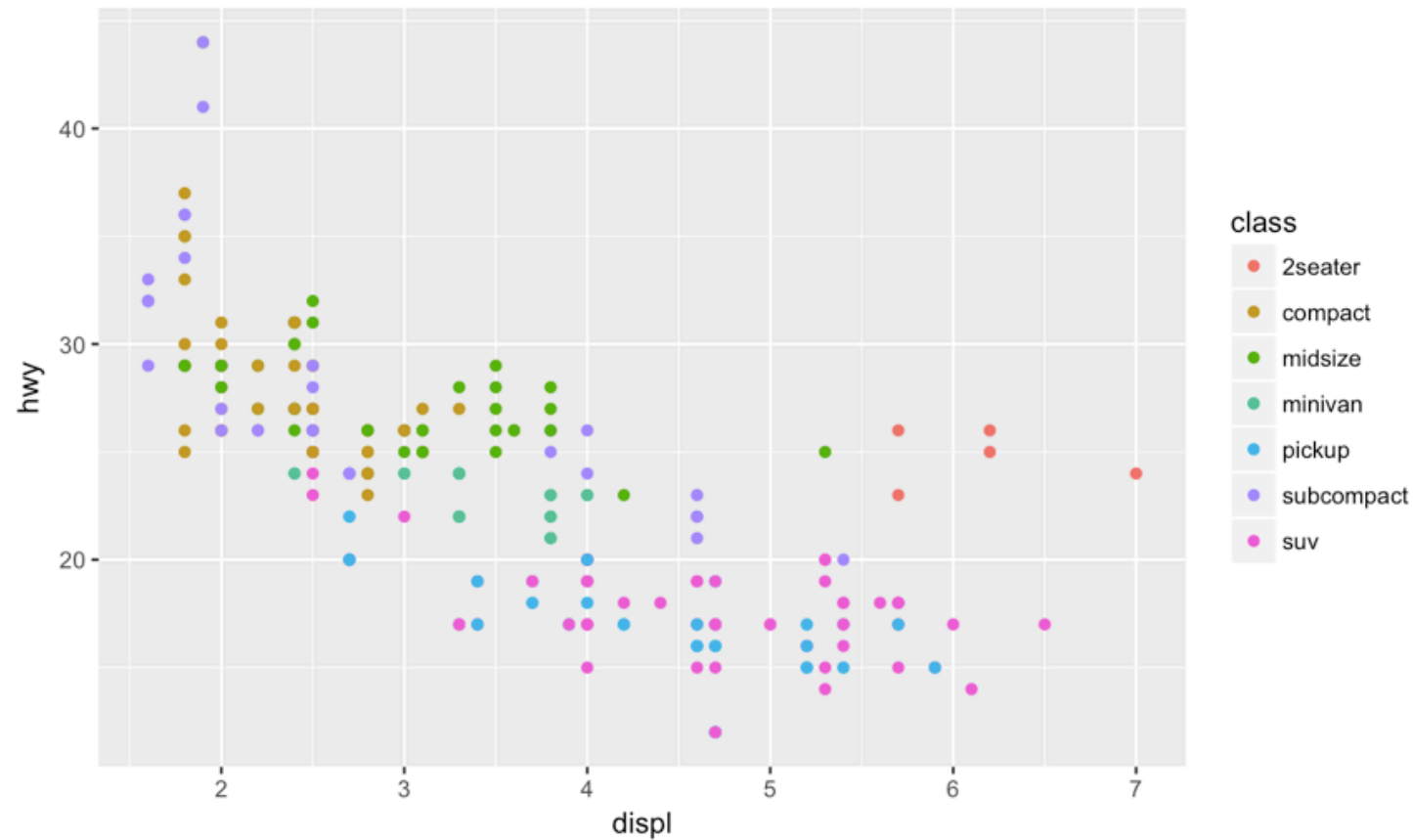


Scales

tweaking scales

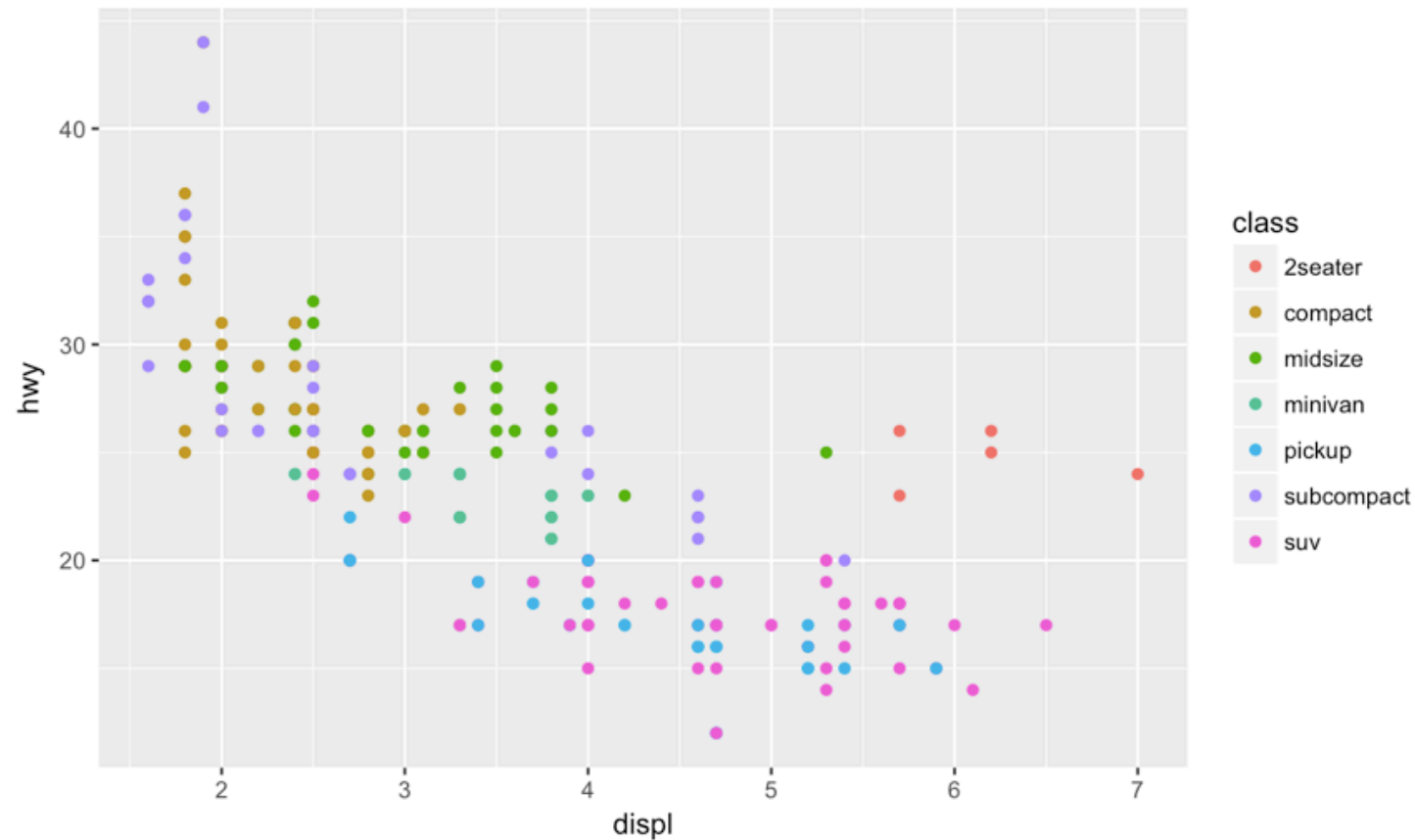
```
ggplot(mpg, aes(displ, hwy)) +  
  geom_point(aes(color = class))
```

when a plot like this is made, scales are added behind the scenes



```
ggplot(mpg, aes(displ, hwy)) +  
  geom_point(aes(color = class)) +  
  scale_x_continuous() +  
  scale_y_continuous() +  
  scale_color_discrete()
```

