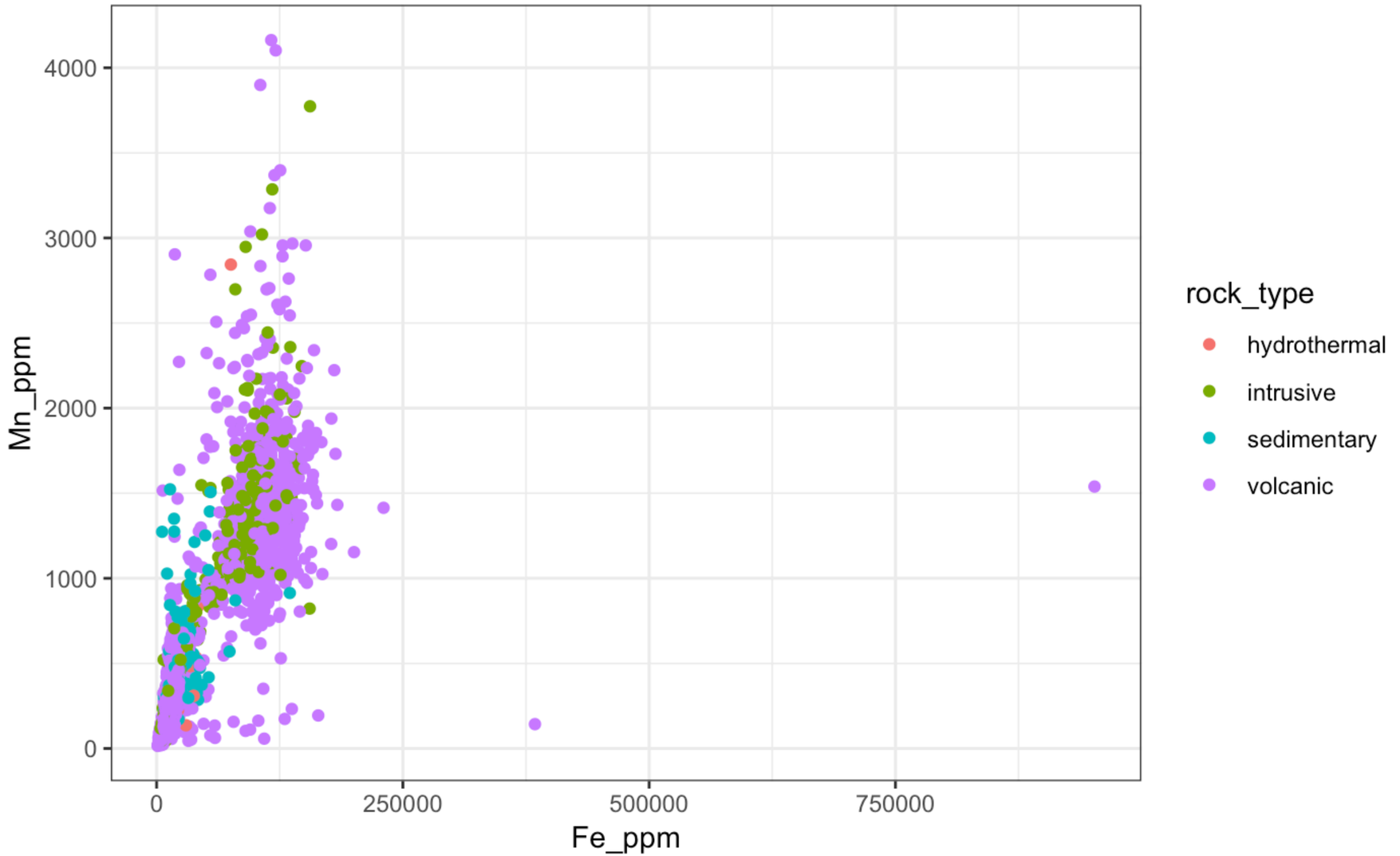



```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type)) +
  theme_bw(8)
```

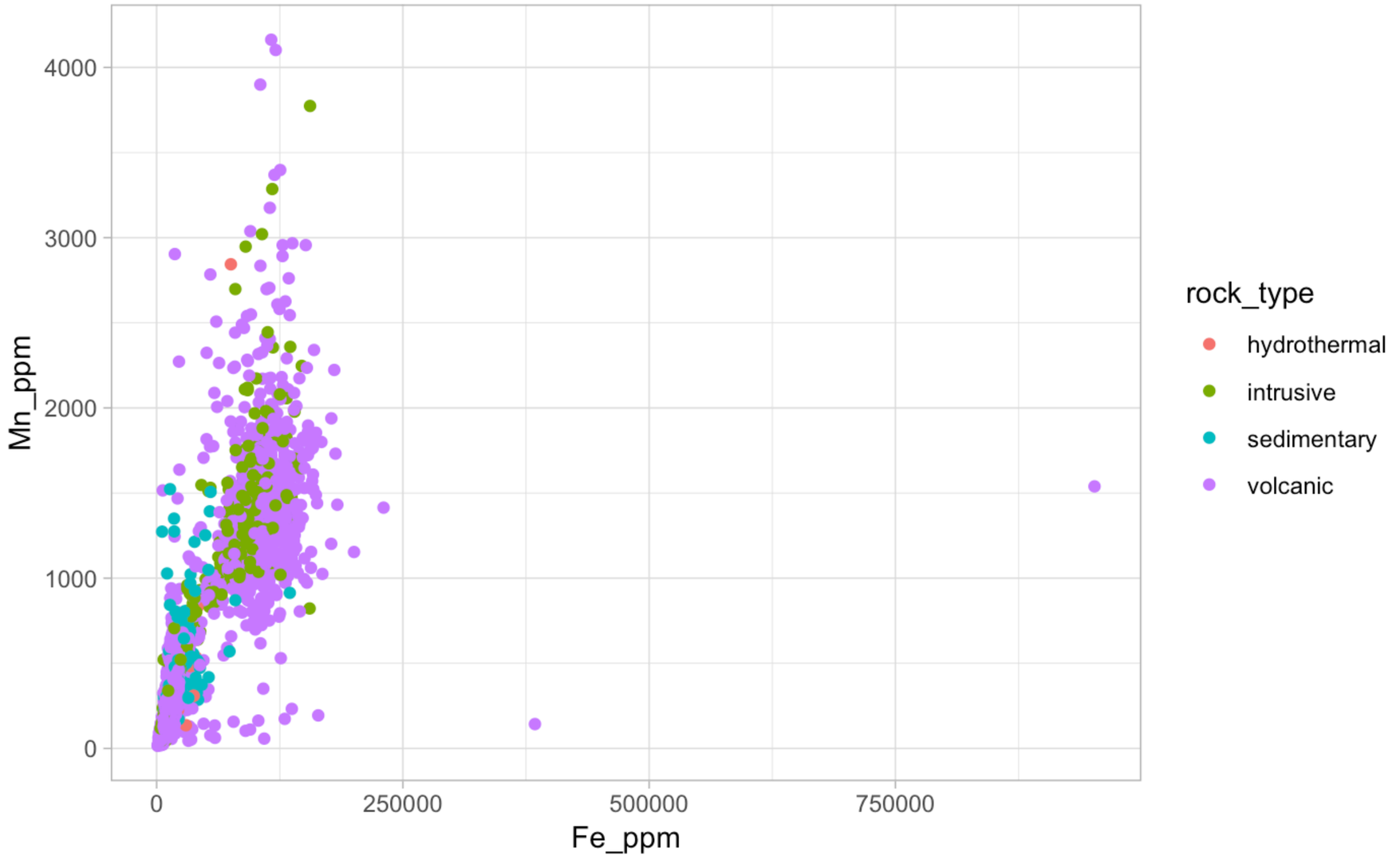
Your Turn 1

Modify the following code to use one of ggplot2's built-in themes with a base font size of 10 points. Experiment! (hint: they all start with `theme_`

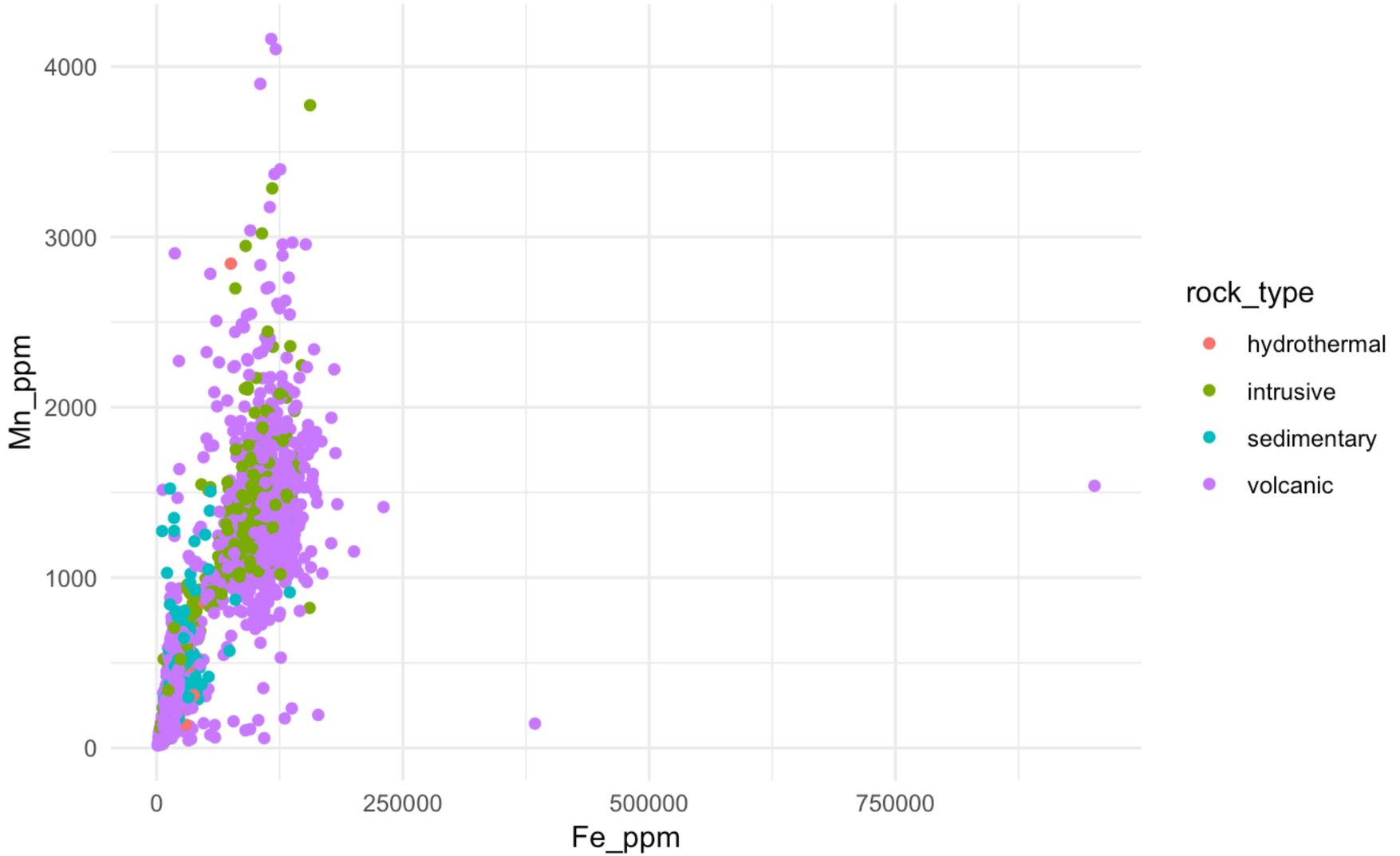
```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type))
```



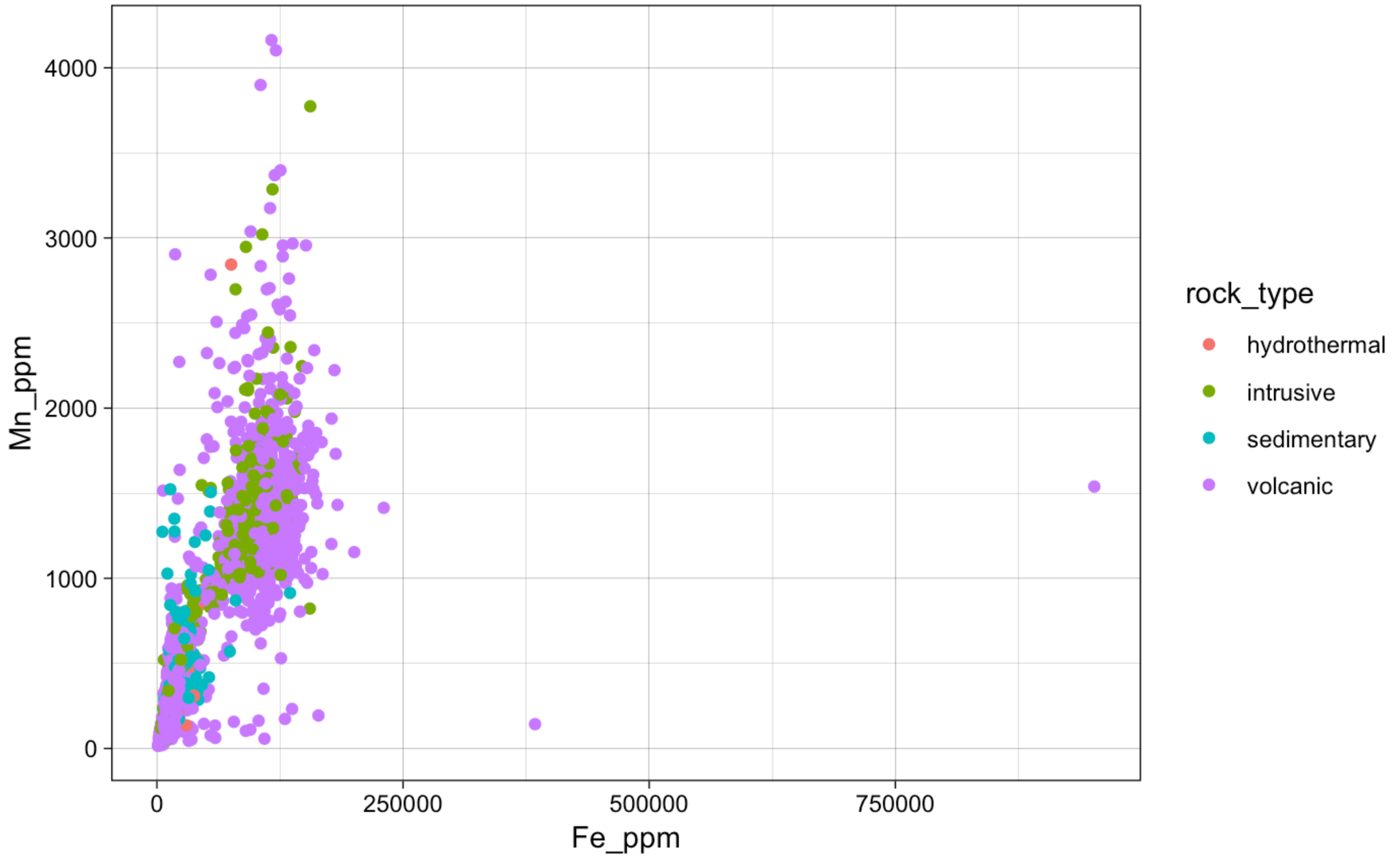

```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type)) +
  theme_bw(10)
```



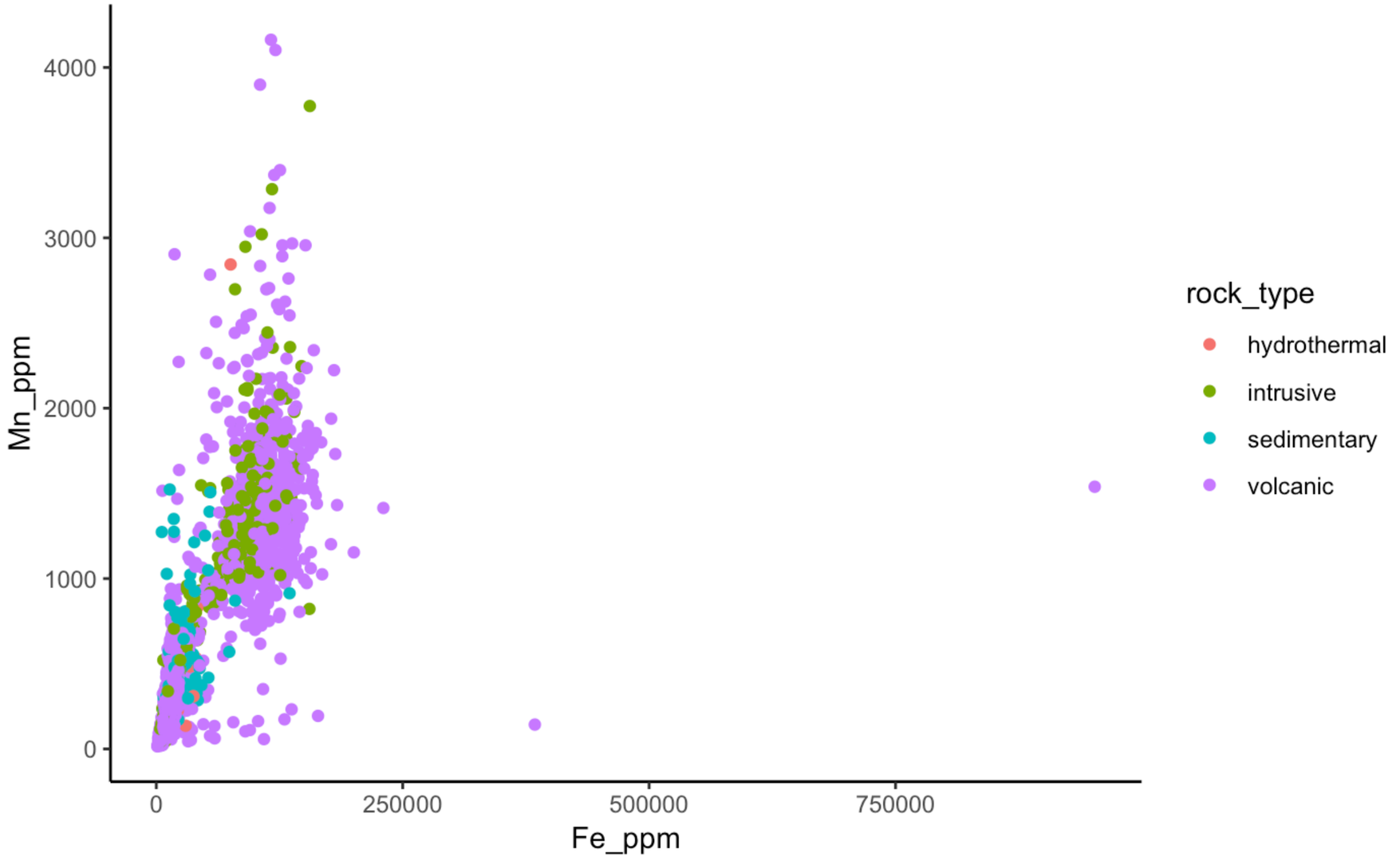

```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type)) +
  theme_light(10)
```



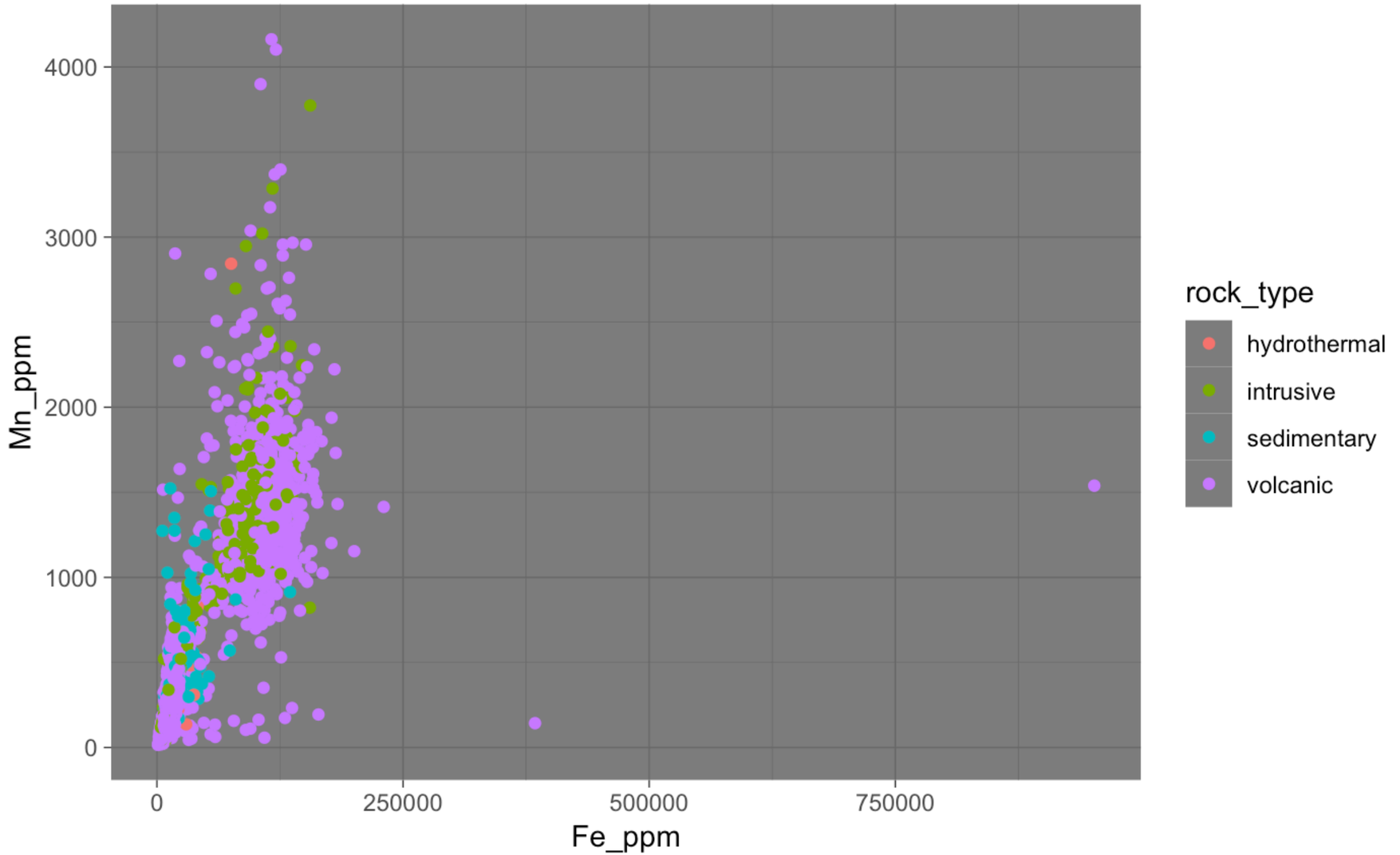

```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type)) +
  theme_minimal(10)
```



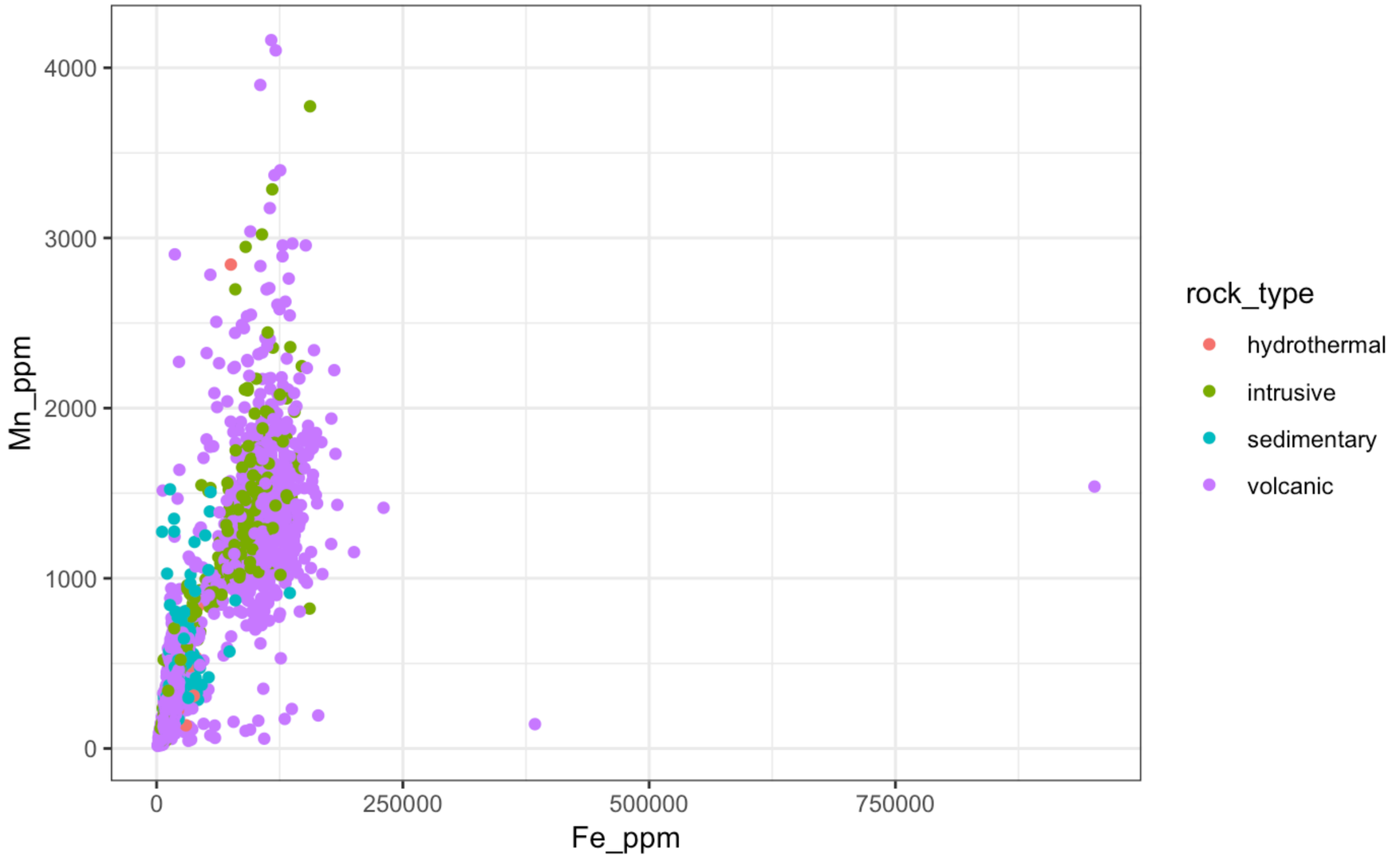

```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type)) +
  theme_linedraw(10)
```



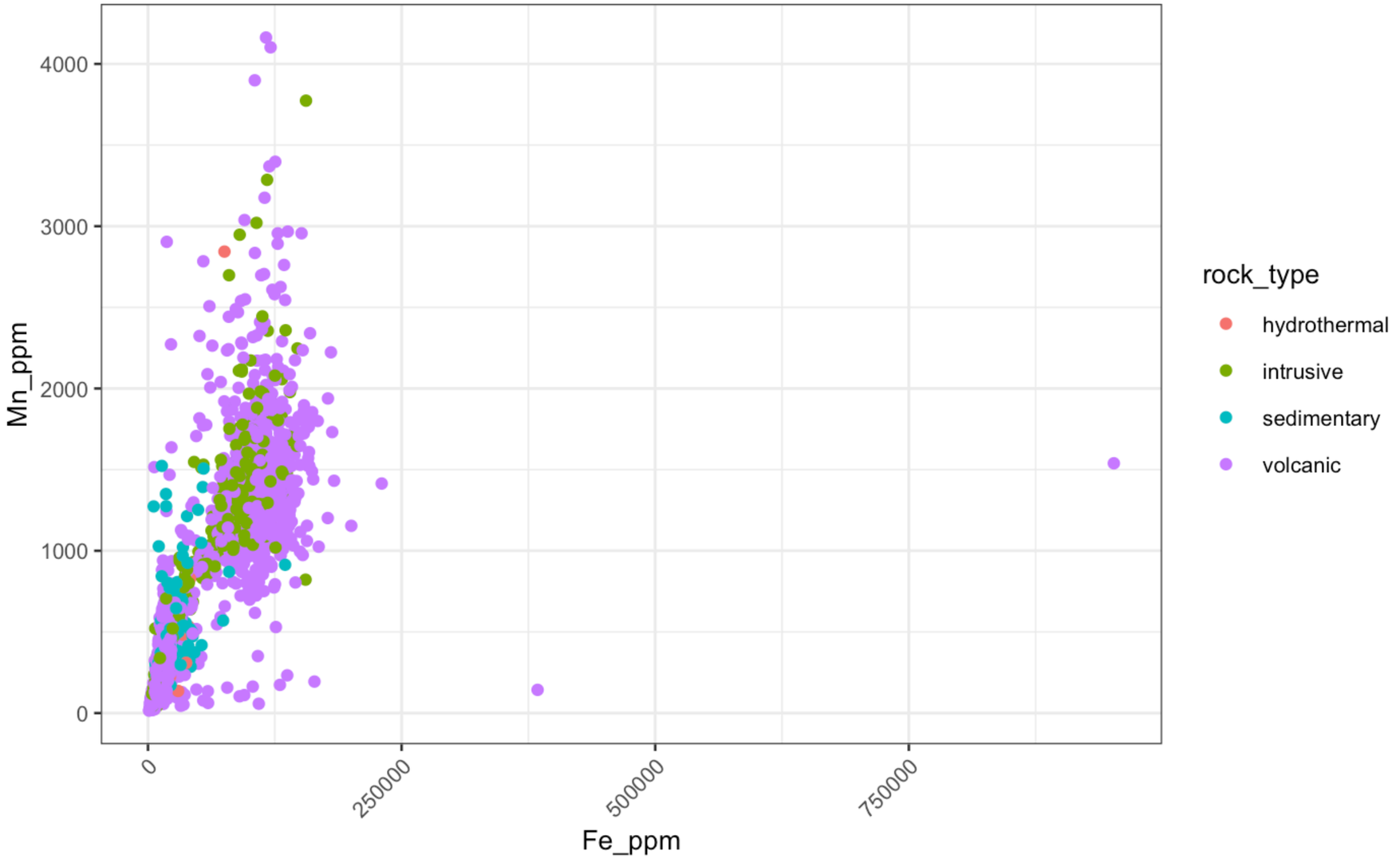

```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type)) +
  theme_classic(10)
```




```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type)) +
  theme_dark(10)
```

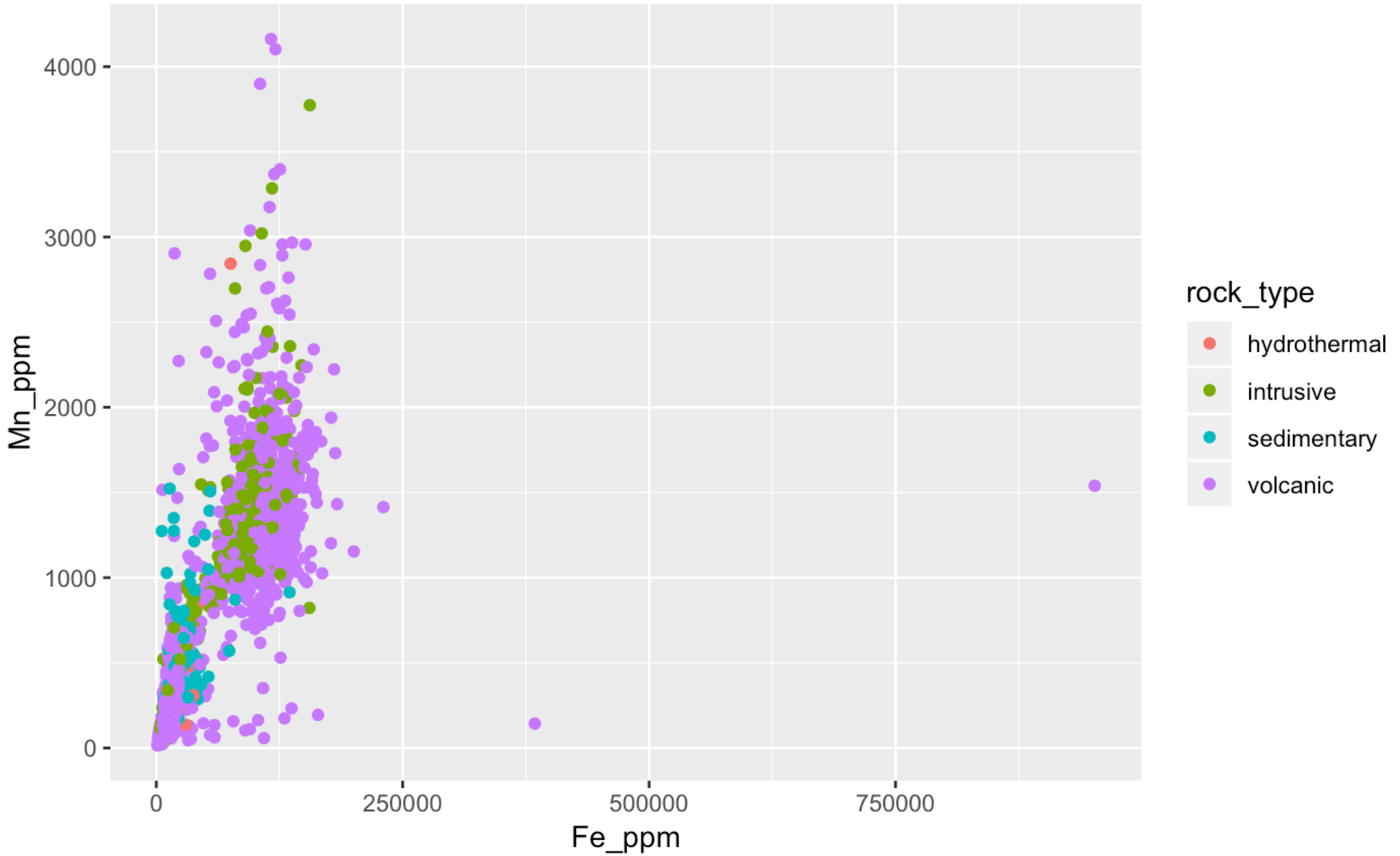



```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type)) +
  theme_bw(10)
```

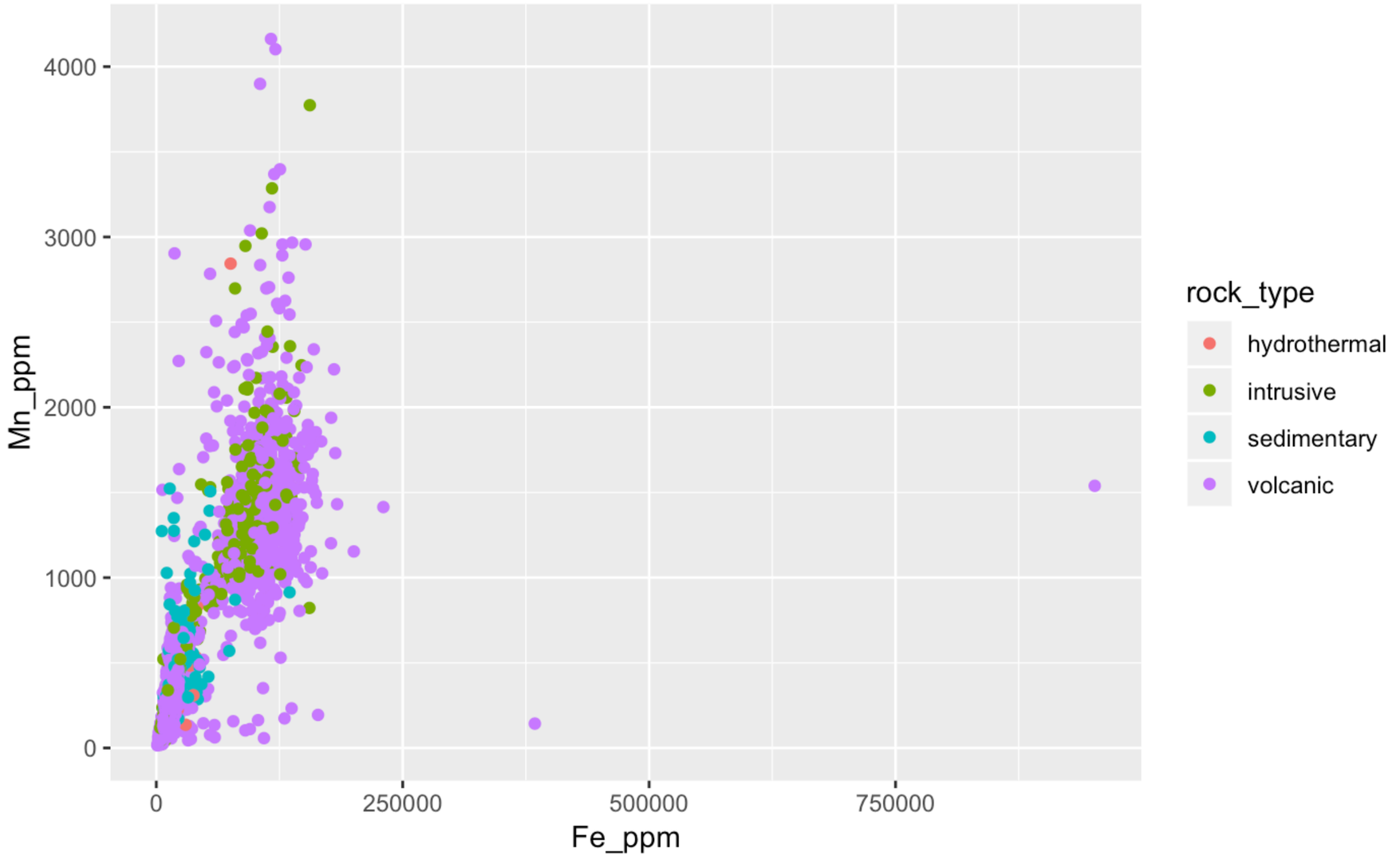



```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type)) +
  theme_bw(10) + theme(axis.text.x = element_text(...))
```

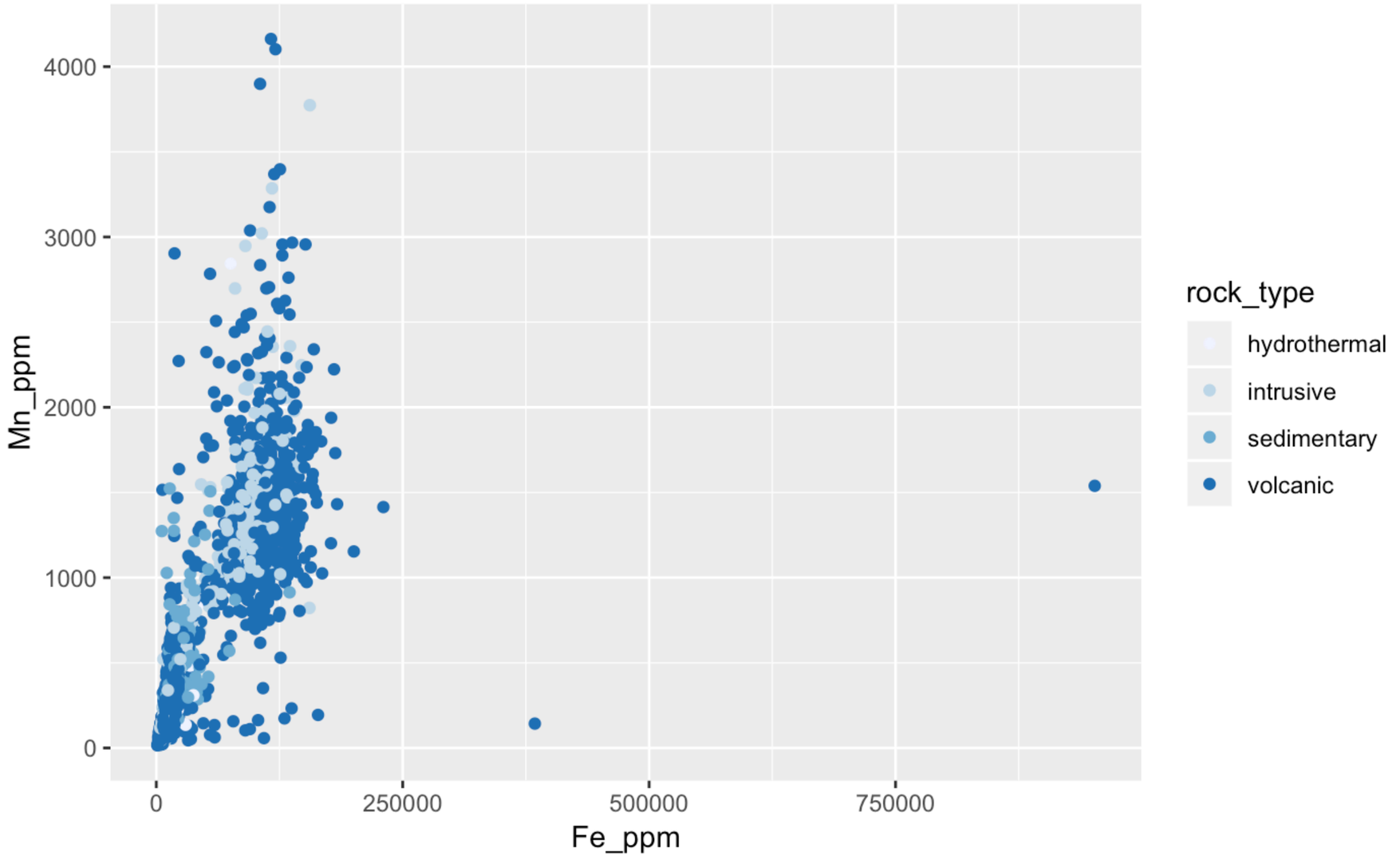
Scales




```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type))
```




```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type)) +
  scale_color_discrete()
```

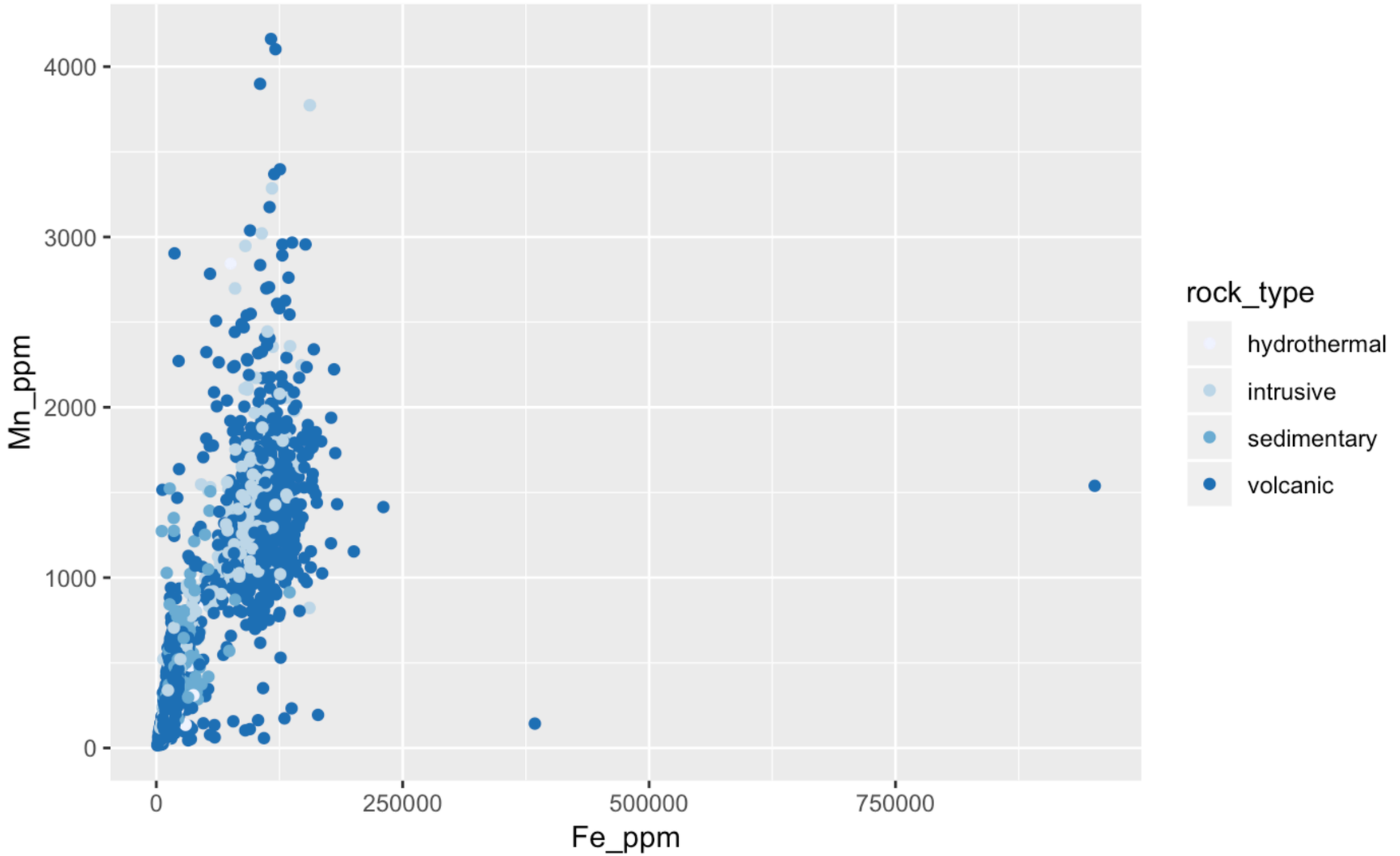



```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type)) +
  scale_color_brewer()
```

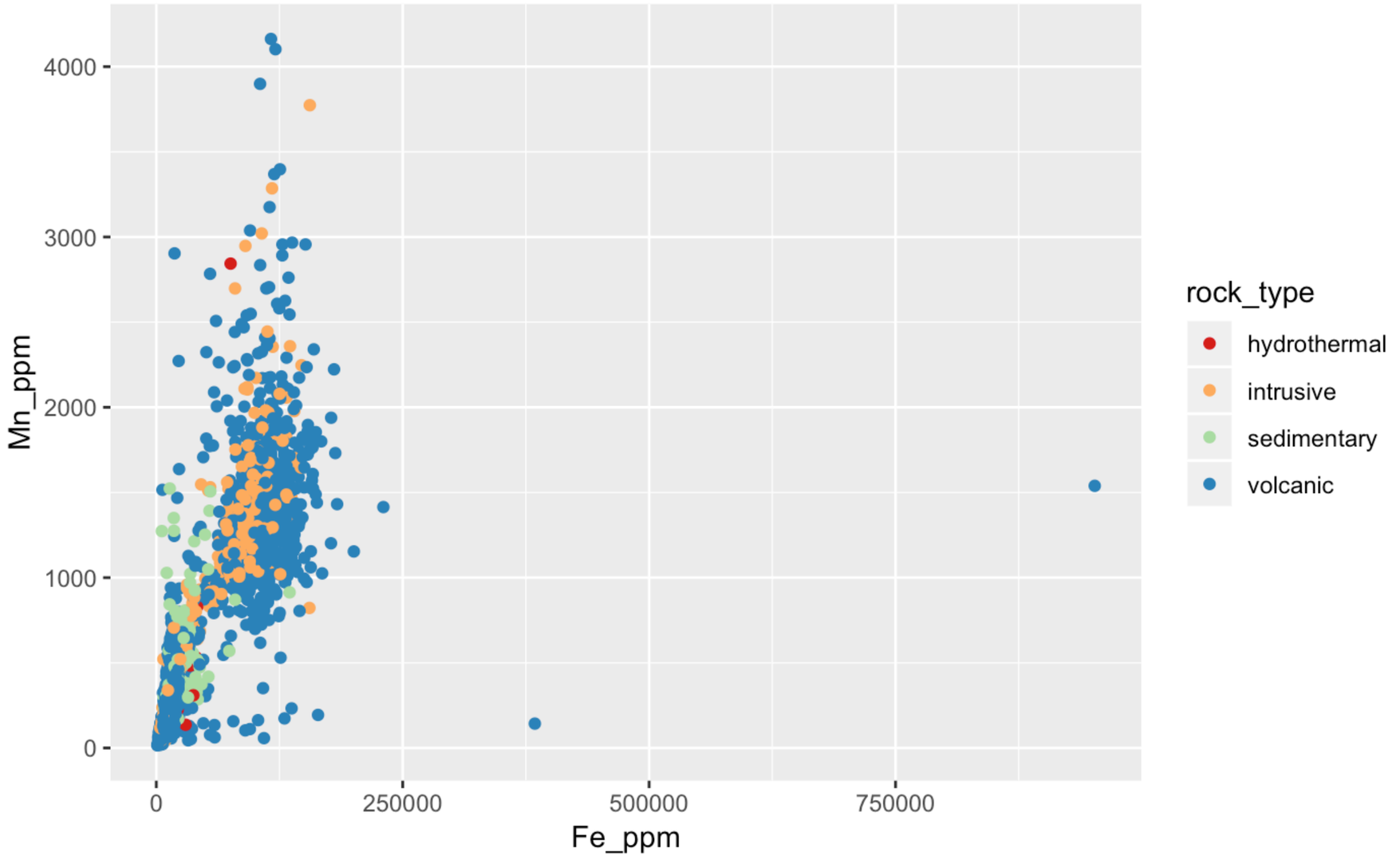
Your Turn 2

Set the palette argument of scale_color_brewer to one of the values listed in the "Palettes" section of the help file. (You can also preview them at http://colorbrewer2.org/).