scales control the mapping from
data values to things you can see



Common syntax

```
scale_<AES>_<DATA TYPE>(…)
```

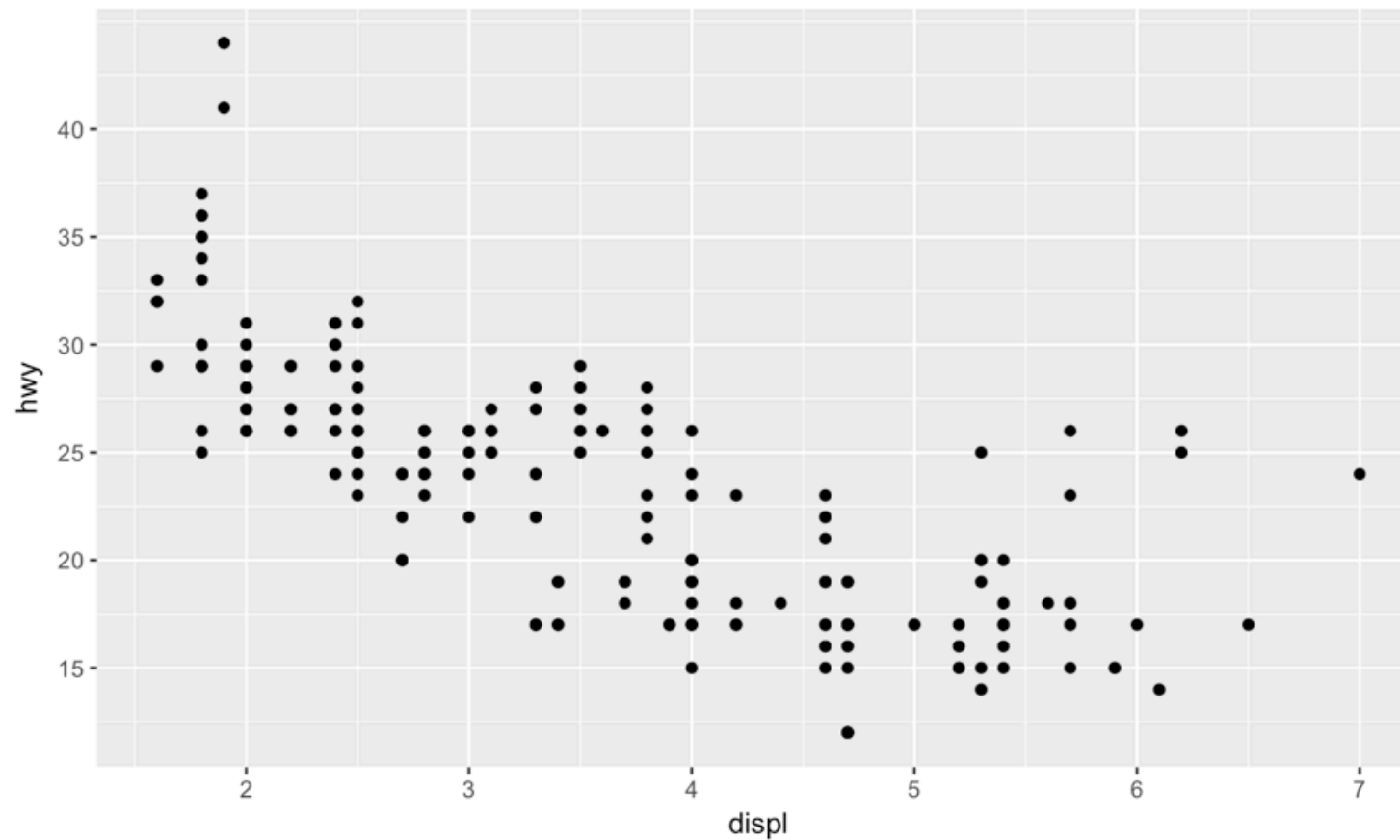
aesthetic the scale
corresponds to

data type variable being
mapped to aesthetic:
continuous, discrete,
date, or datetime