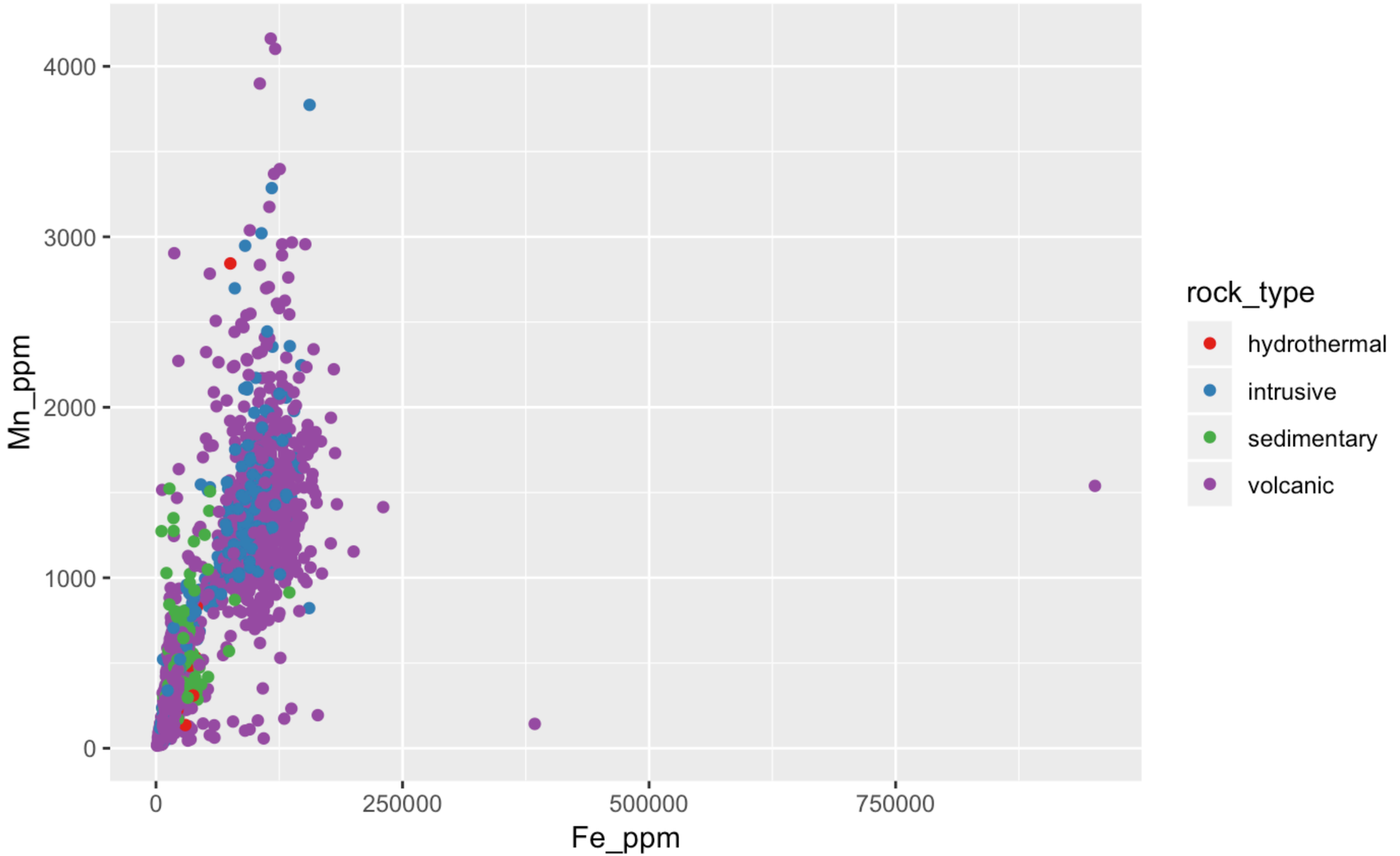
```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type)) +
  scale_color_brewer(...)
```



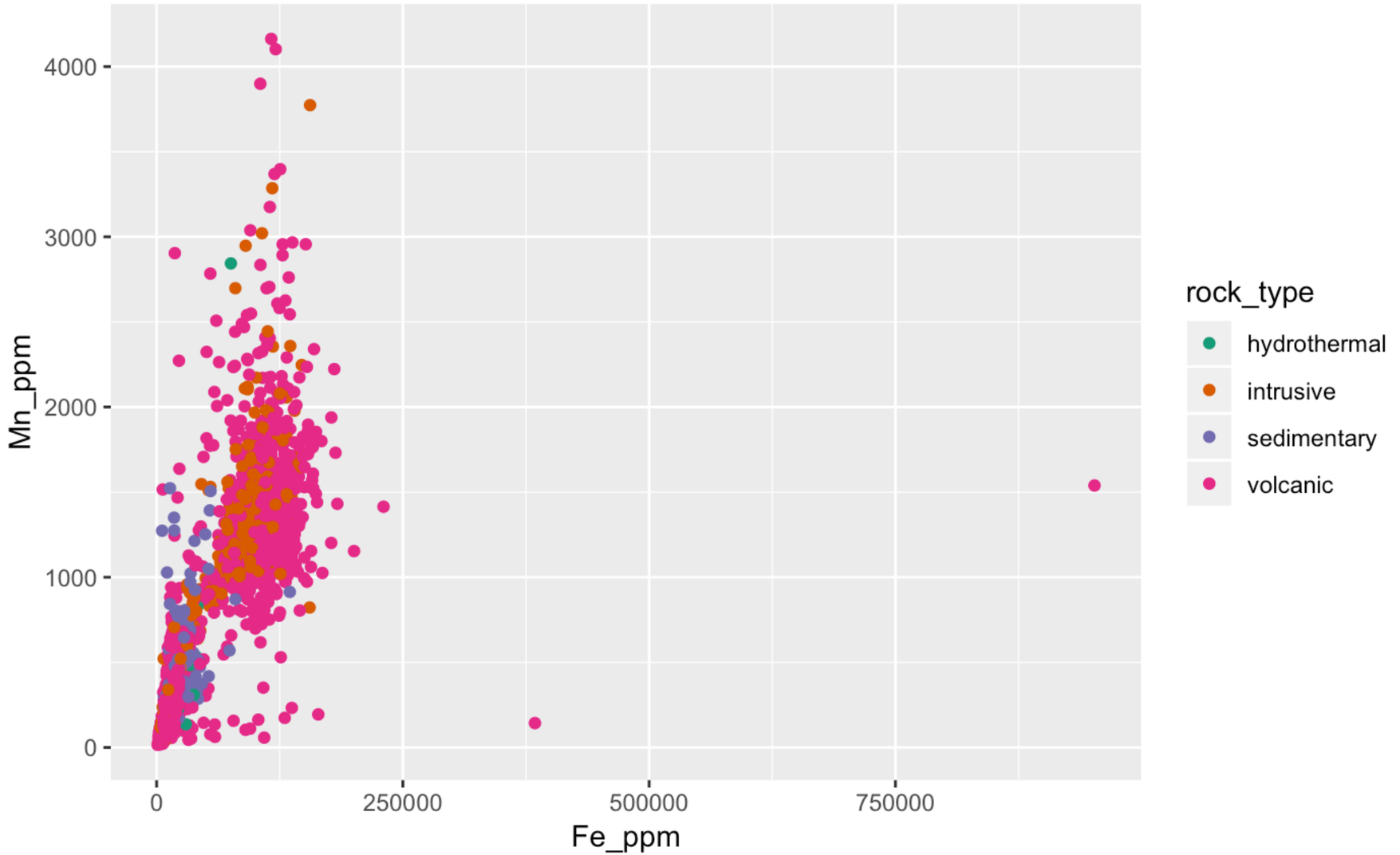

```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type)) +
  scale_color_brewer(palette = "Blues")
```




```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type)) +
  scale_color_brewer(palette = "Spectral")
```

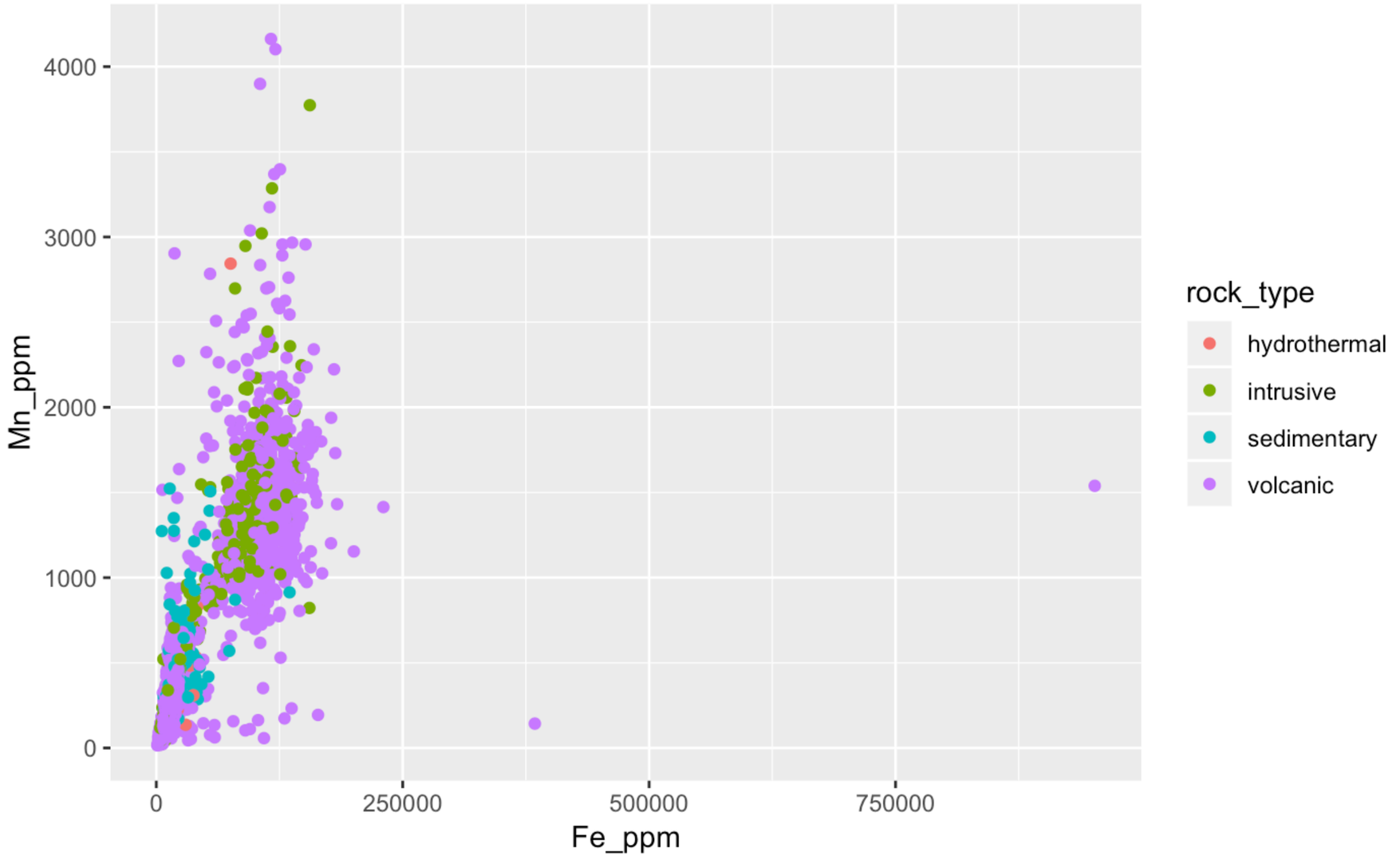



```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type)) +
  scale_color_brewer(palette = "Set1")
```

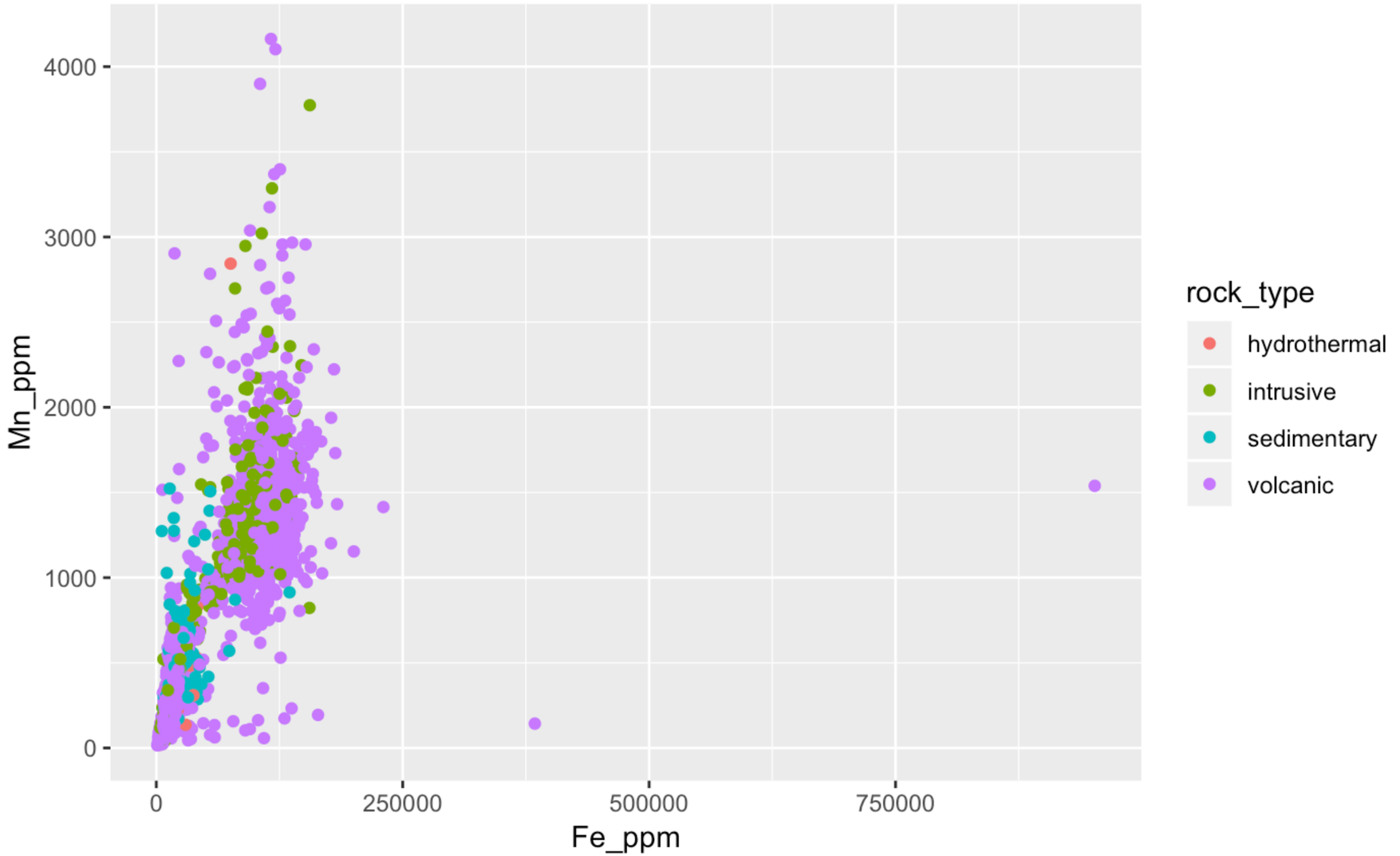



```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type)) +
  scale_color_brewer(palette = "Dark1")
```

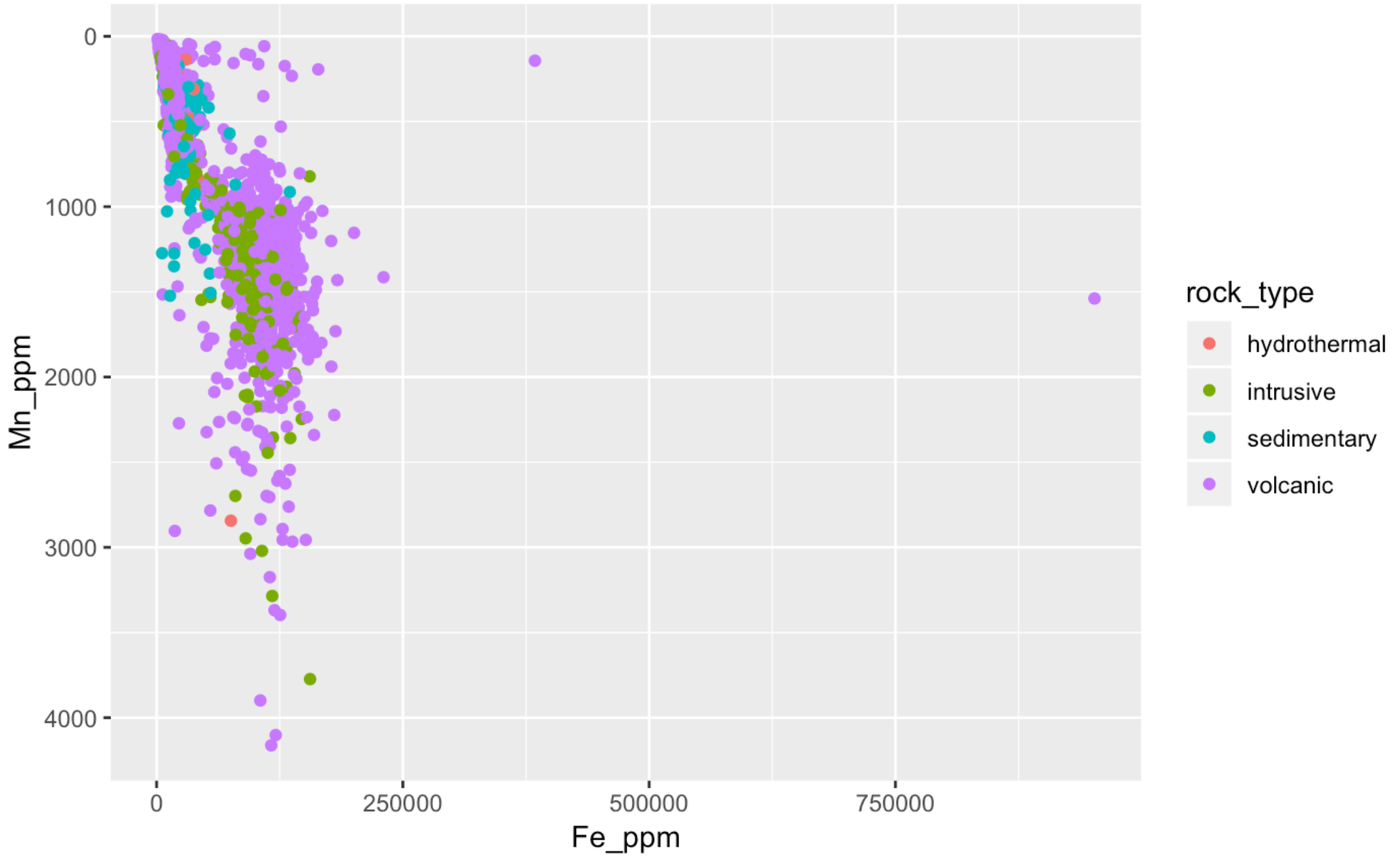
Axis Scales




```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type))
```



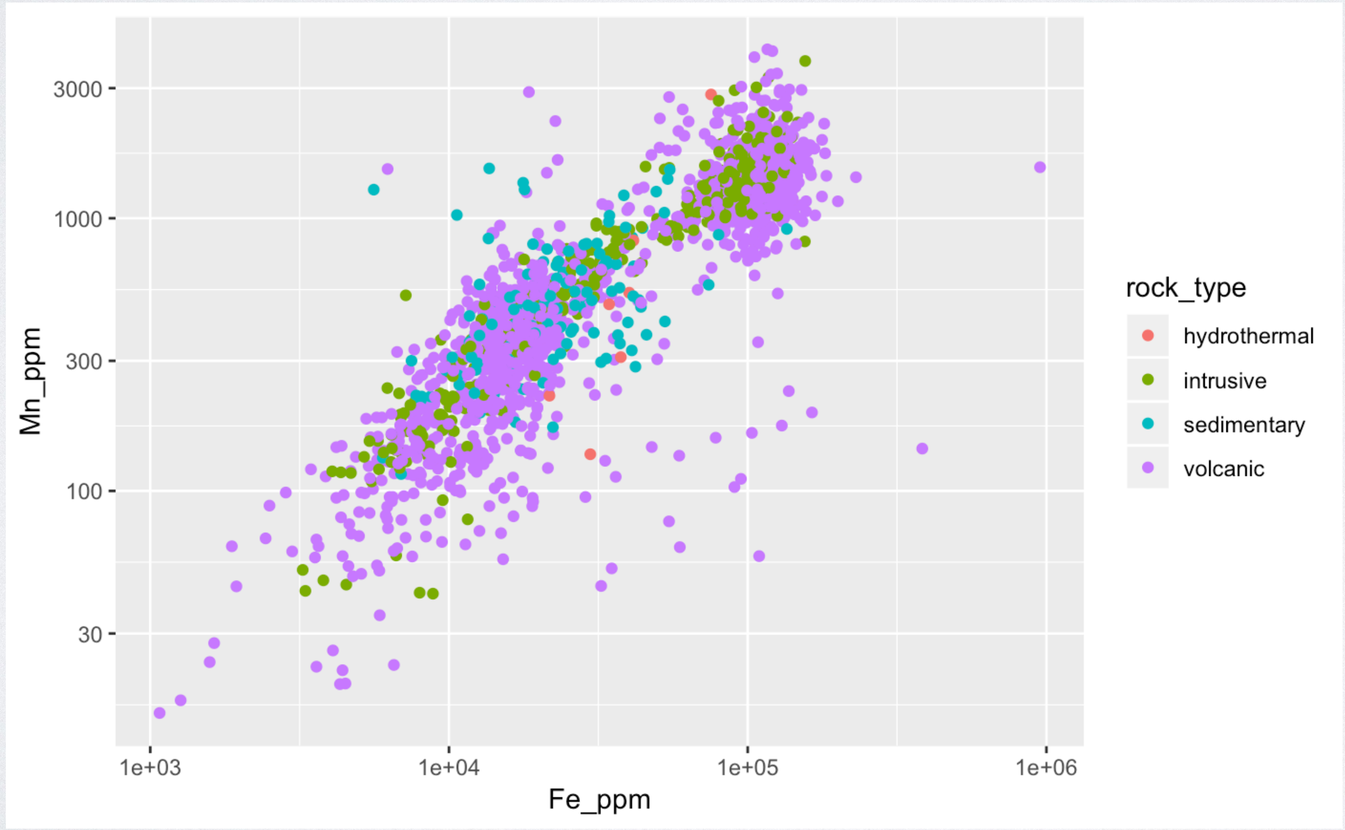

```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type)) +
  scale_x_continuous() + scale_y_continuous()
```



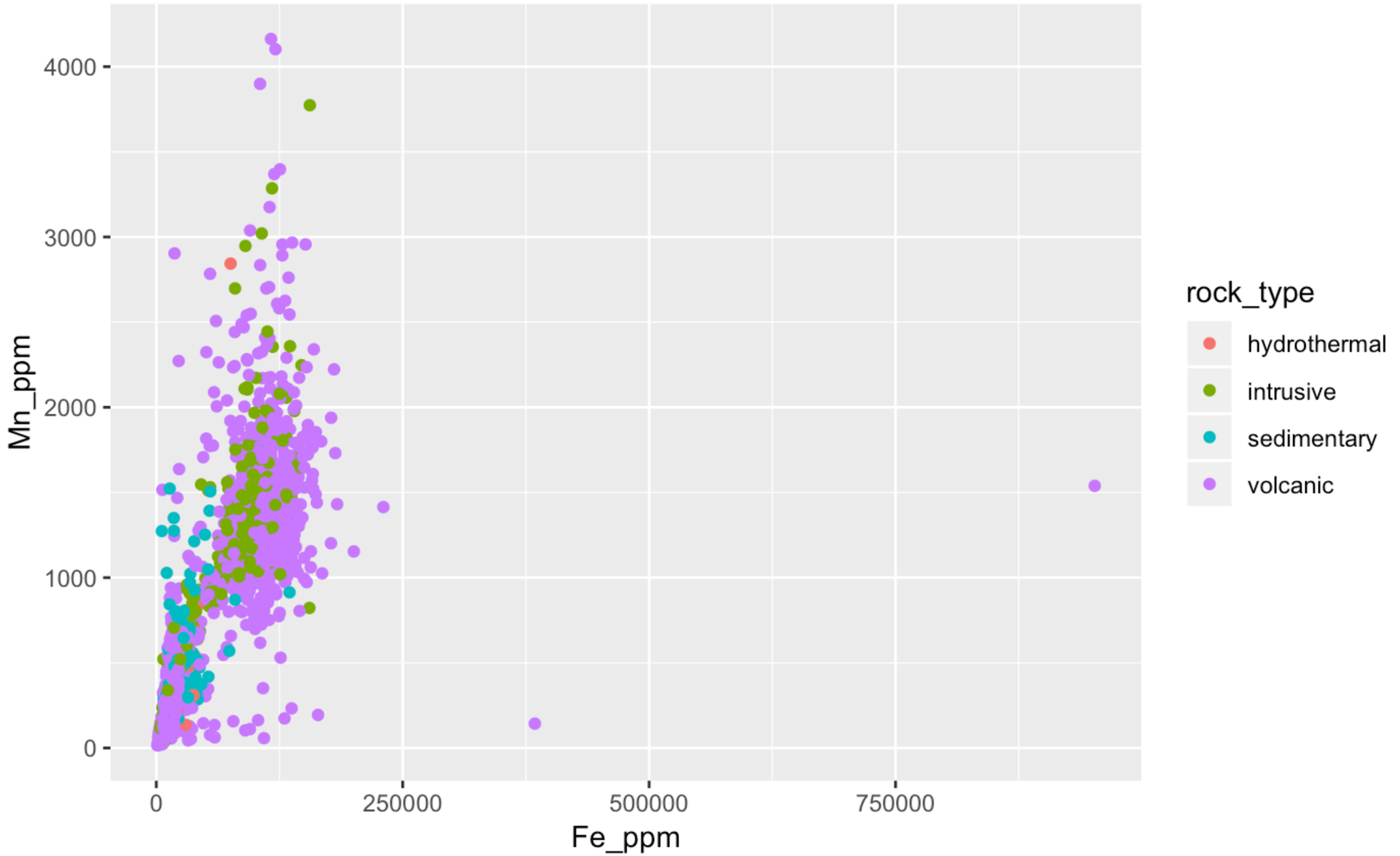

```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type)) +
  scale_x_continuous() + scale_y_reverse()
```