arguments depend
on the scale

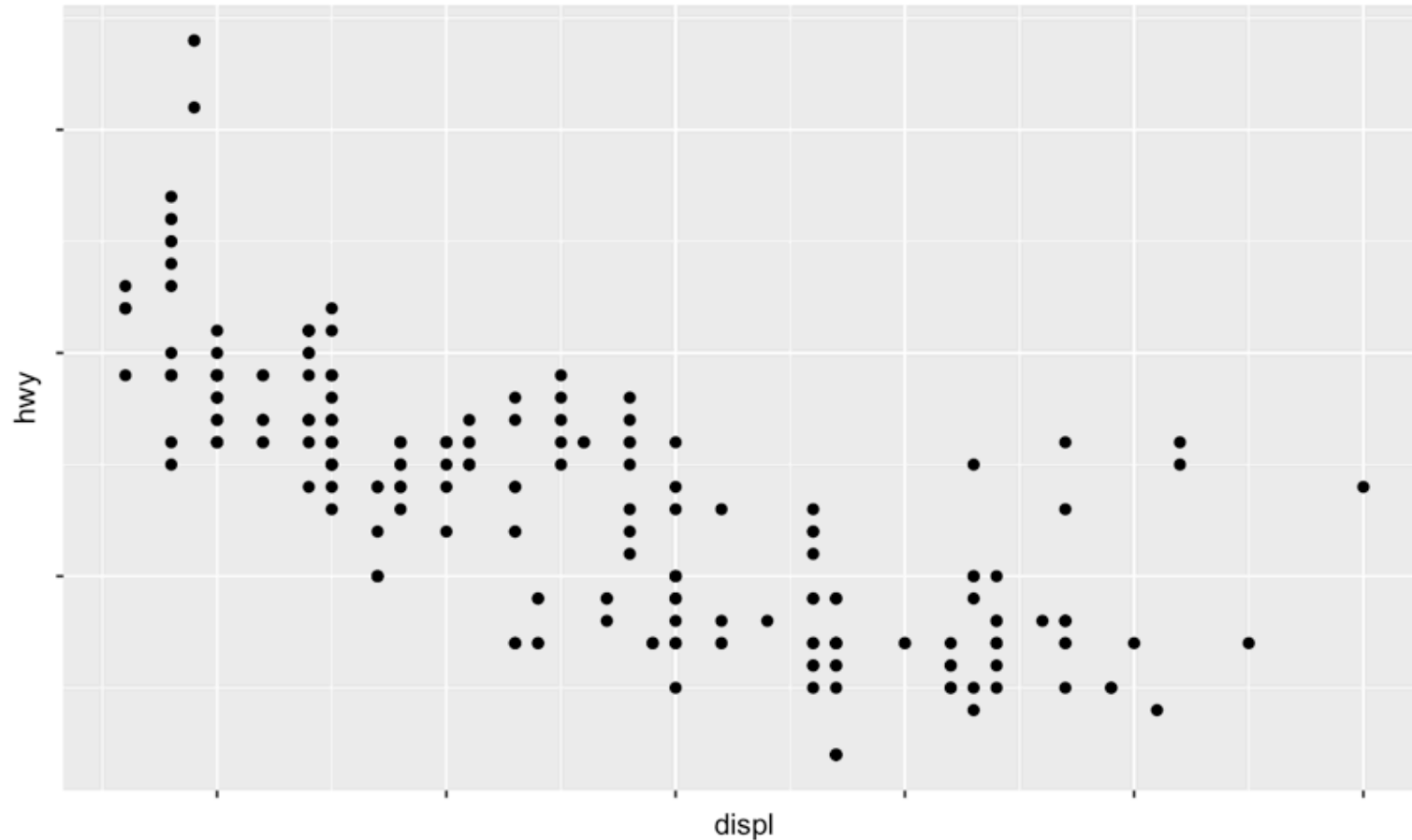
```
ggplot(mpg, aes(displ, hwy)) +  
  geom_point() +  
  scale_y_continuous(breaks = seq(15, 40, by = 5))
```

set what values appear along
the axis

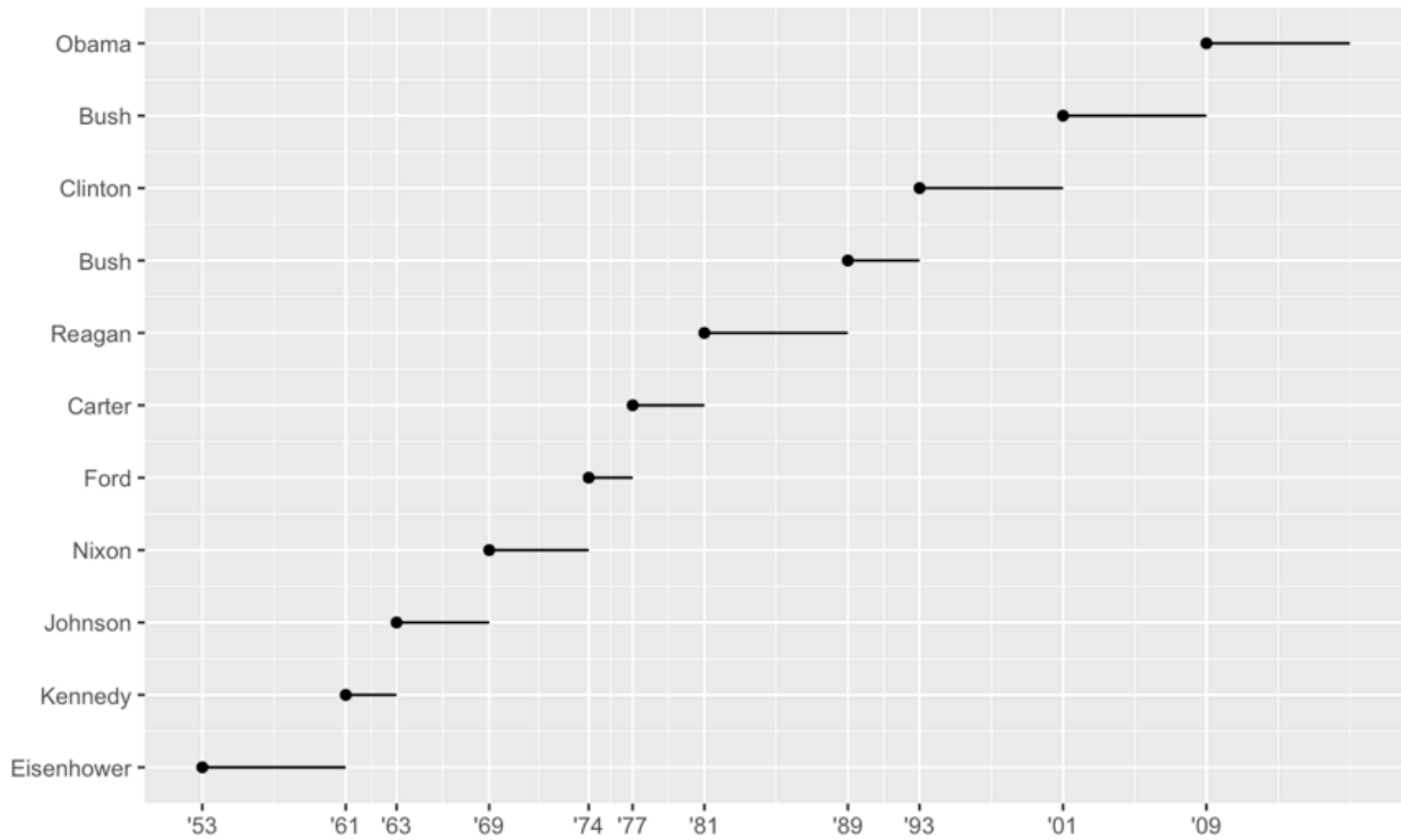


```
ggplot(mpg, aes(displ, hwy)) +  
  geom_point() +  
  scale_x_continuous(labels = NULL) +  
  scale_y_continuous(labels = NULL)
```