Your Turn 3

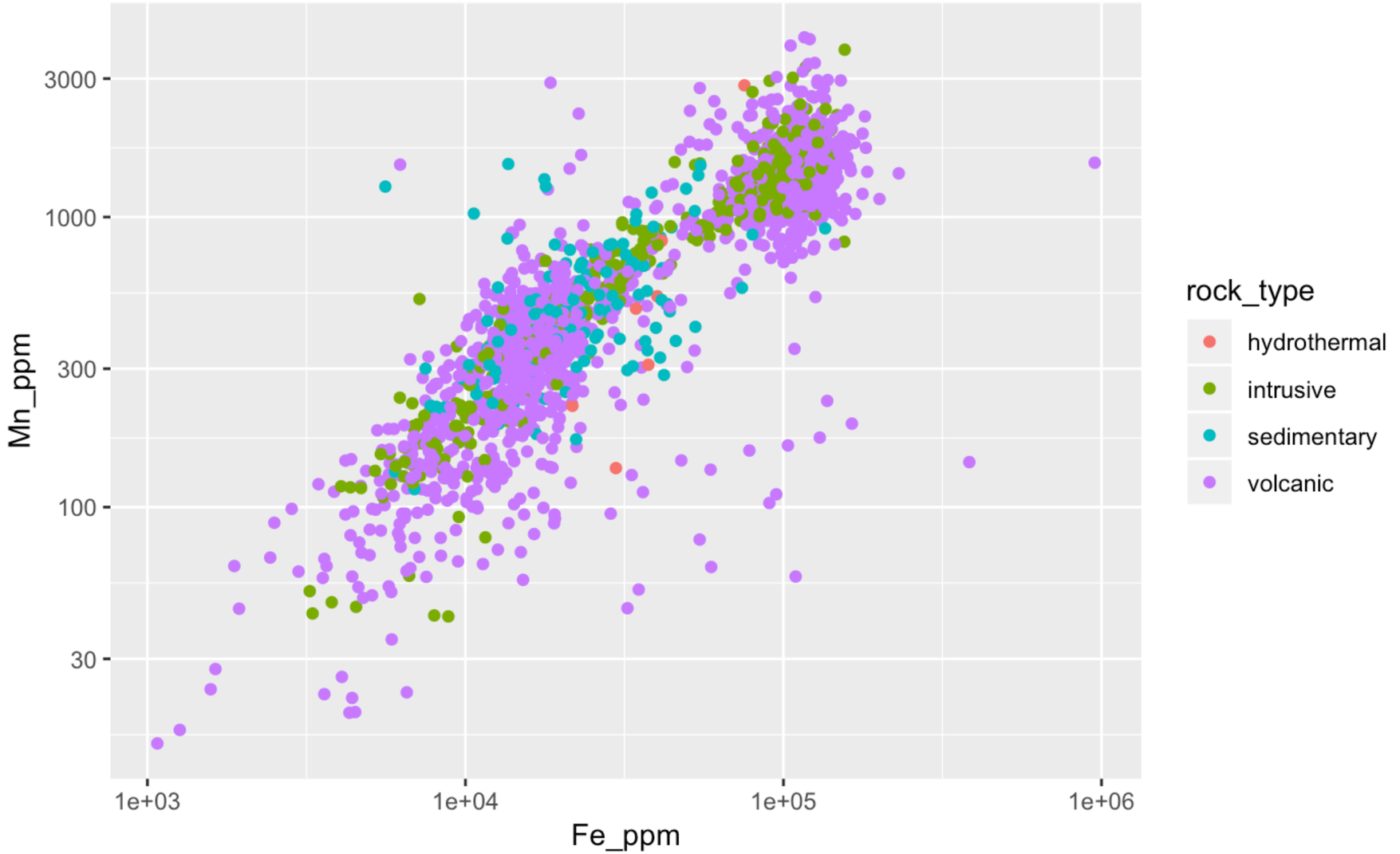
Use the appropriate scale_x_* and scale_y_* functions to create a plot with logged X and Y axes.



04:00

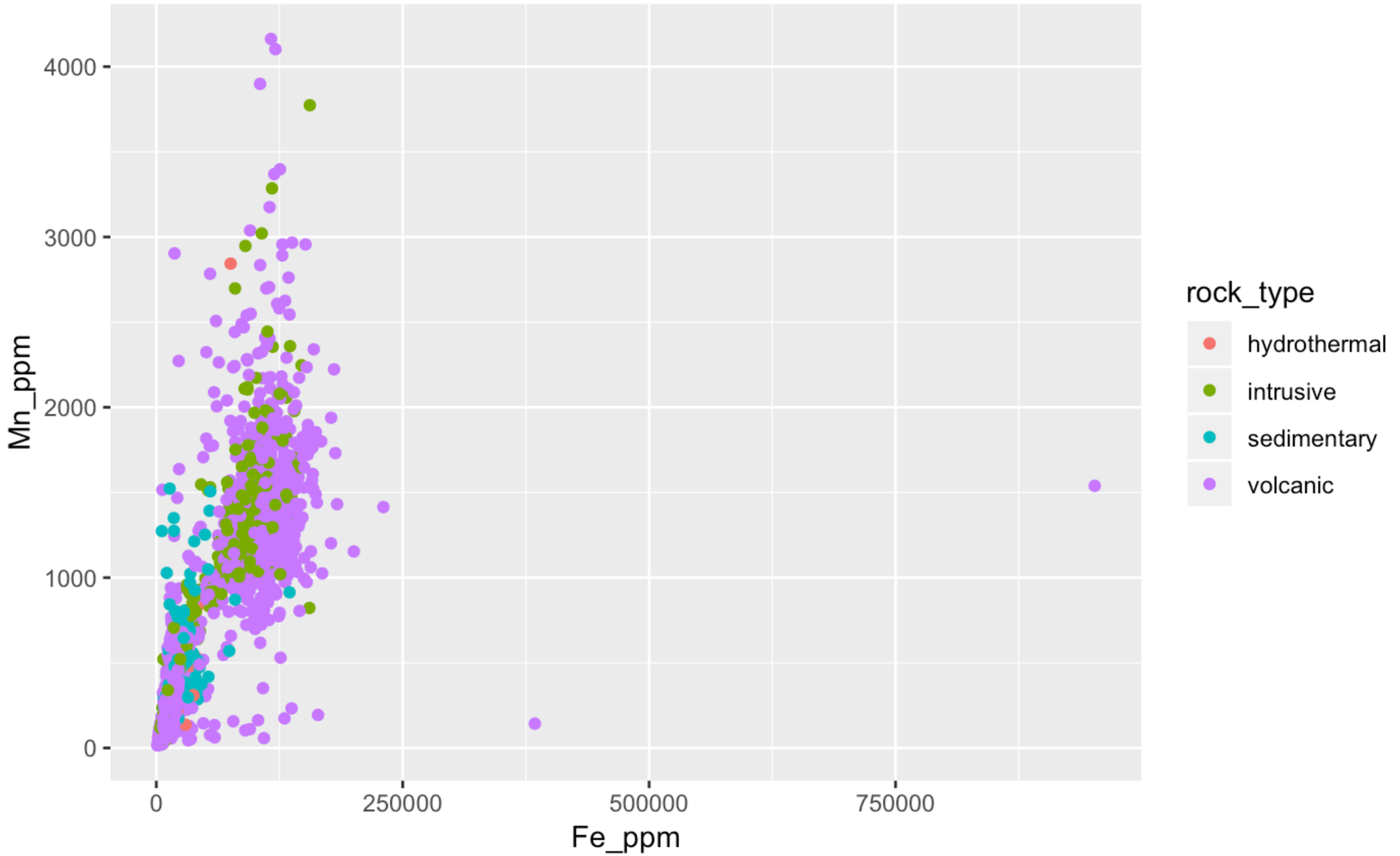



```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type)) +
  scale_x_continuous() + scale_y_continuous()
```

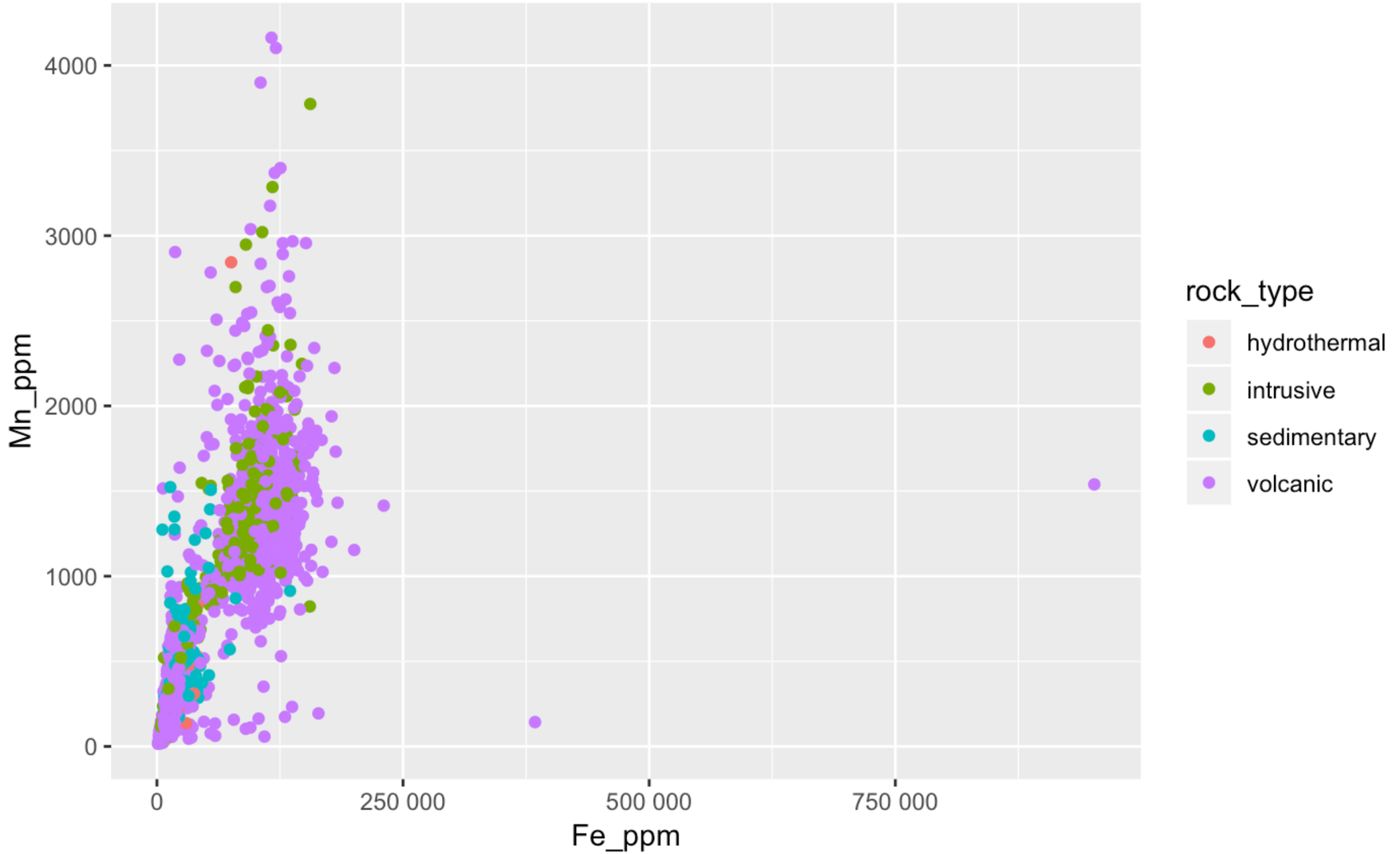



```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type)) +
  scale_x_log10() + scale_y_log10()
```

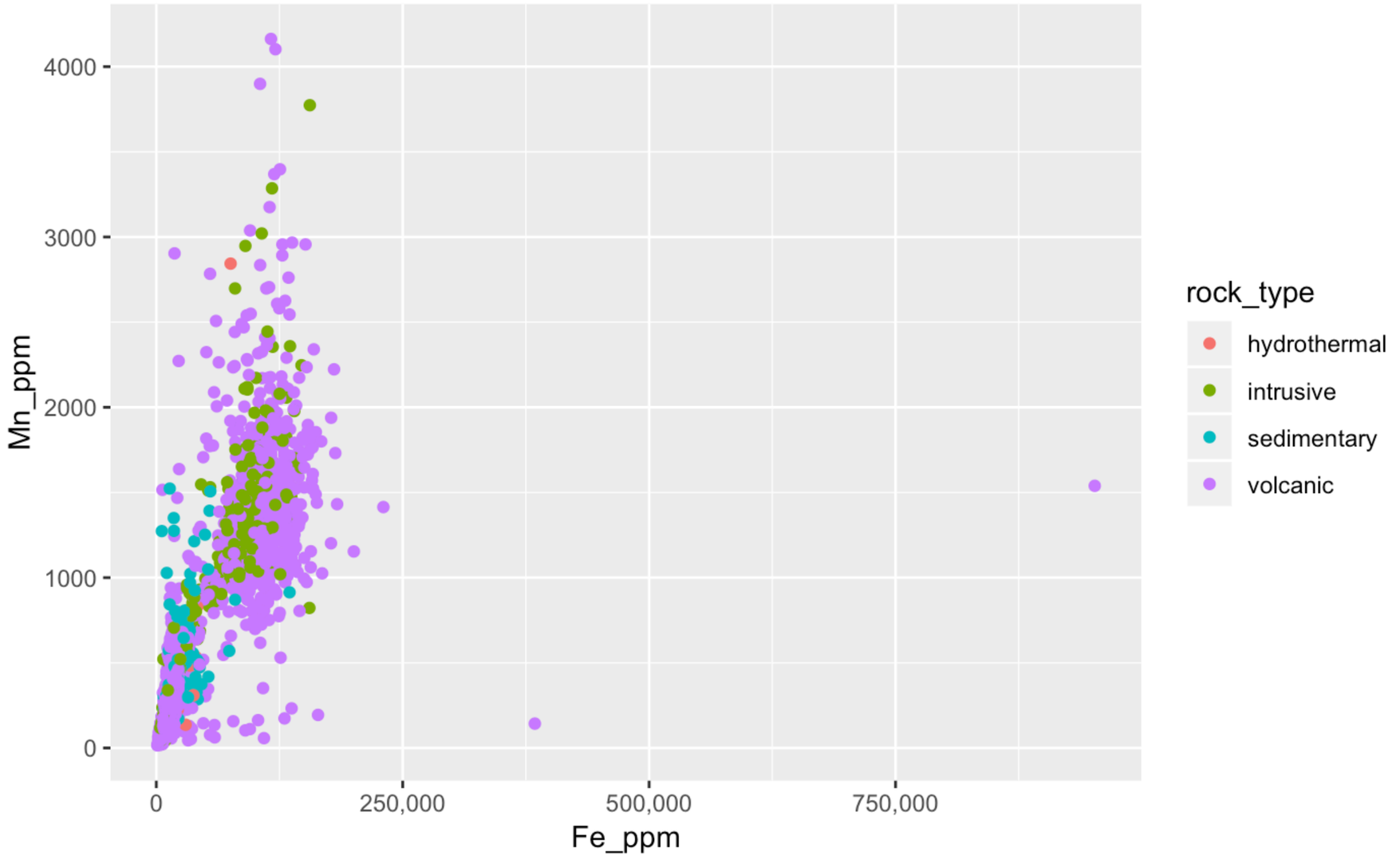
Continuous Scales



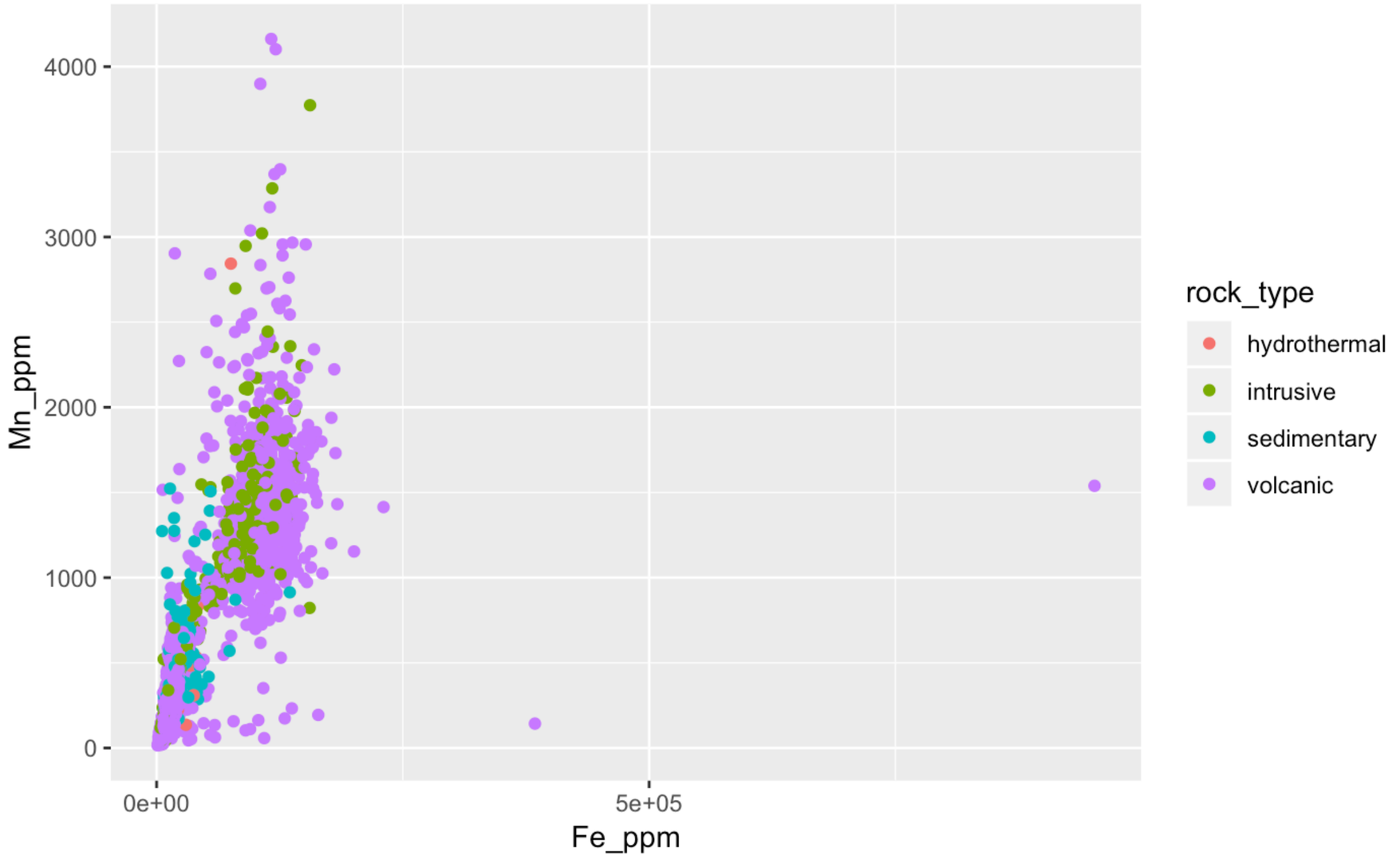

```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type)) +
  scale_x_continuous()
```



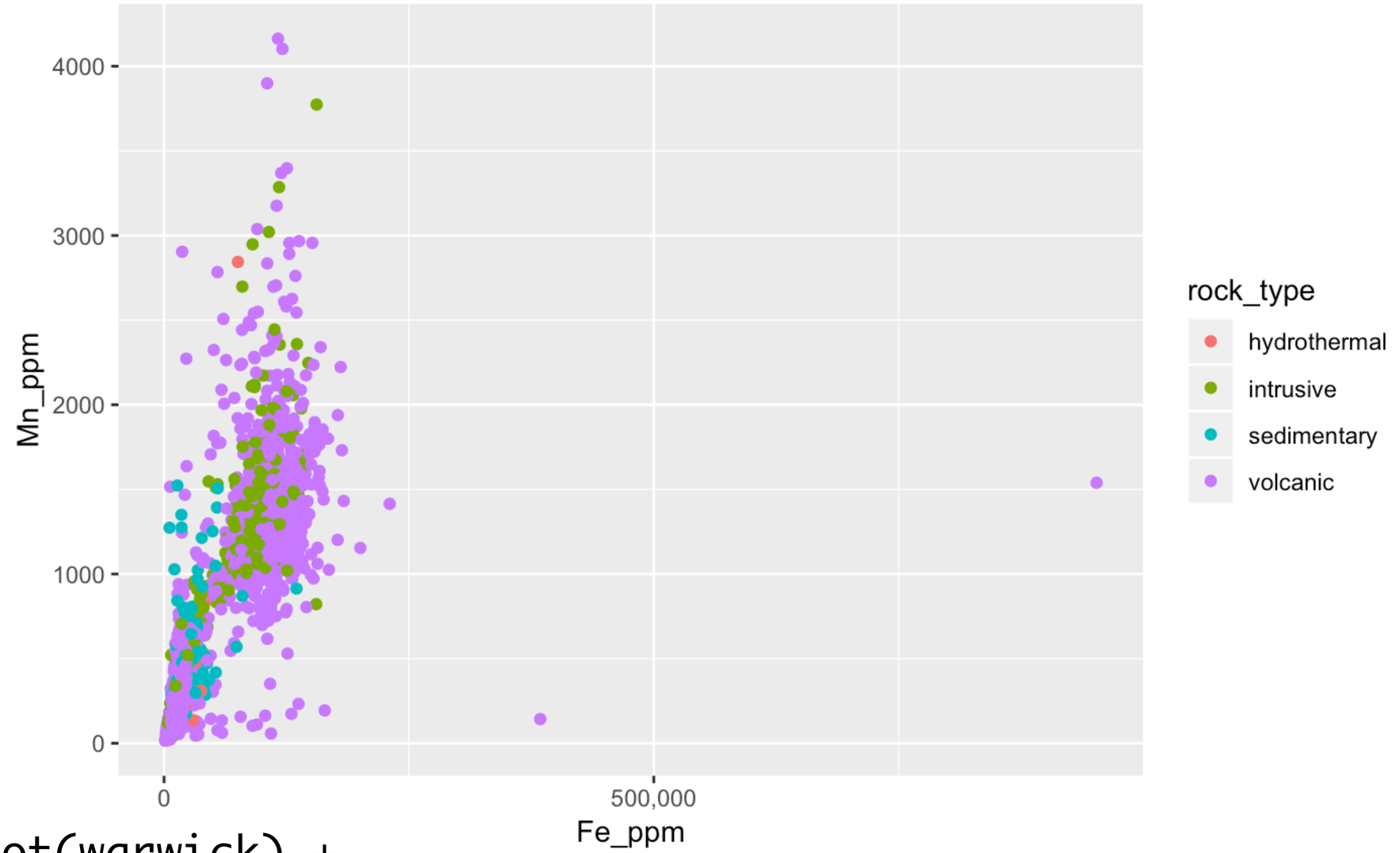

```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type)) +
  scale_x_continuous(labels = scales::number_format())
```




```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type)) +
  scale_x_continuous(labels = scales::comma_format())
```

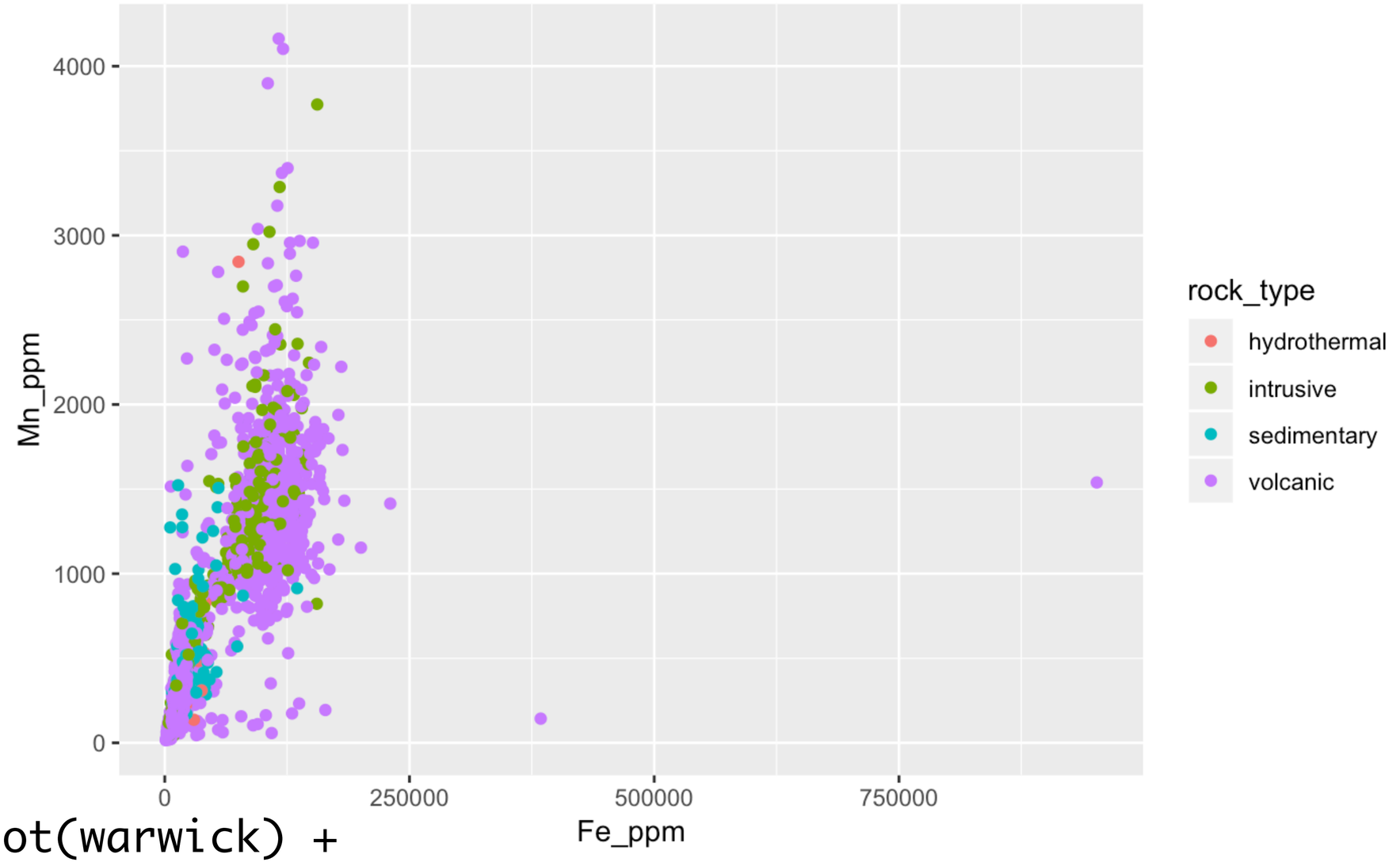



```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type)) +
  scale_x_continuous(breaks = c(0, 500000))
```

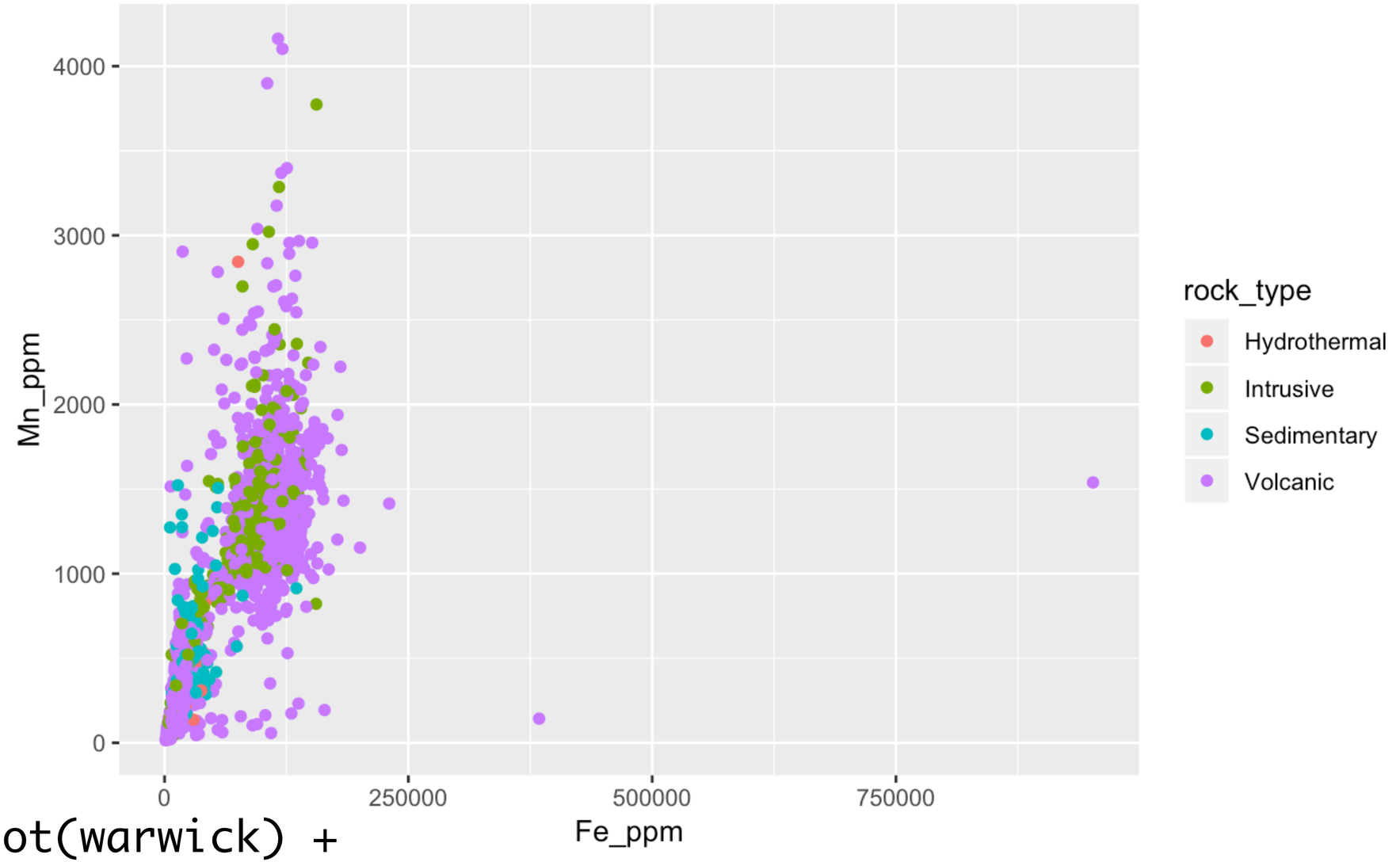



```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type)) +
  scale_x_continuous(breaks = c( 0,   500000),
                     labels = c("0", "500,000"))
```

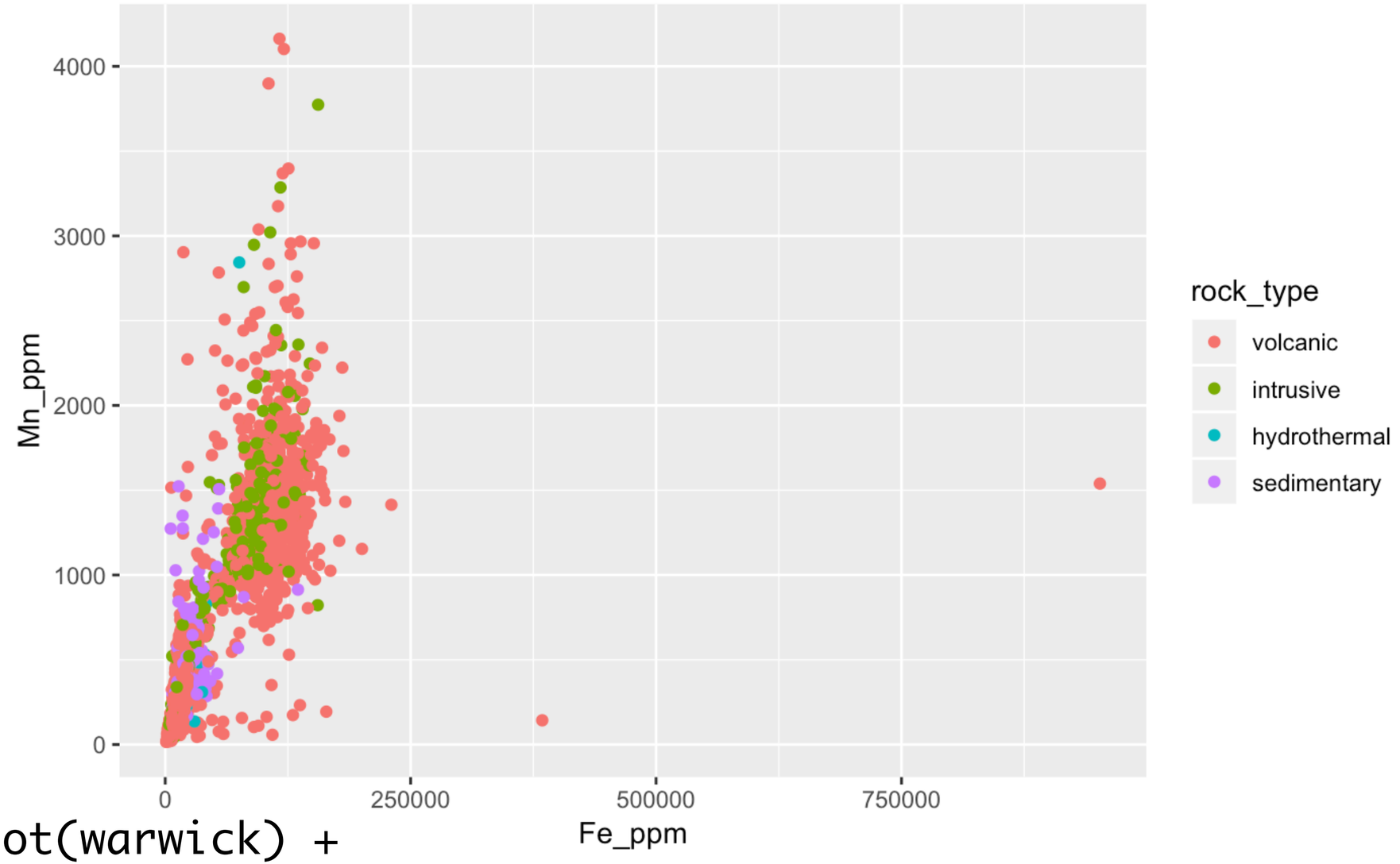
Discrete Scales




```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type)) +
  scale_color_discrete()
```

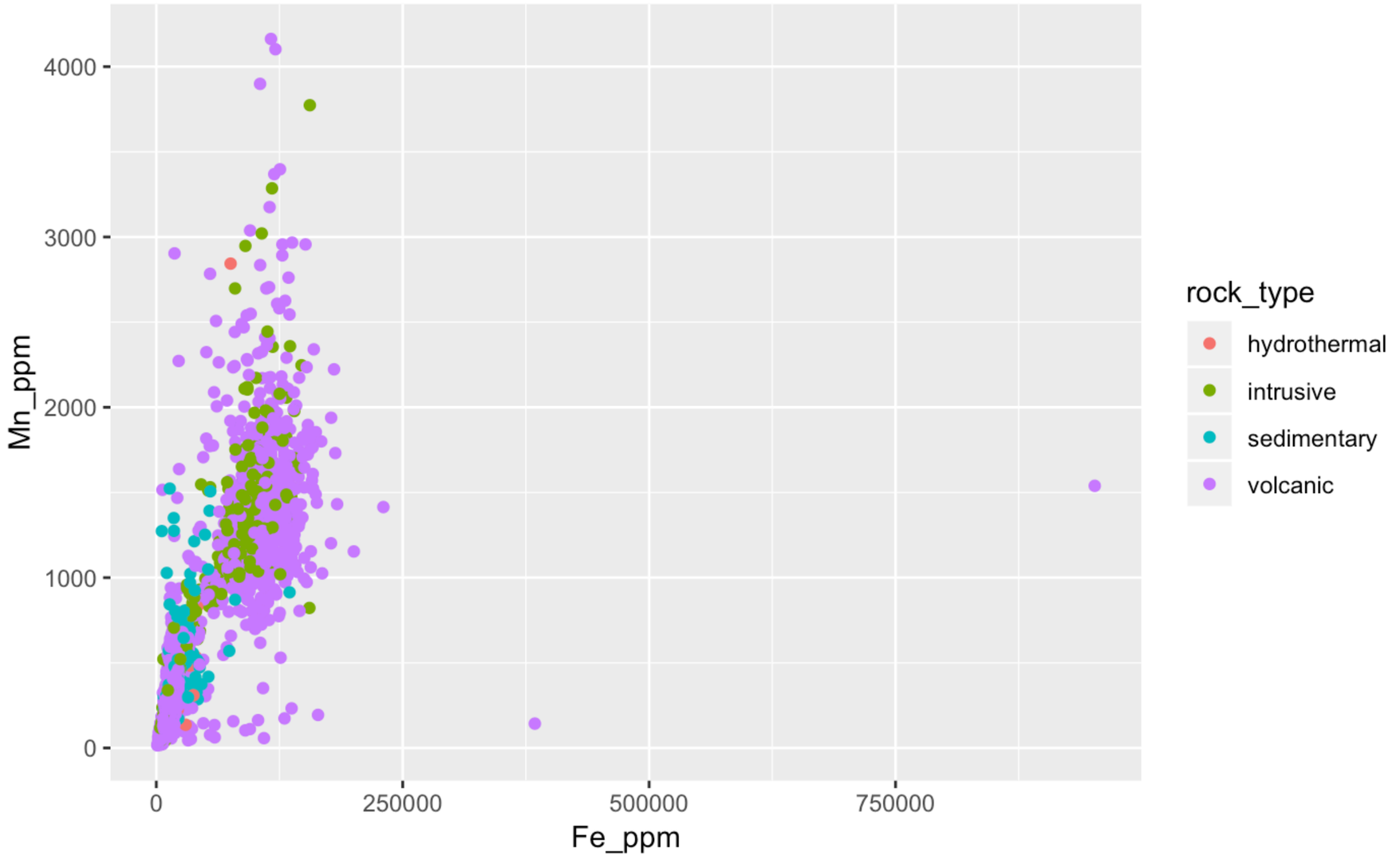



```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type)) +
  scale_color_discrete(labels = c("Hydrothermal", "Intrusive",
                                  "Sedimentary", "Volcanic"))
```

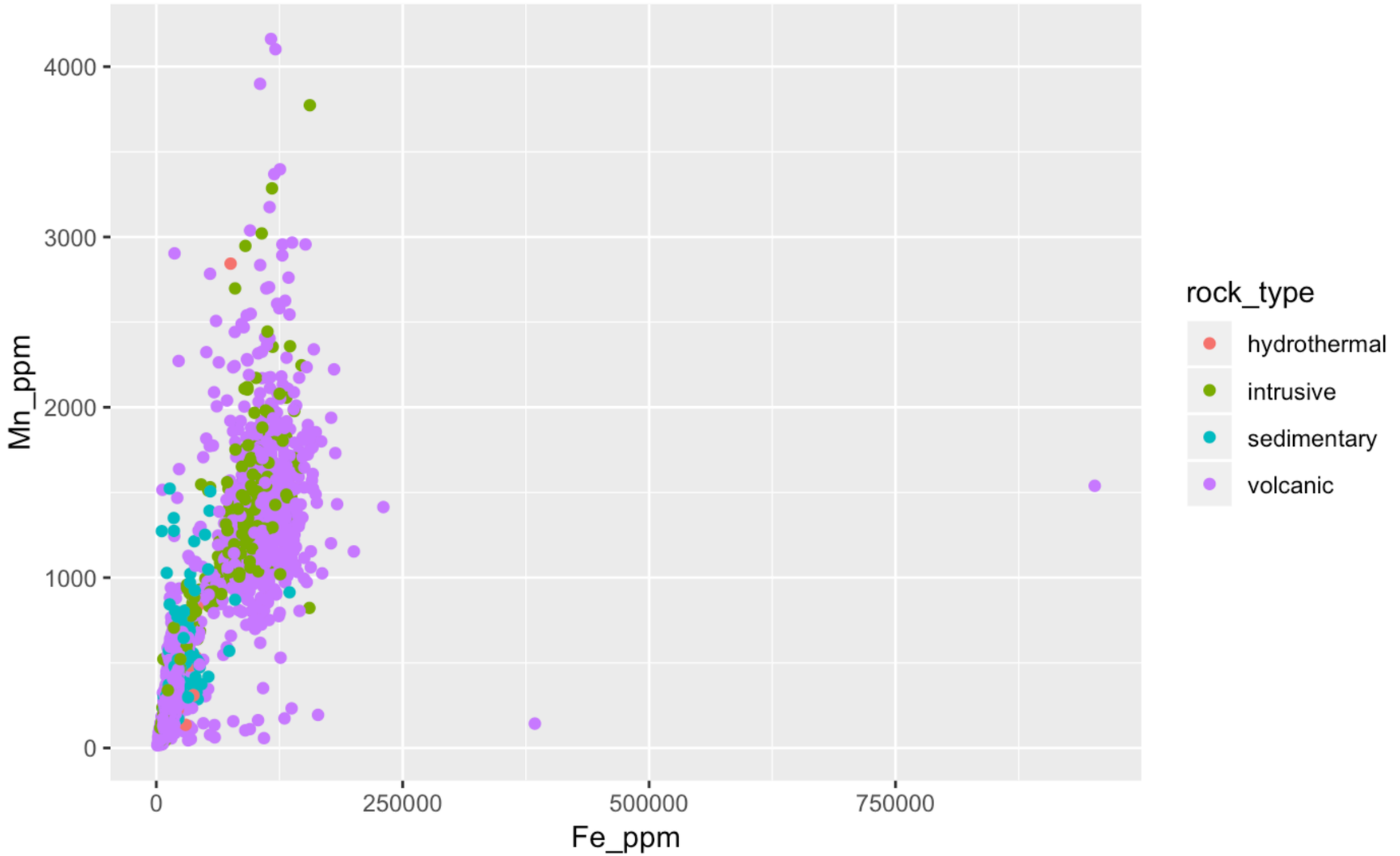



```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type)) +
  scale_color_discrete(limits = c("volcanic", "intrusive",
                                  "hydrothermal", "sedimentary"))
```

Facets




```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type))
```

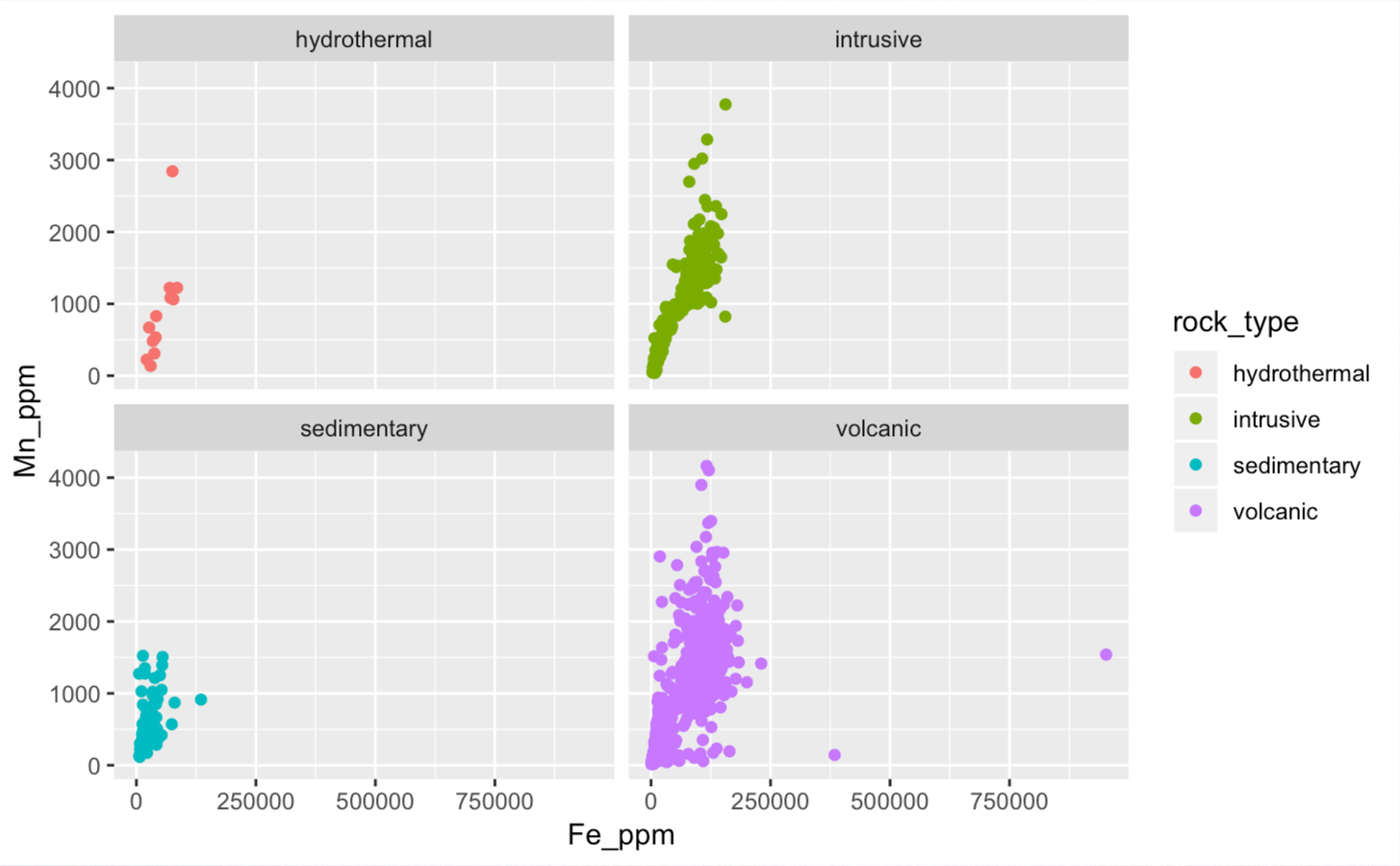


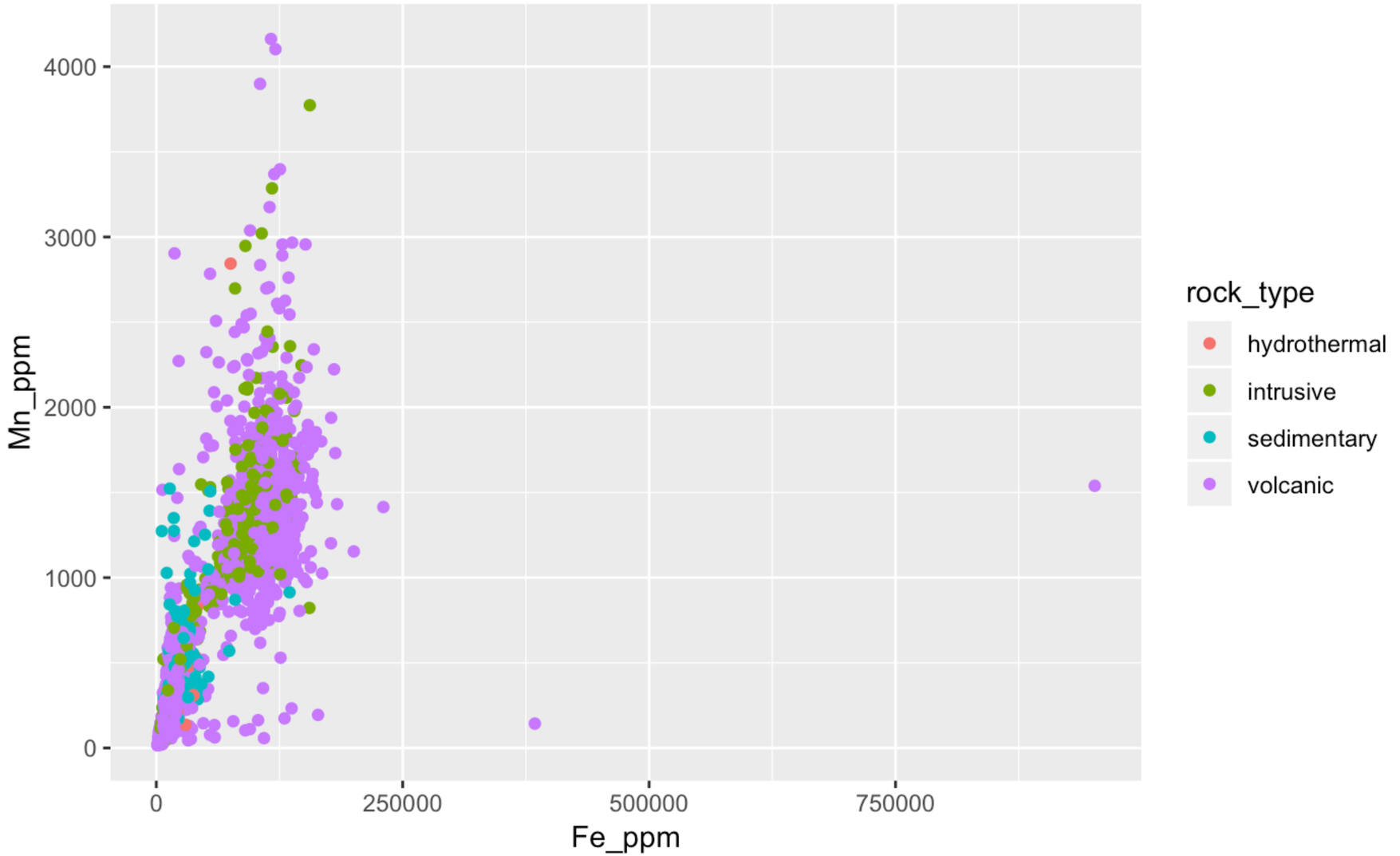

```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type)) +
  facet_null()
```

Your Turn 4

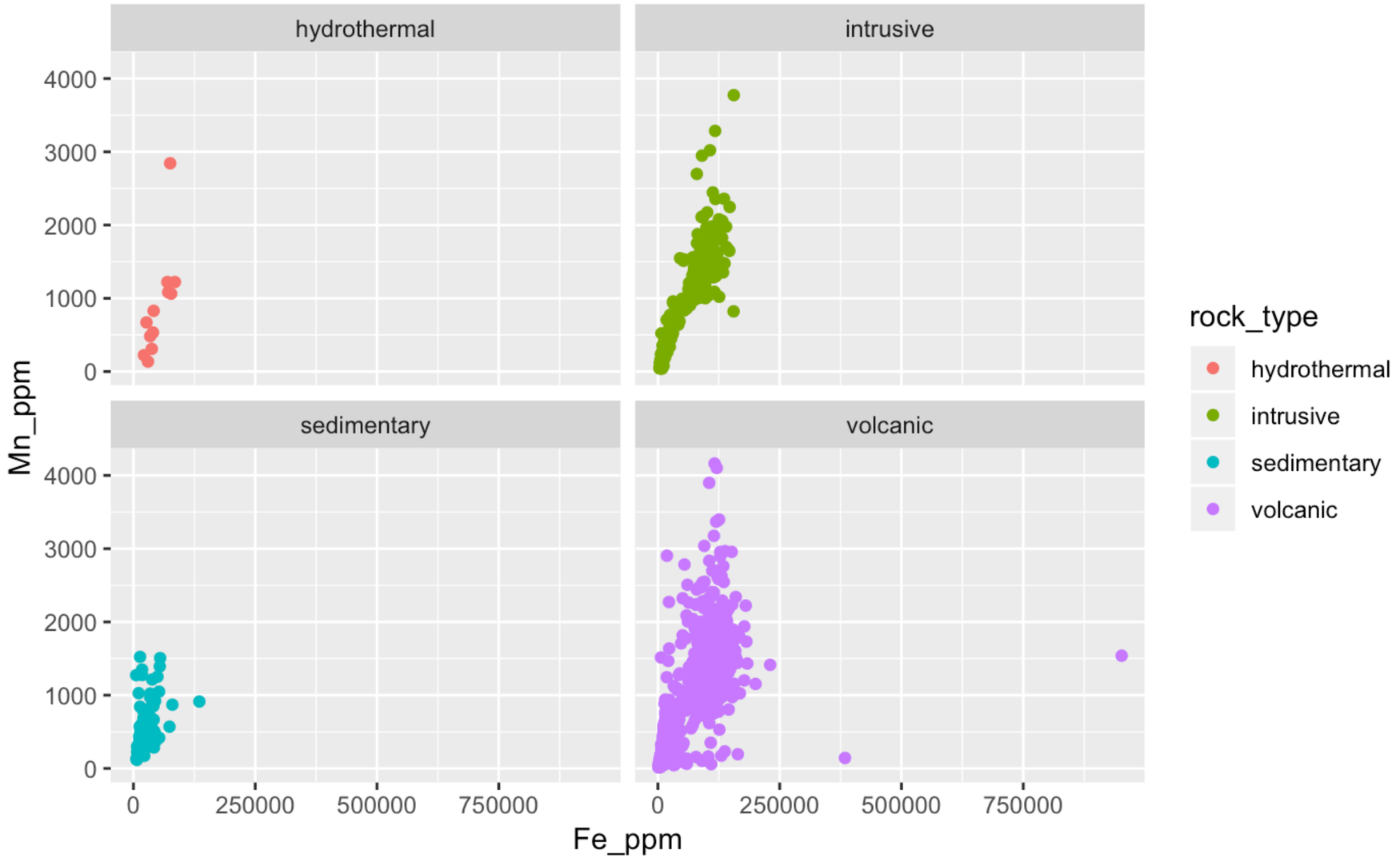
Use `facet_wrap()` to make facets for each `rock_type`.