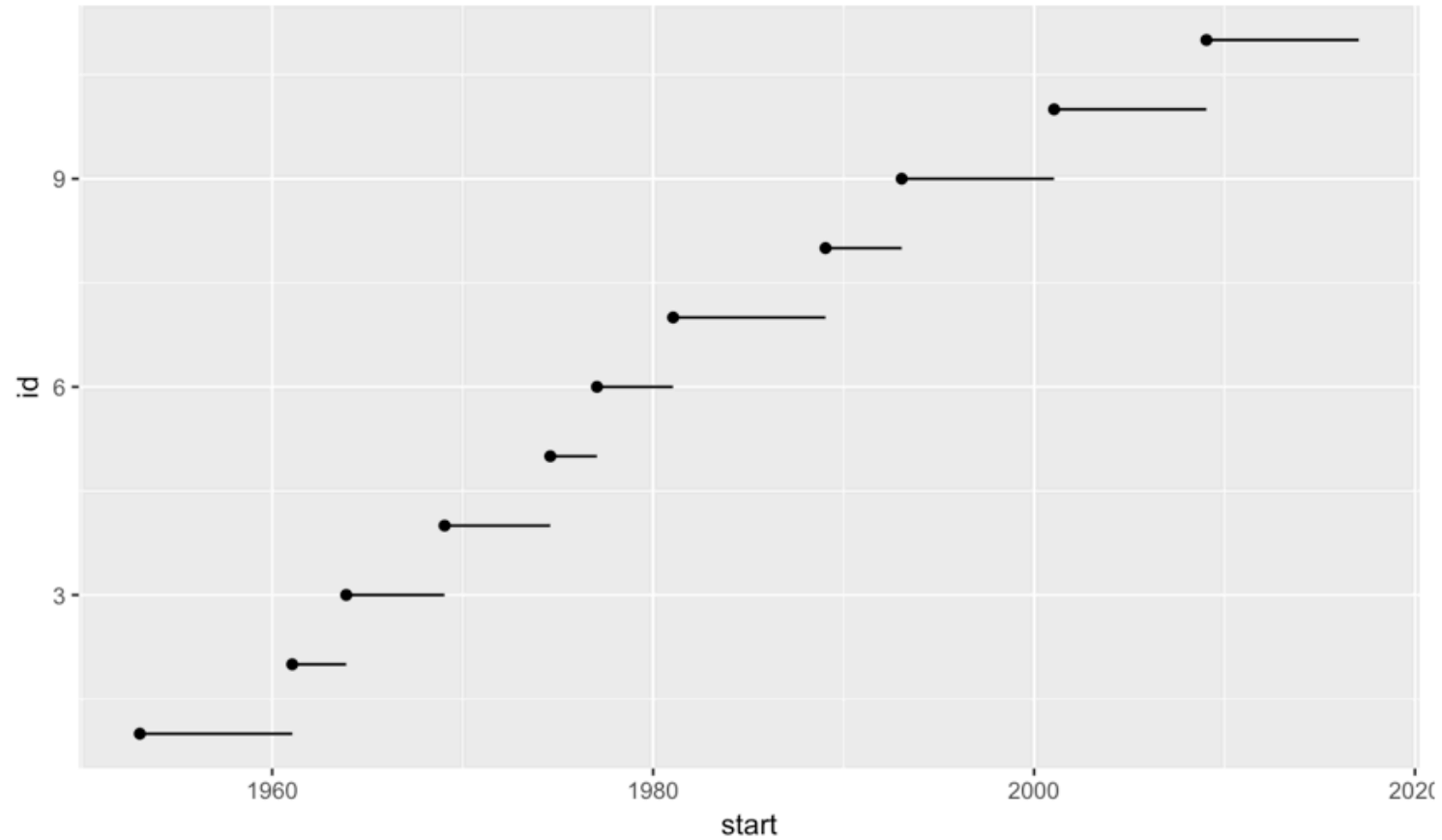
alter the text that appears along
the axis



```
## # A tibble: 11 x 4
##   name      start      end      party
##   <chr>    <date>    <date>    <chr>
## 1 Eisenhower 1953-01-20 1961-01-20 Republican
## 2 Kennedy    1961-01-20 1963-11-22 Democratic
## 3 Johnson    1963-11-22 1969-01-20 Democratic
## 4 Nixon      1969-01-20 1974-08-09 Republican
## 5 Ford       1974-08-09 1977-01-20 Republican
## 6 Carter     1977-01-20 1981-01-20 Democratic
## 7 Reagan     1981-01-20 1989-01-20 Republican
## 8 Bush       1989-01-20 1993-01-20 Republican
## 9 Clinton   1993-01-20 2001-01-20 Democratic
## 10 Bush      2001-01-20 2009-01-20 Republican
## 11 Obama     2009-01-20 2017-01-20 Democratic
```

```
presidential %>%  
  mutate(id = row_number()) %>%  
  ggplot(aes(start, id)) +  
  geom_point() +  
  geom_segment(aes(xend = end, yend = id))
```



```
presidential %>%
```