Experiment with the `scales` argument.

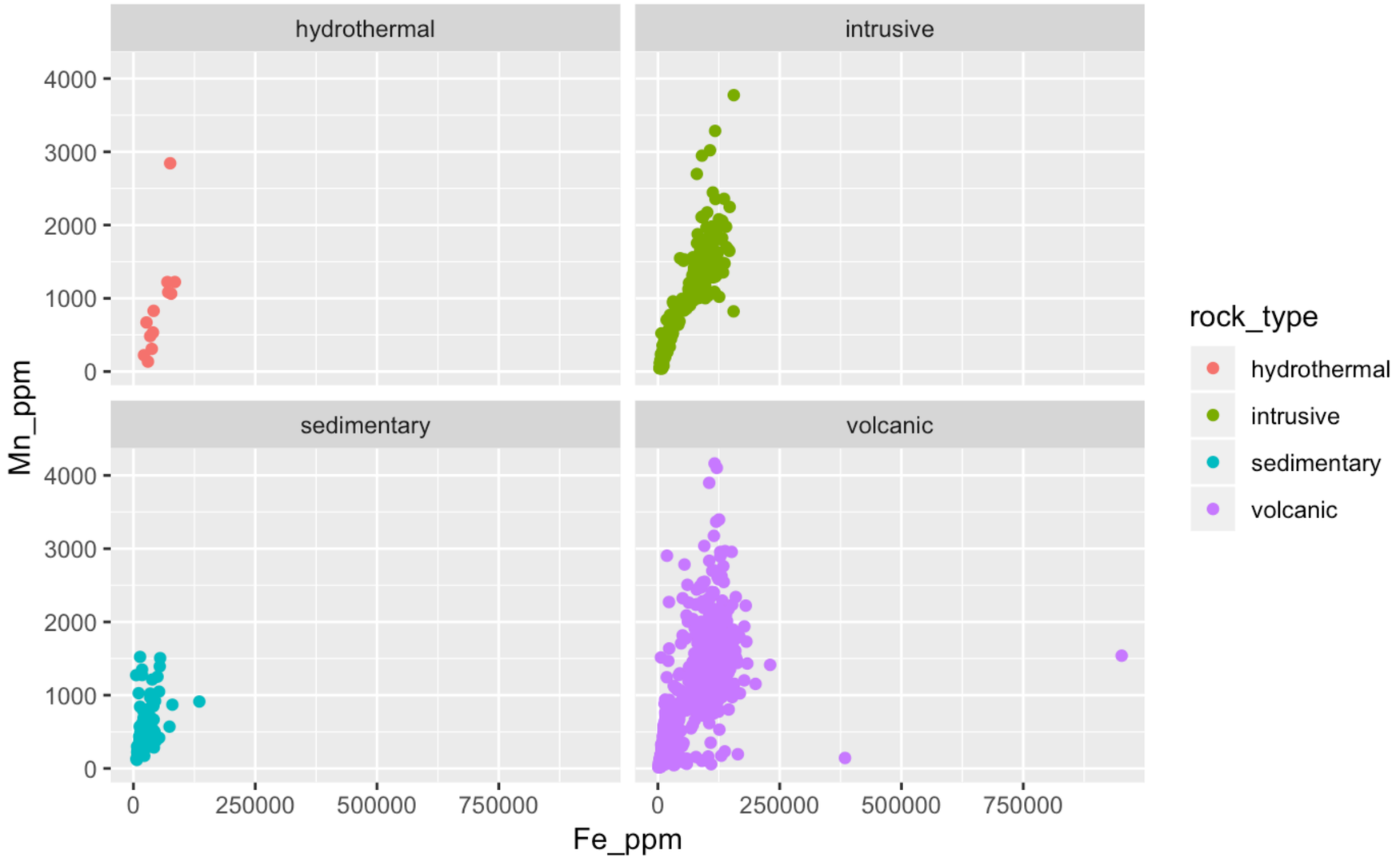




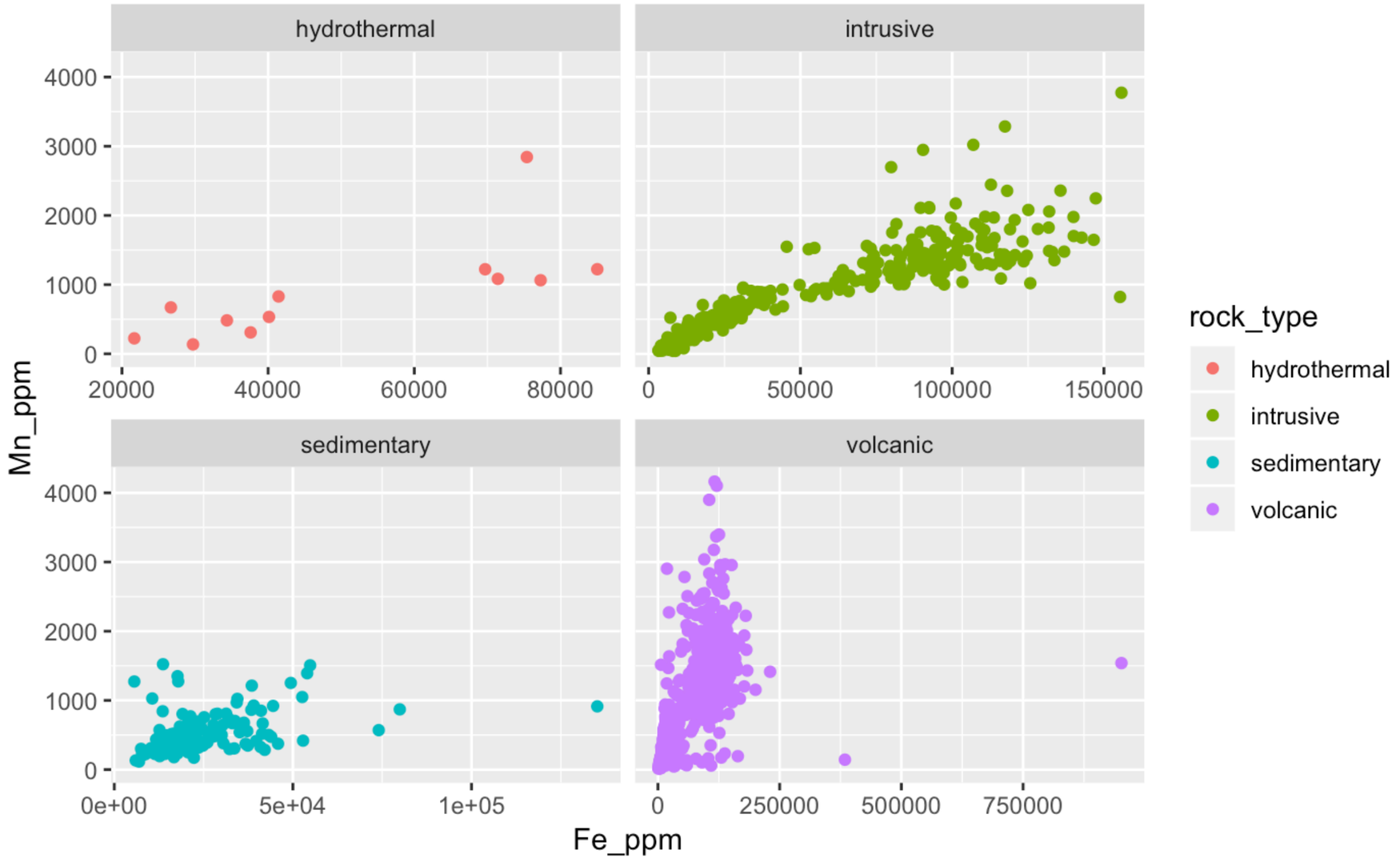

```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type)) +
  facet_null()
```



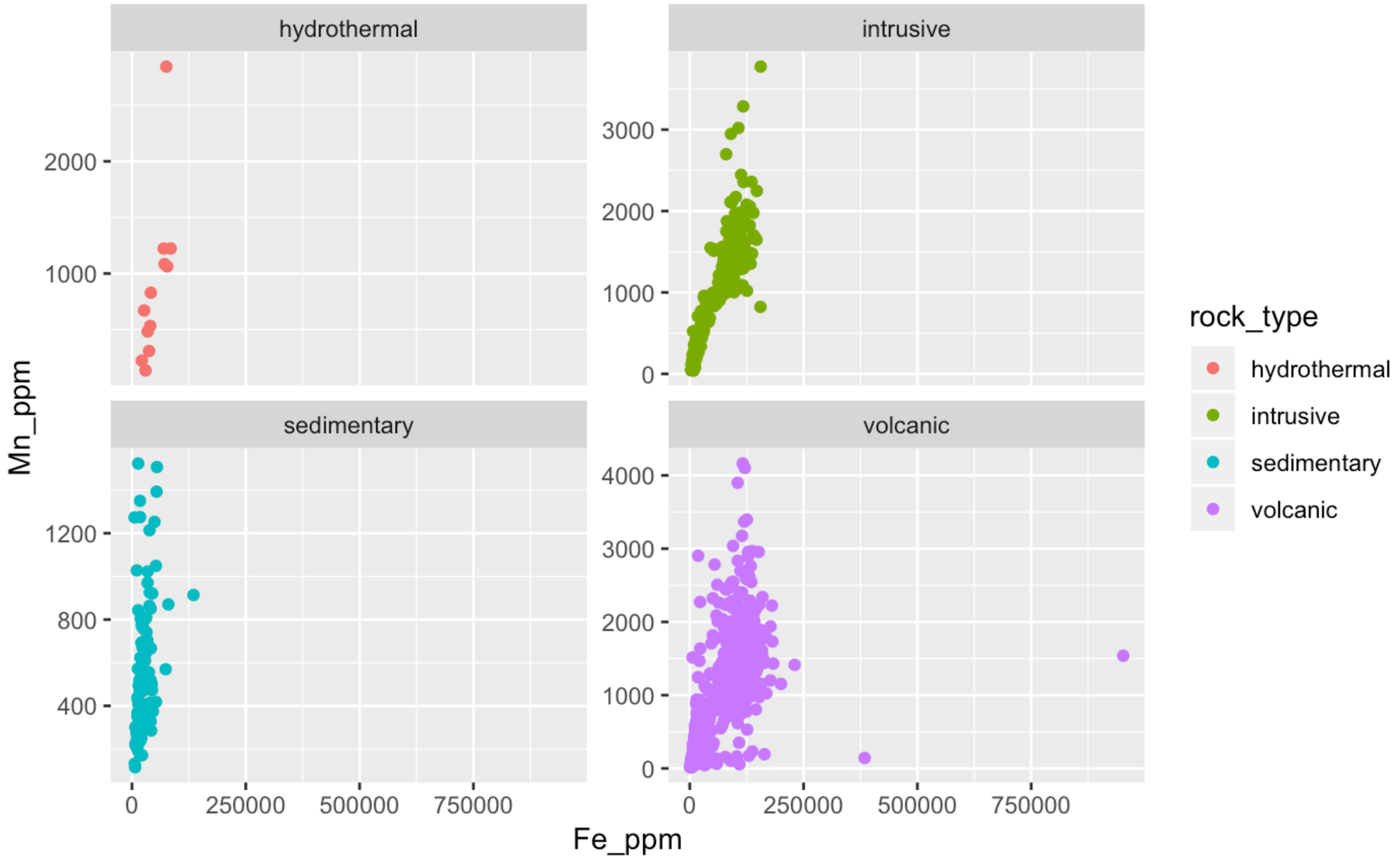

```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type)) +
  facet_wrap()
```



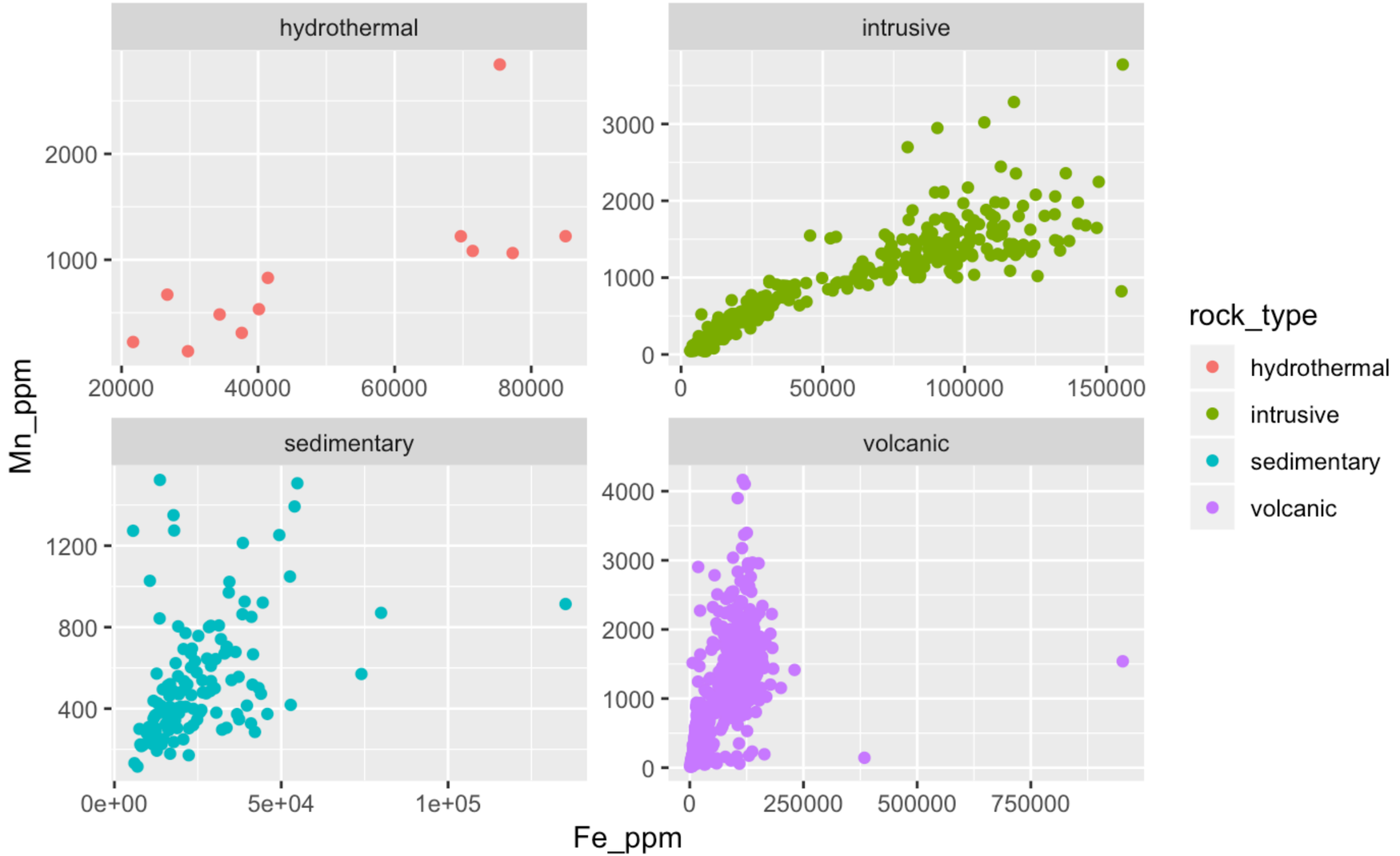

```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type)) +
  facet_wrap(scales = "fixed")
```




```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type)) +
  facet_wrap(scales = "free_x")
```

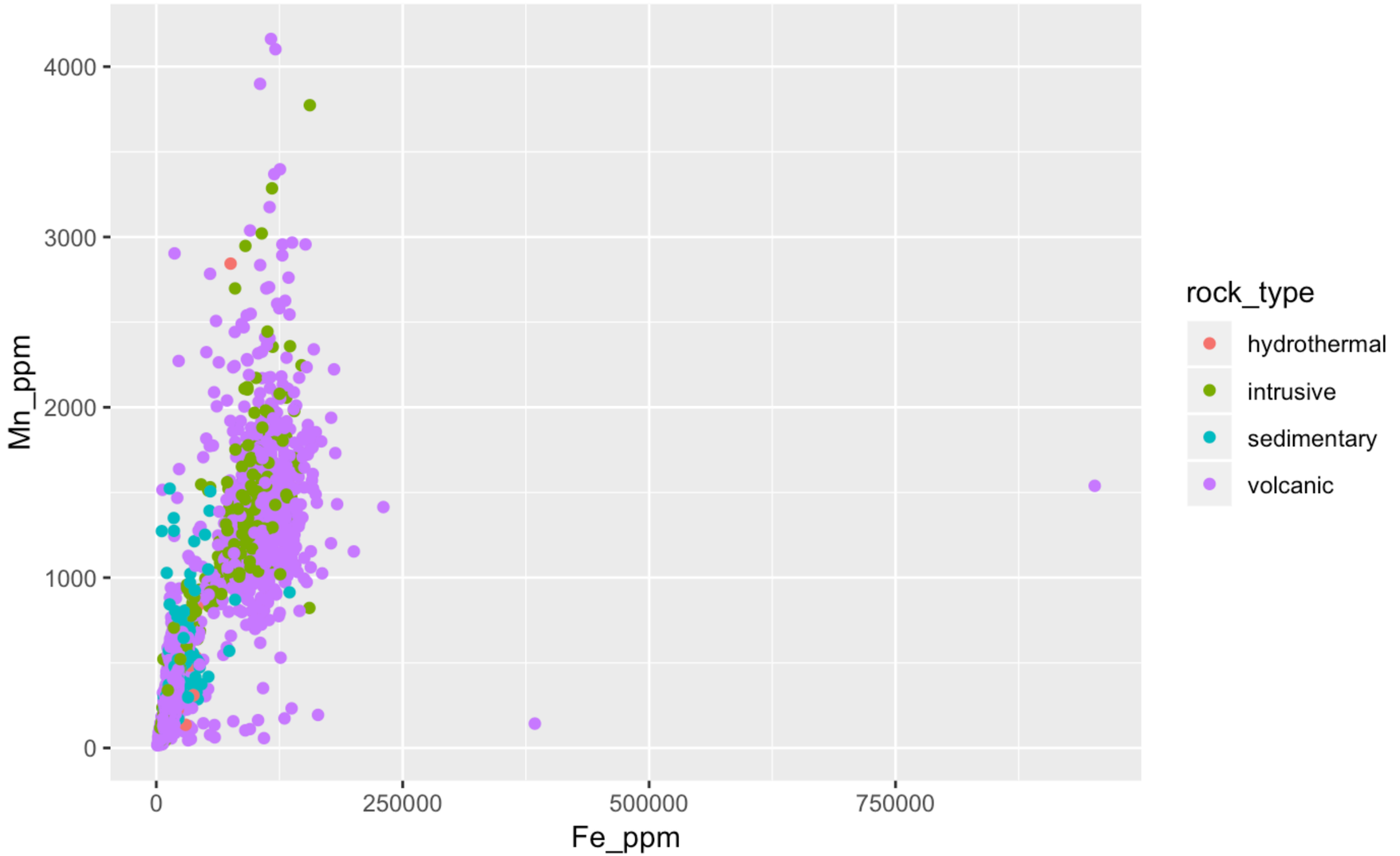



```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type)) +
  facet_wrap(scales = "free_y")
```

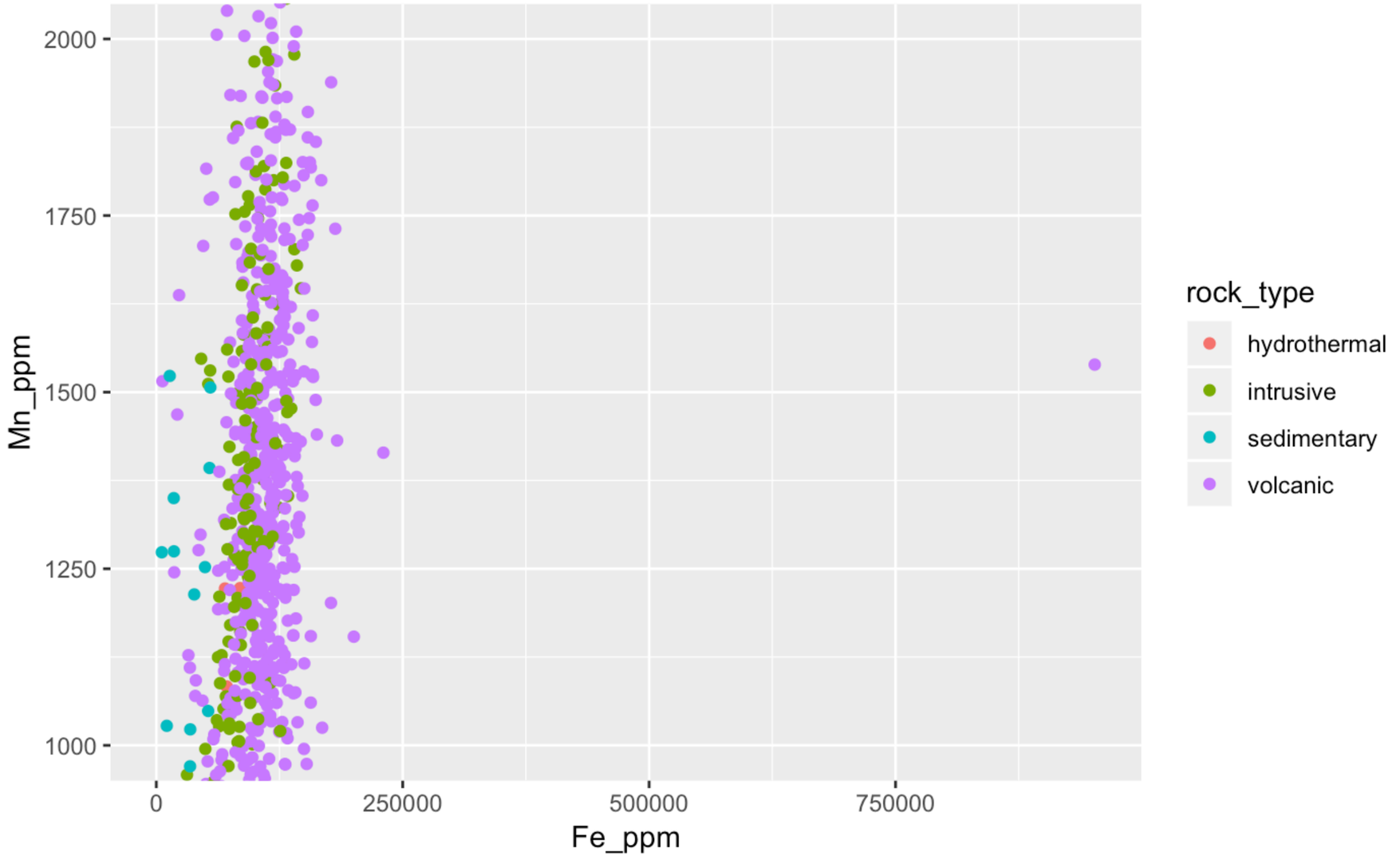



```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type)) +
  facet_wrap(scales = "free")
```

Coordinates



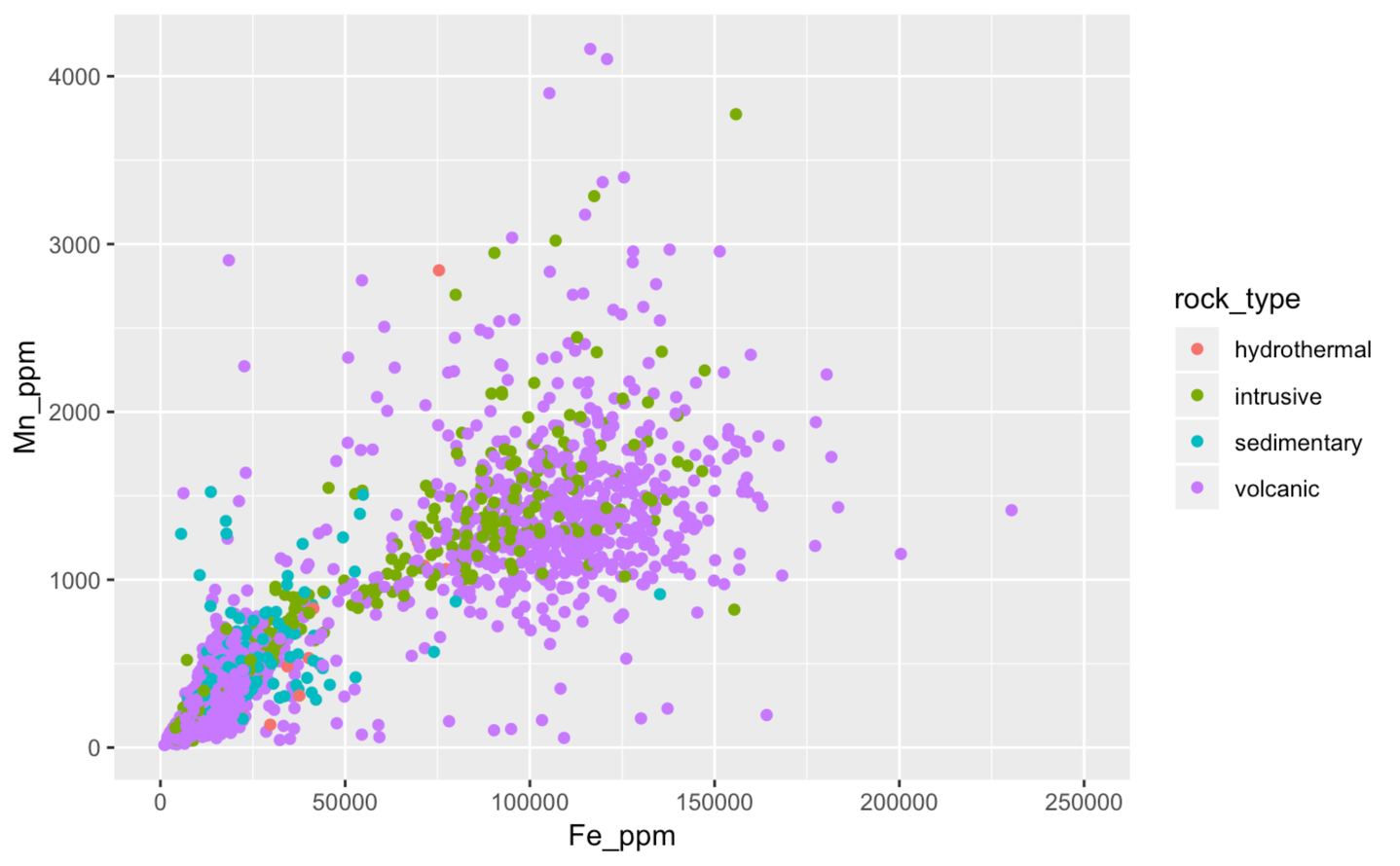

```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type)) +
  coord_cartesian()
```

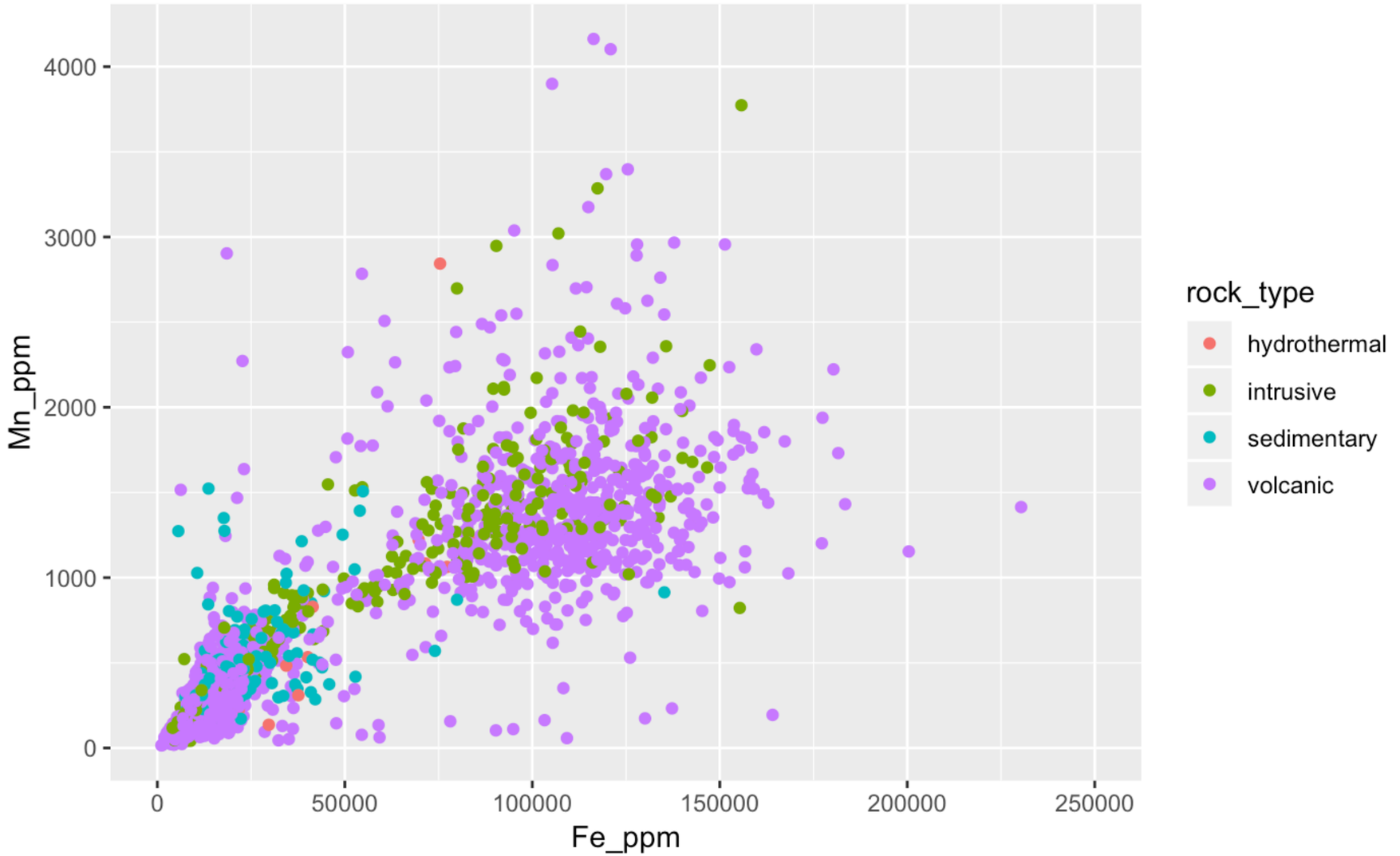



```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type)) +
  coord_cartesian(ylim = c(1000, 2000))
```

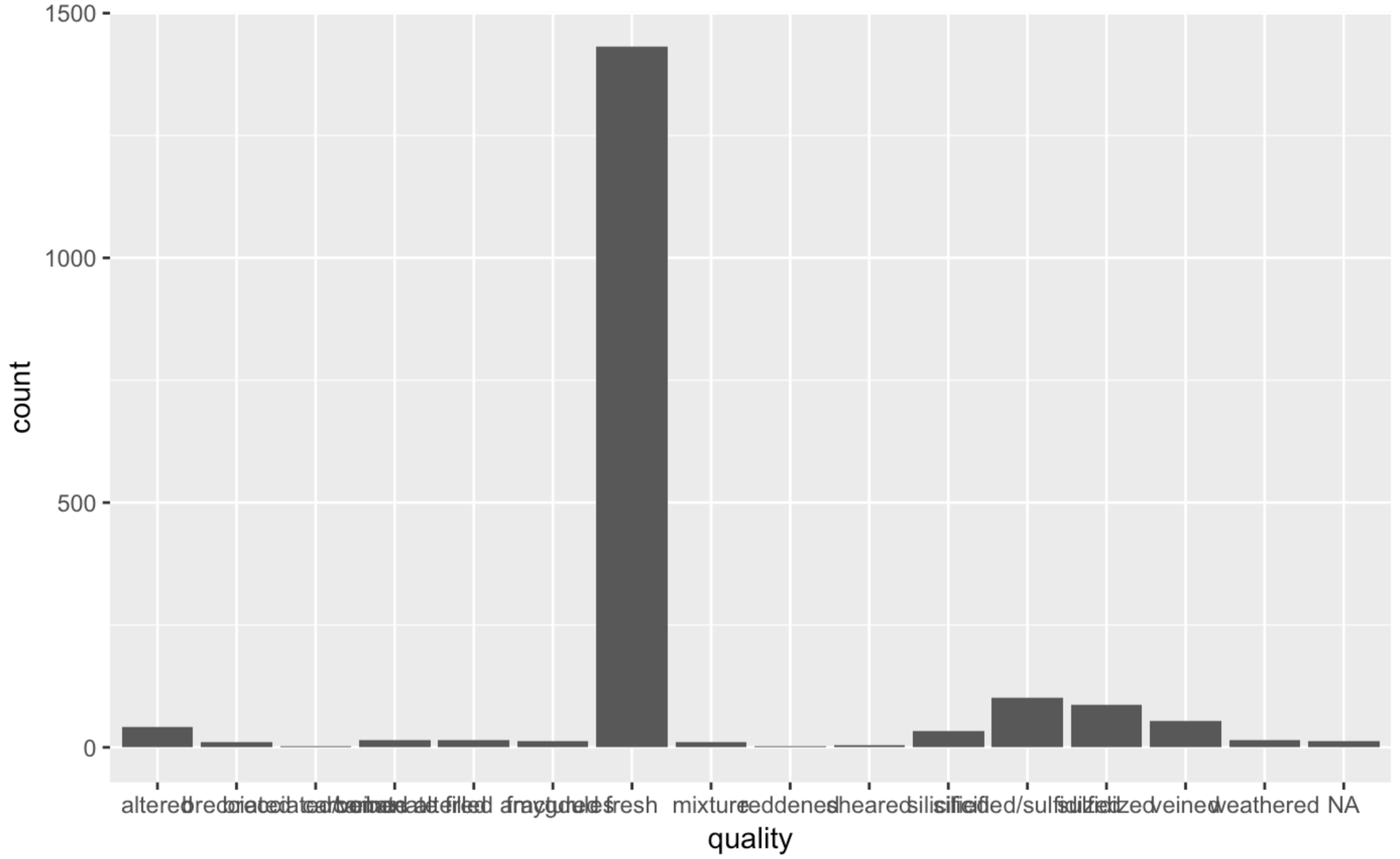
Your Turn 5

Use the `xlim` argument of `coord_cartesian()` to remove the two outlying points with high Fe.

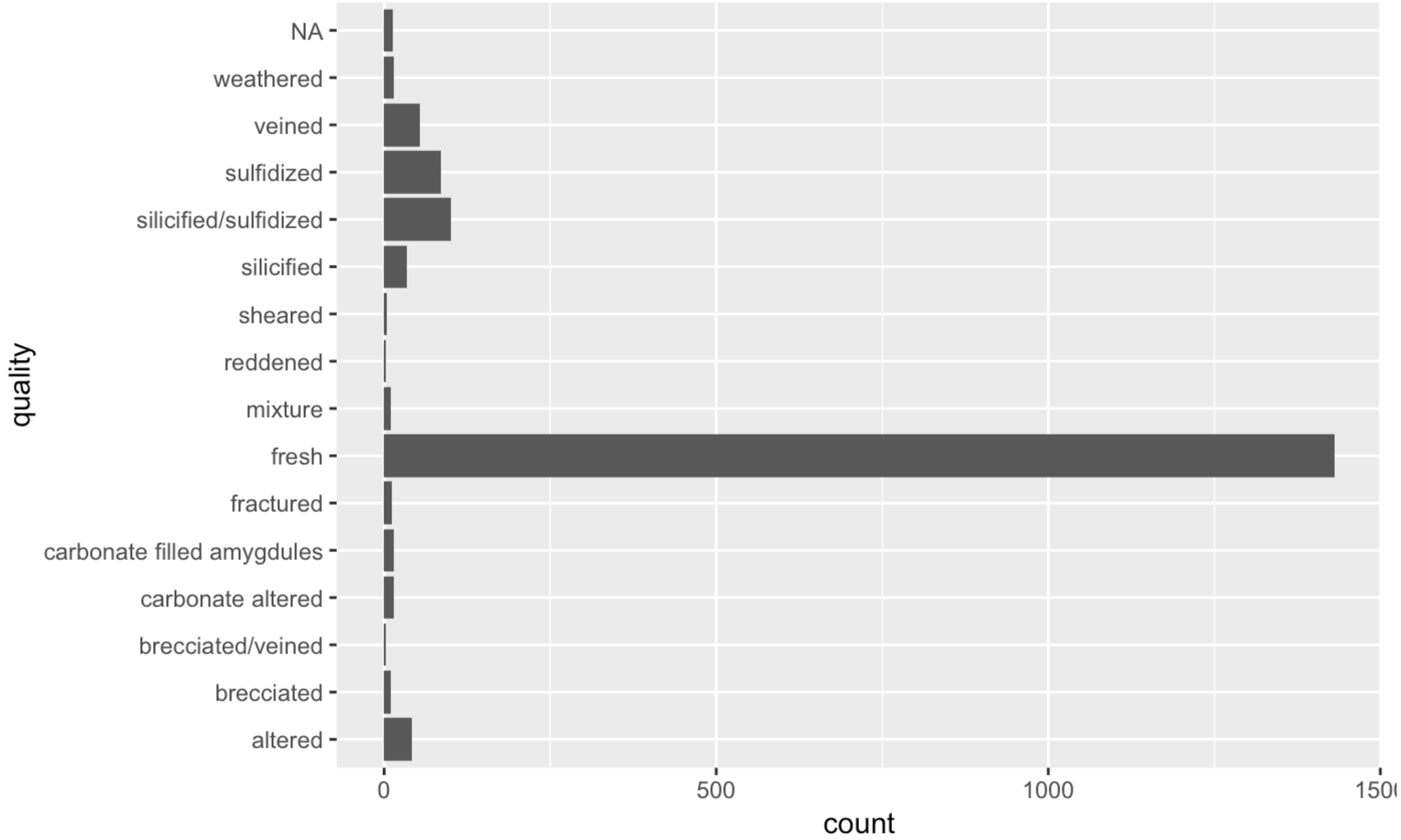





```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type)) +
  coord_cartesian(xlim = c(0, 250000))
```

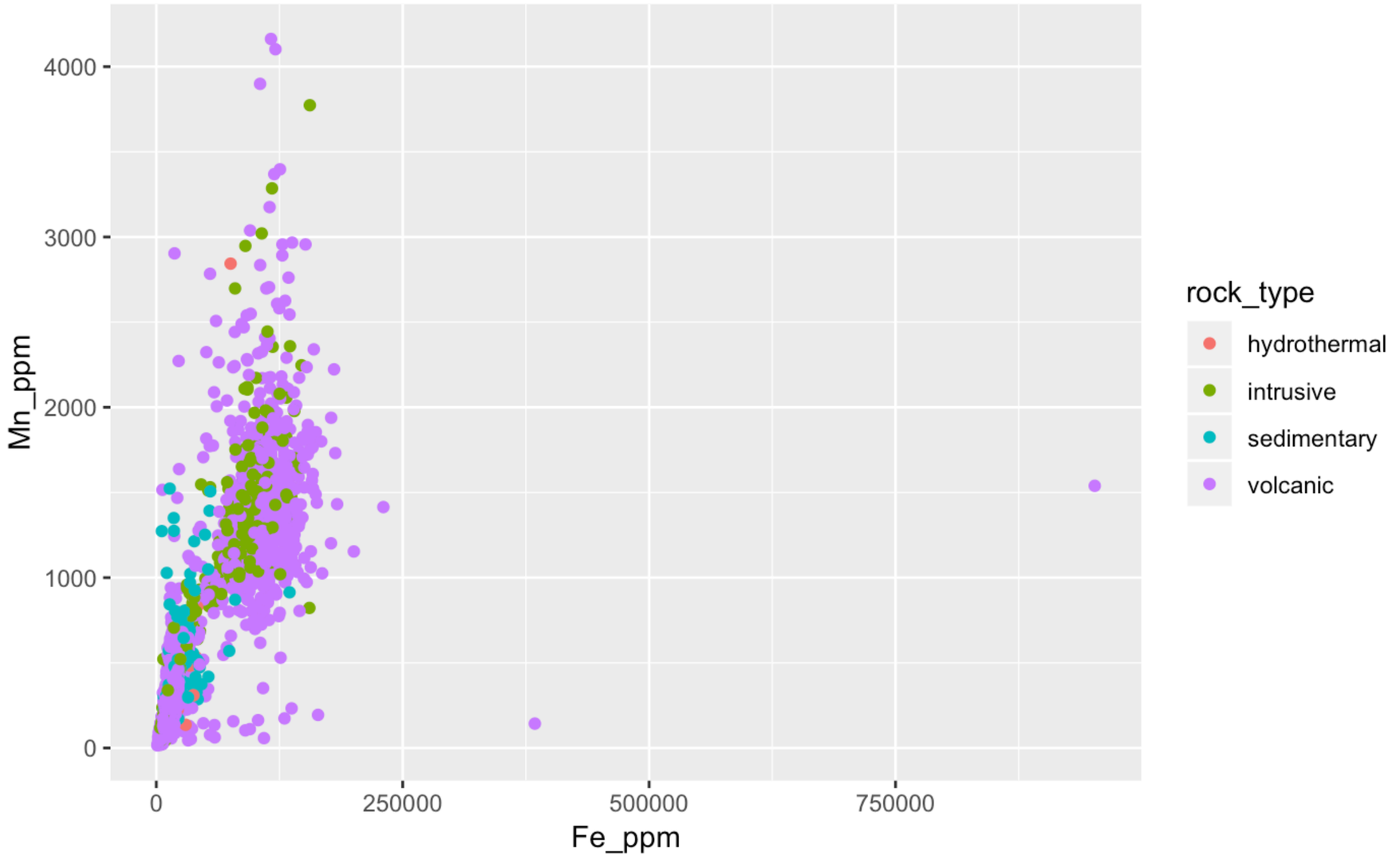



```
ggplot(warwick) +
  geom_bar(aes(x = quality)
```

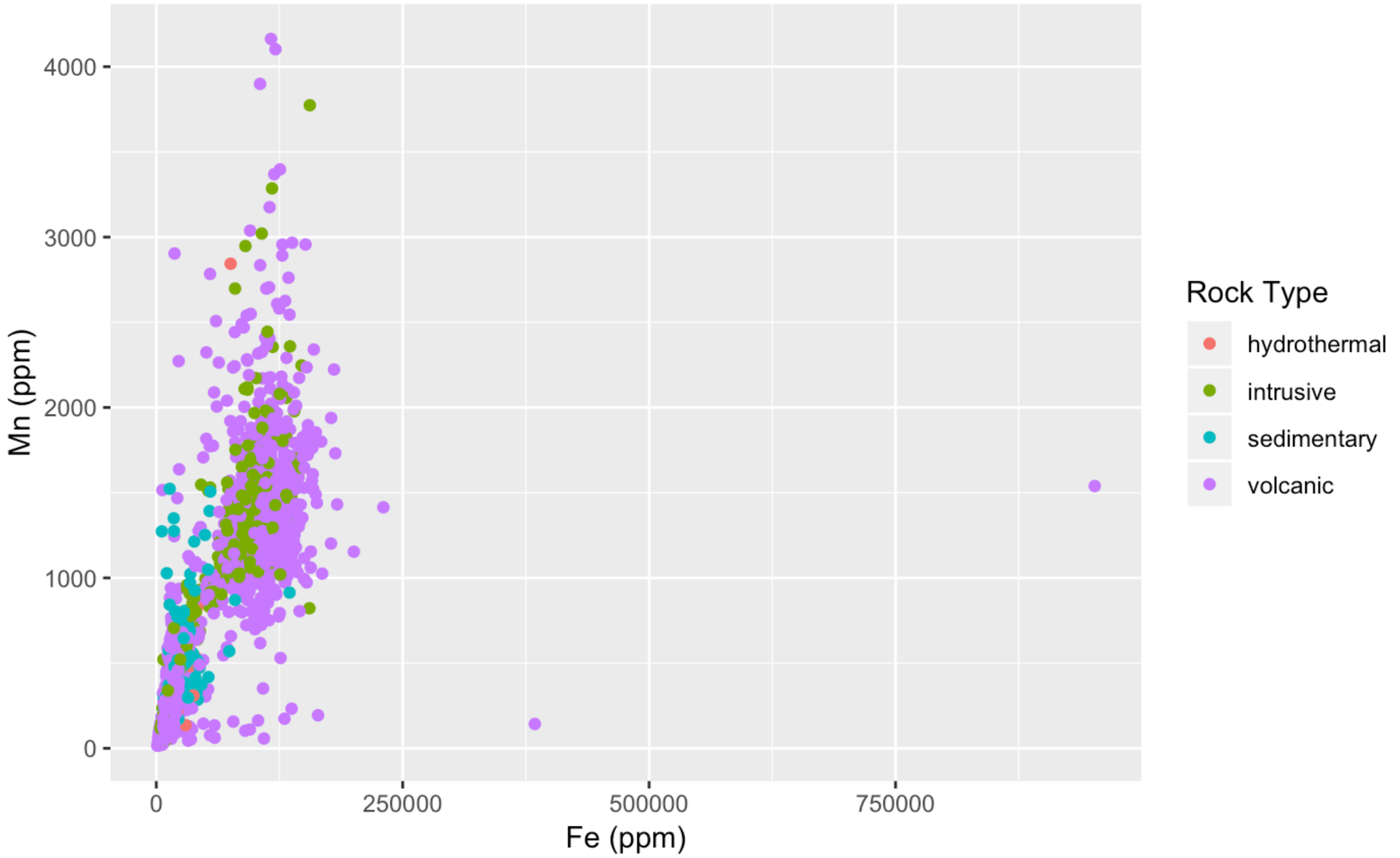



```
ggplot(warwick) +
  geom_bar(aes(x = quality) +
  coord_flip()
```

Labels




```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type))
```

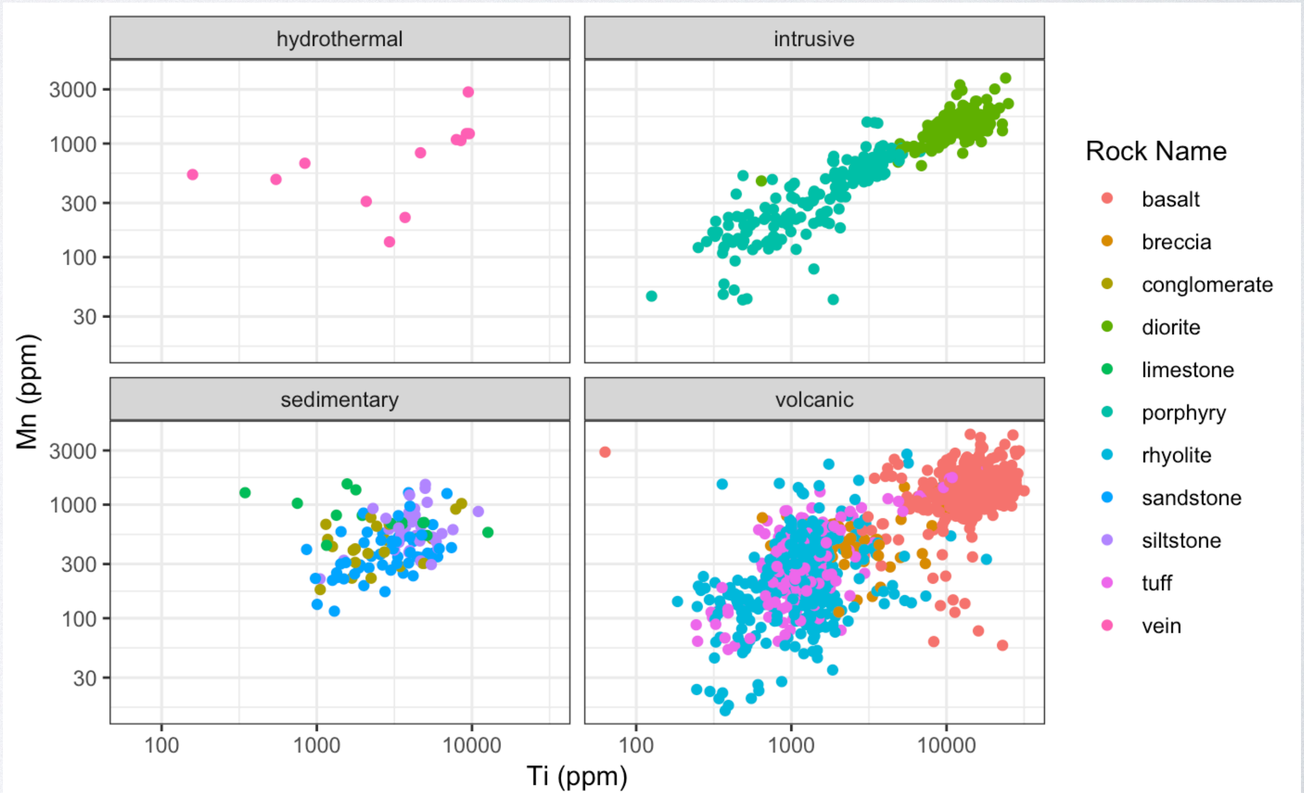



```
ggplot(warwick) +
  geom_point(aes(x = Fe_ppm, y = Mn_ppm, color = rock_type)) +
  labs(x = "Fe (ppm)", y = "Mn (ppm)", color = "Rock Type")
```