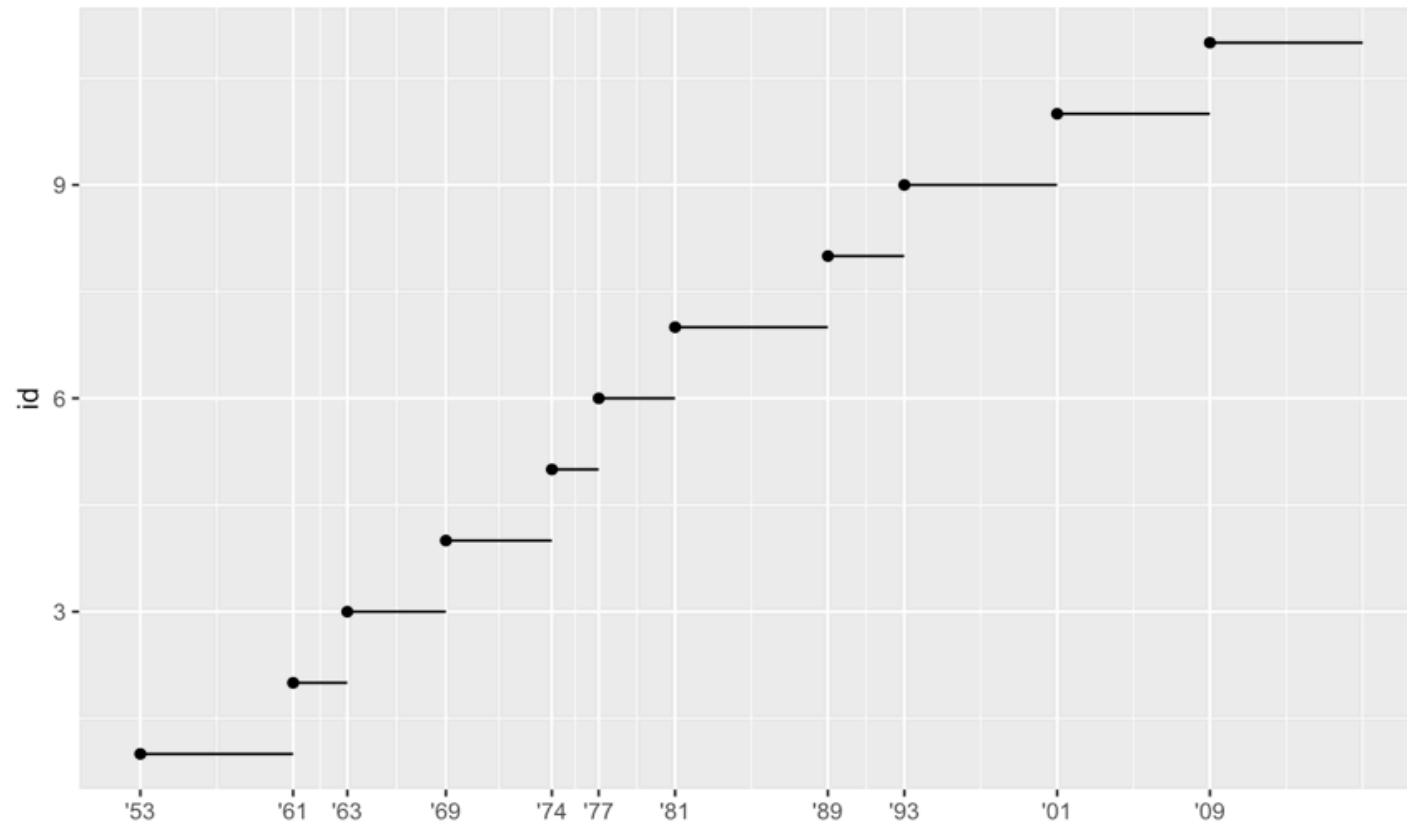
```
  mutate(id = row_number()) %>%
```

```
  ggplot(aes(start, id)) +
```

```
  geom_point() +
```

```
  geom_segment(aes(xend = end, yend = id)) +
```

```
  scale_x_date(NULL, breaks = presidential$start, date_labels = "'%y")
```



```
presidential %>%
```

```
  mutate(id = row_number()) %>%
```