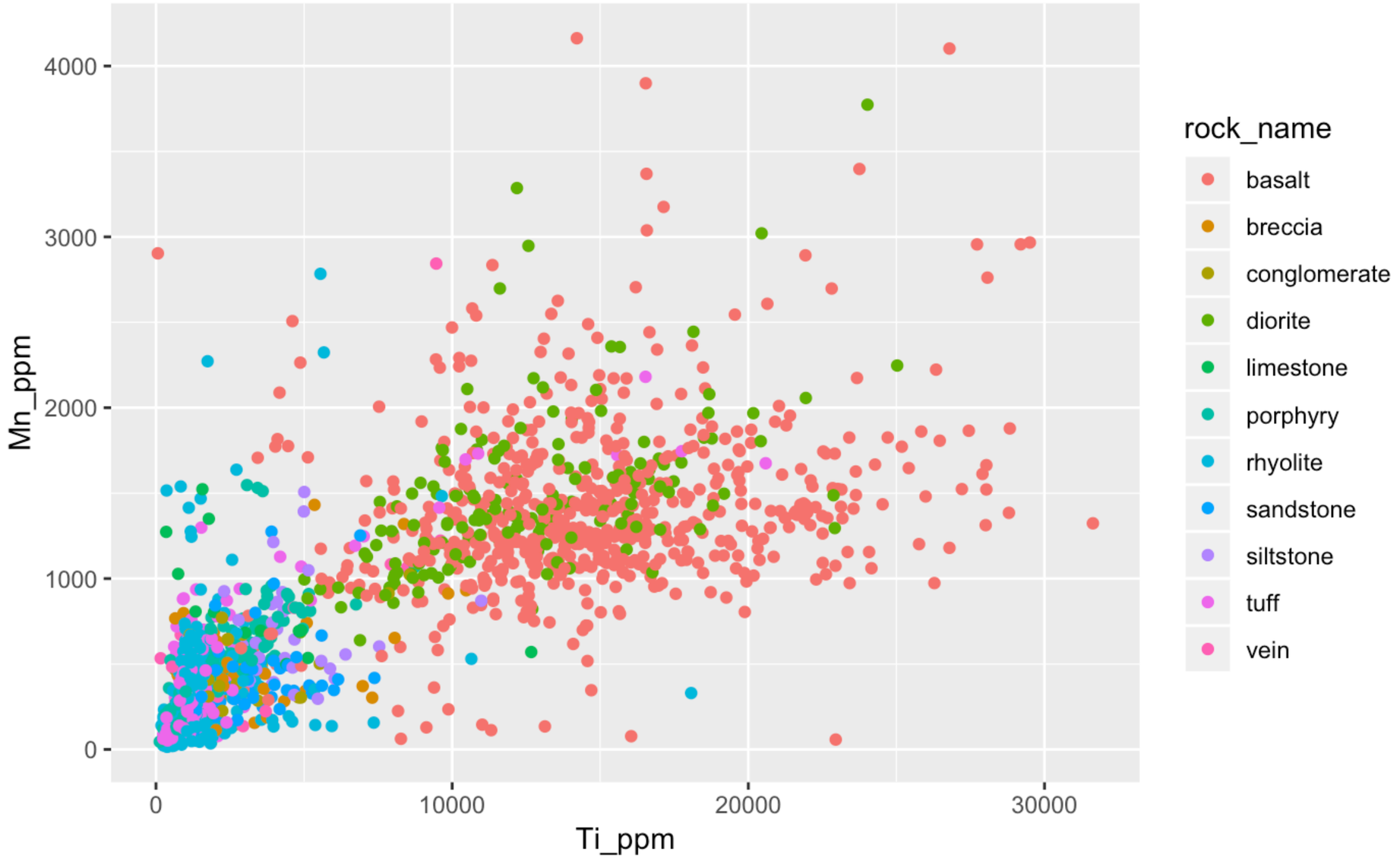
Exam

Create a scatterplot of Ti_ppm and Mn_ppm with:

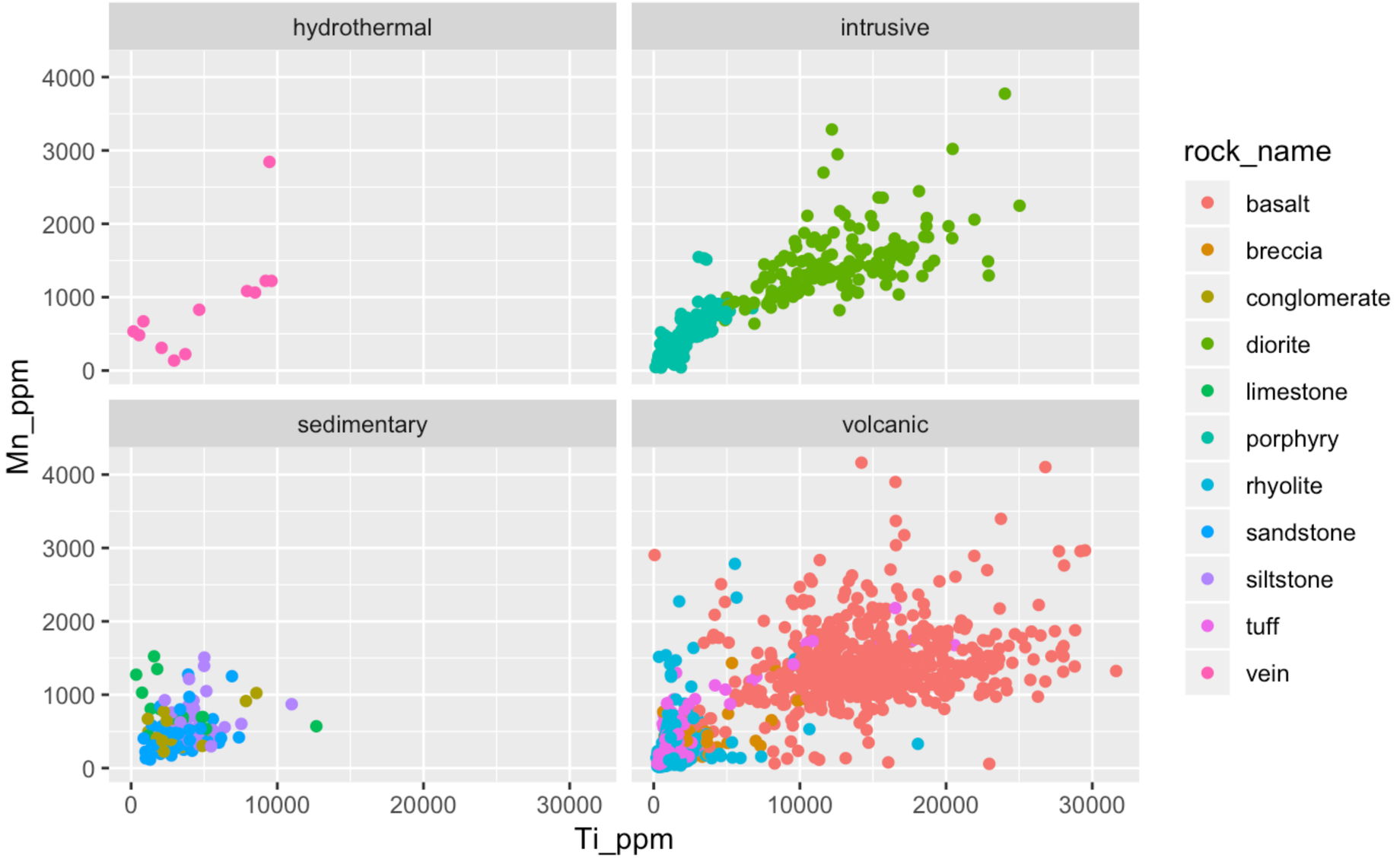
- Points colored by rock_name
- Facetted by rock_type
- Your favourite theme



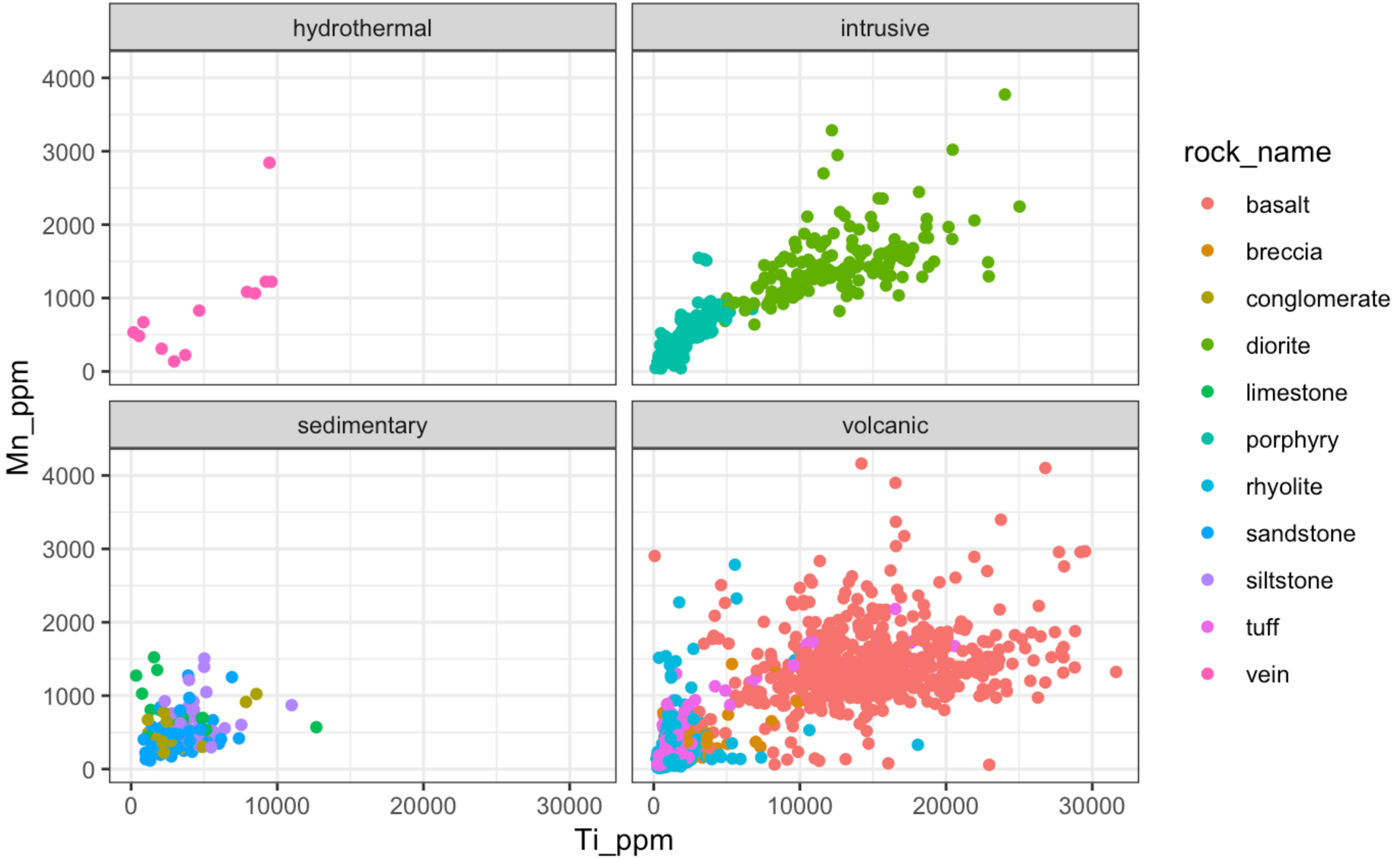
```
ggplot(warwick)
```



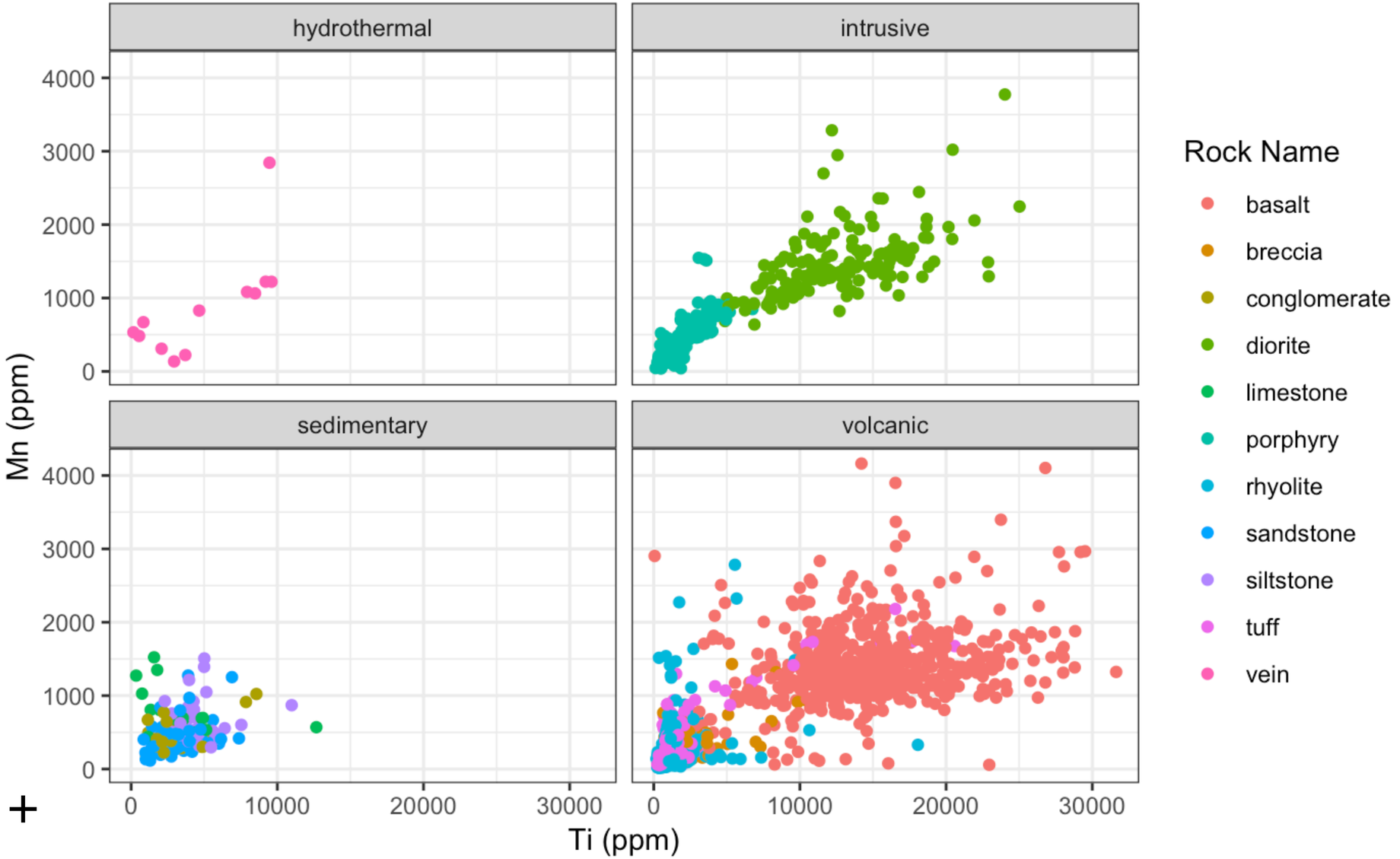

```
ggplot(warwick) +
  geom_point(aes(x = Ti_ppm, y = Mn_ppm, color = rock_name))
```



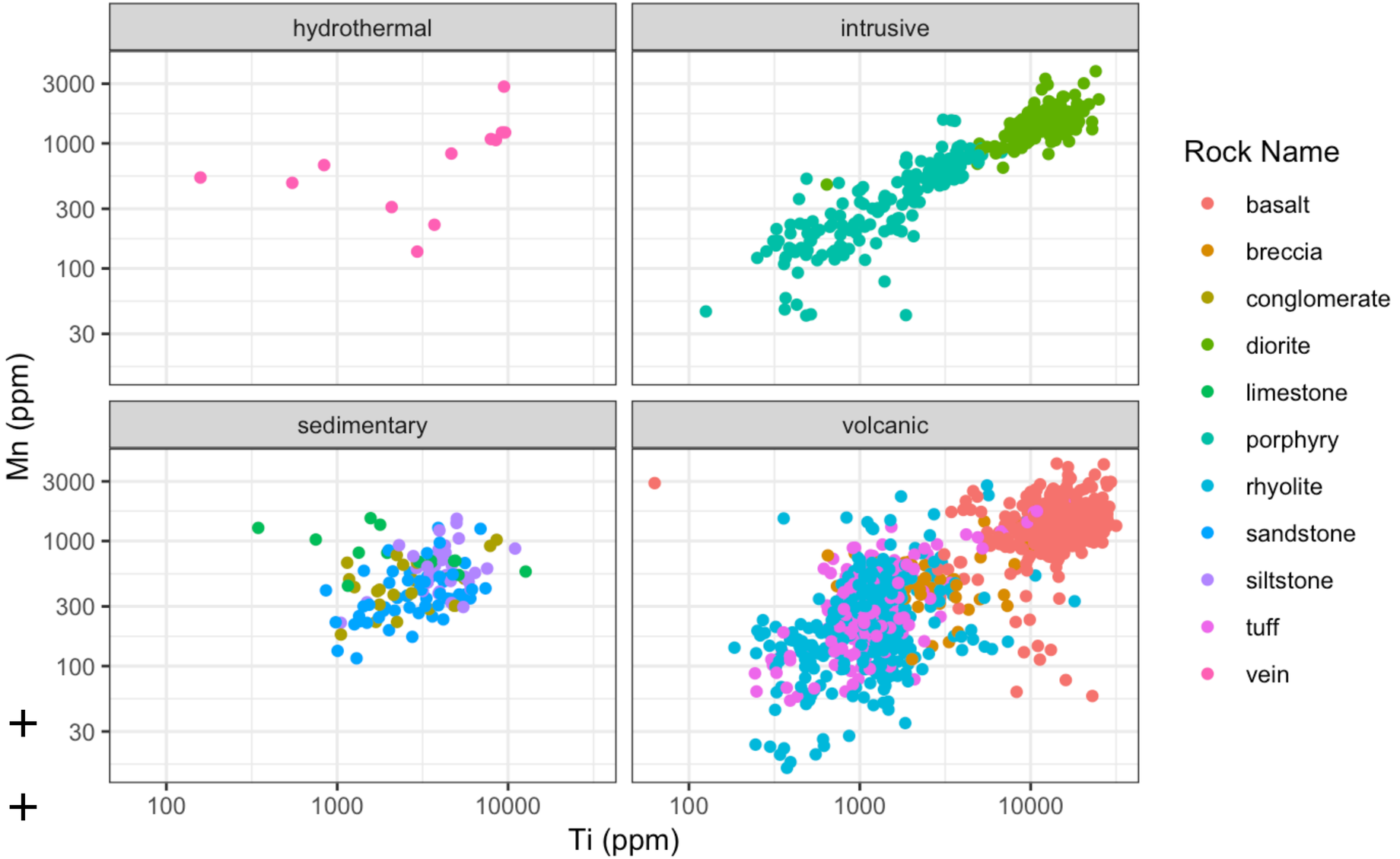

```
ggplot(warwick) +
  geom_point(aes(x = Ti_ppm, y = Mn_ppm, color = rock_name)) +
  facet_wrap(~rock_type)
```



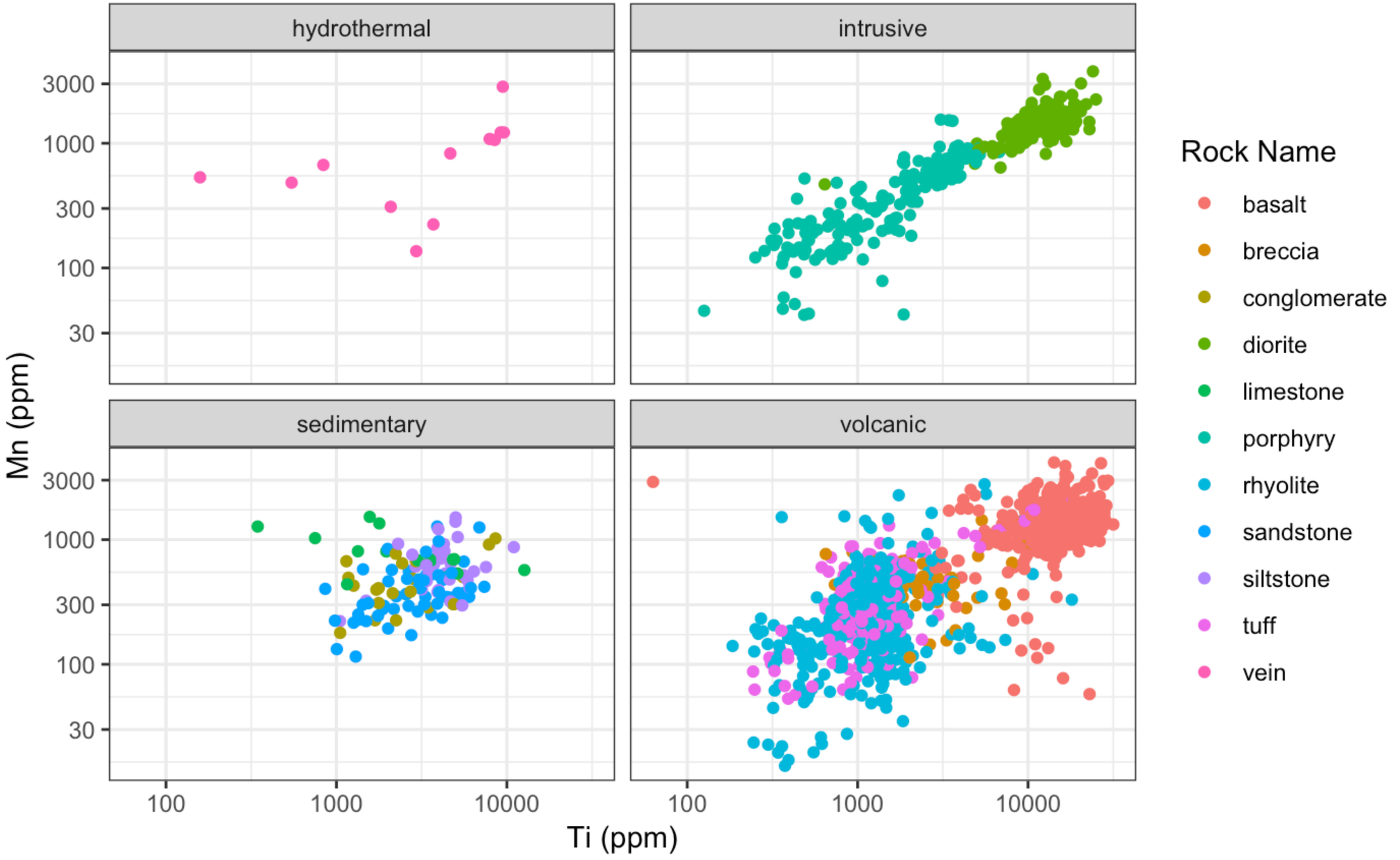

```
ggplot(warwick) +
  geom_point(aes(x = Ti_ppm, y = Mn_ppm, color = rock_name)) +
  facet_wrap(~rock_type) +
  theme_bw()
```




```
ggplot(warwick) +
  geom_point(...) +
  facet_wrap(~rock_type) +
  theme_bw() +
  labs(x = "Ti (ppm)", y = "Mn (ppm)", color = "Rock Name")
```




```
ggplot(warwick) +
  geom_point(...) +
  facet_wrap(...) +
  theme_bw() +
  labs(x = "Ti (ppm)", y = "Mn (ppm)", color = "Rock Name") +
  scale_x_log10() + scale_y_log10()
```


hydrothermal
intrusive
sedimentary
volcanic
Mn (ppm)
3000
1000
300
100
30
100
1000
10000
Ti (ppm)
Rock Name
basalt
breccia
conglomerate
diorite
limestone
porphyry
rhyolite
sandstone
siltstone
tuff
vein

Visualize (More) Data with