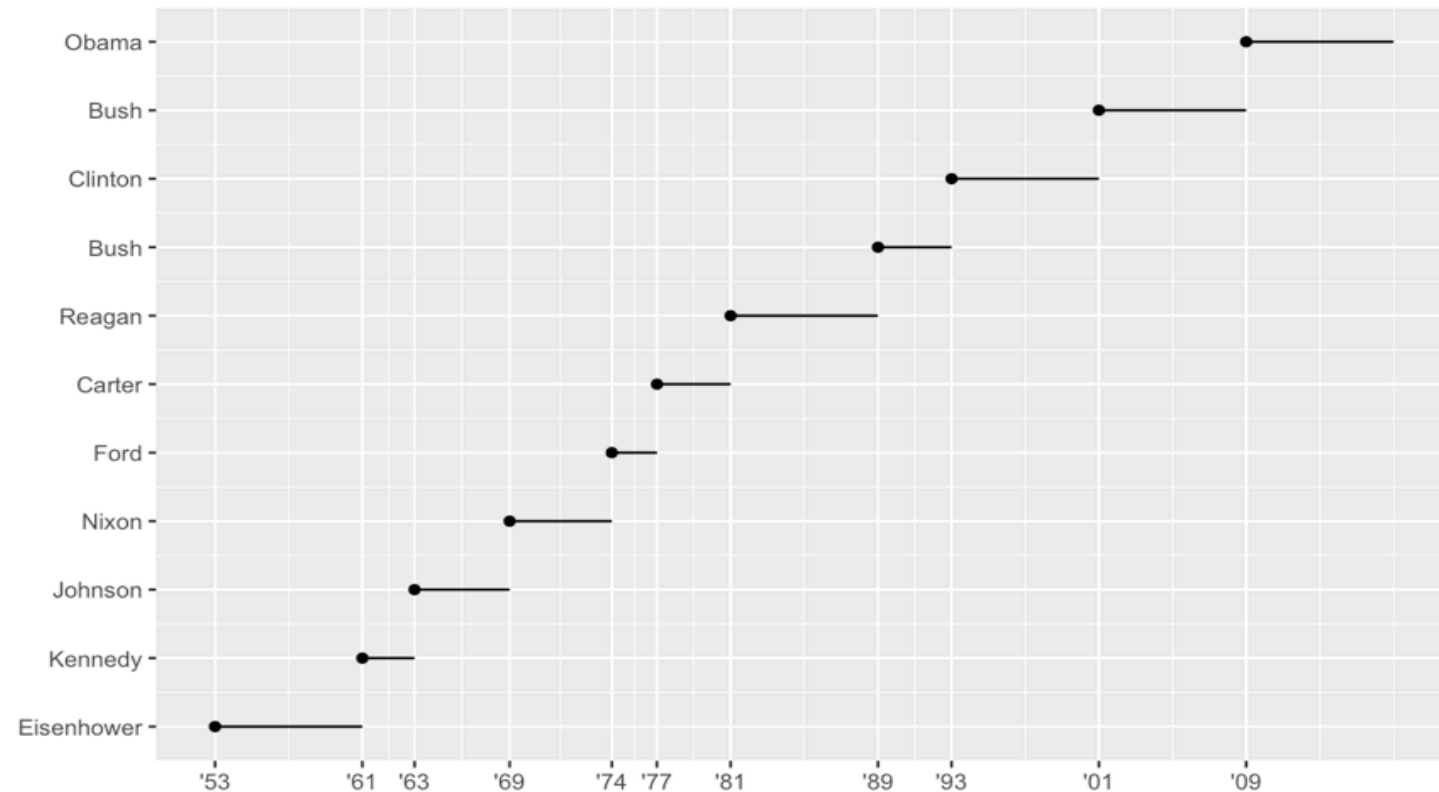
```
  ggplot(aes(start, id)) +
```

```
  geom_point() +
```

```
  geom_segment(aes(xend = end, yend = id)) +
```

```
  scale_x_date(NULL, breaks = presidential$start, date_labels = "'%y") +
```

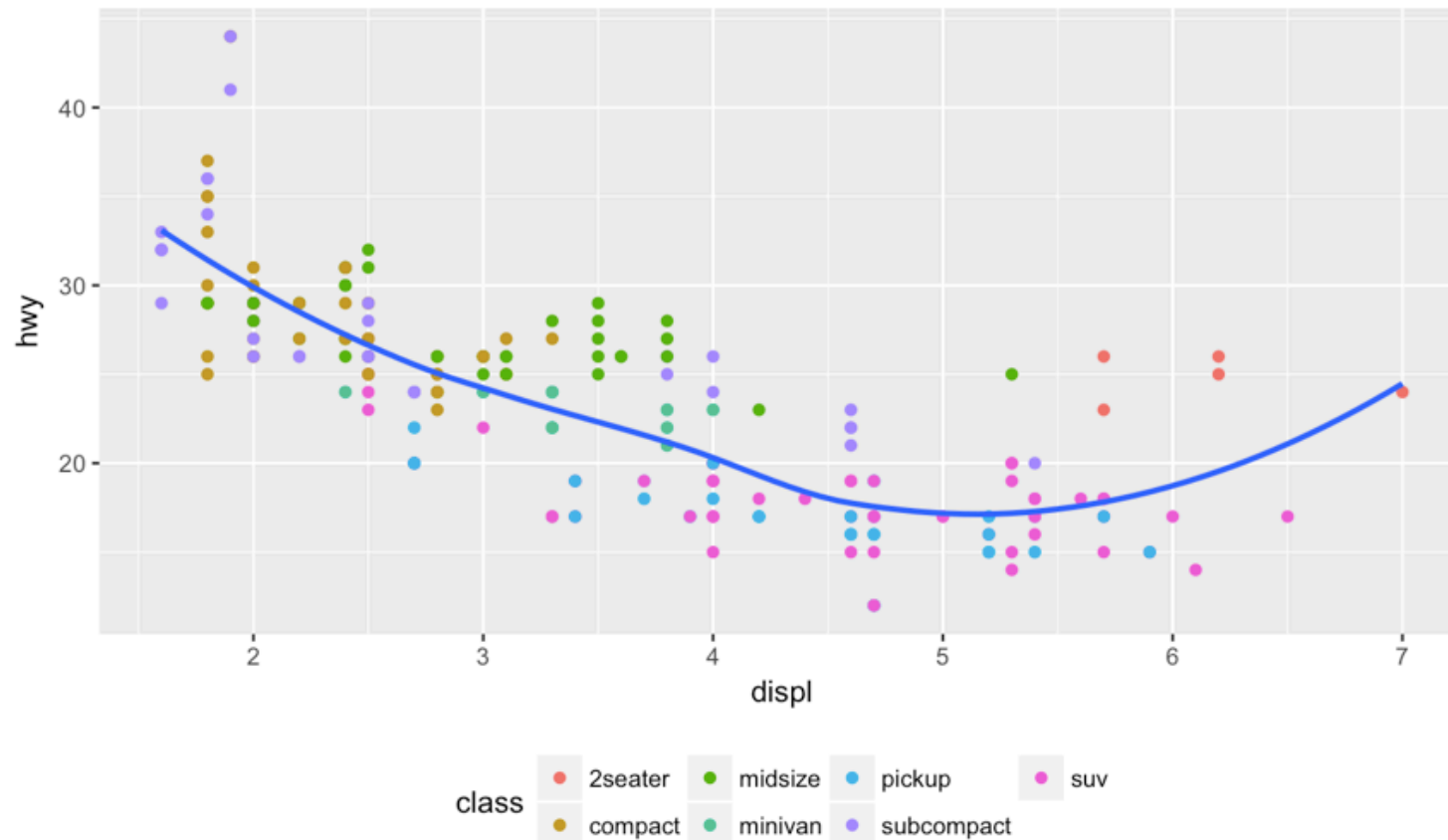
```
  scale_y_continuous(NULL, breaks = seq(1, 11, by = 1), labels = presidential$name)
```



legend layout

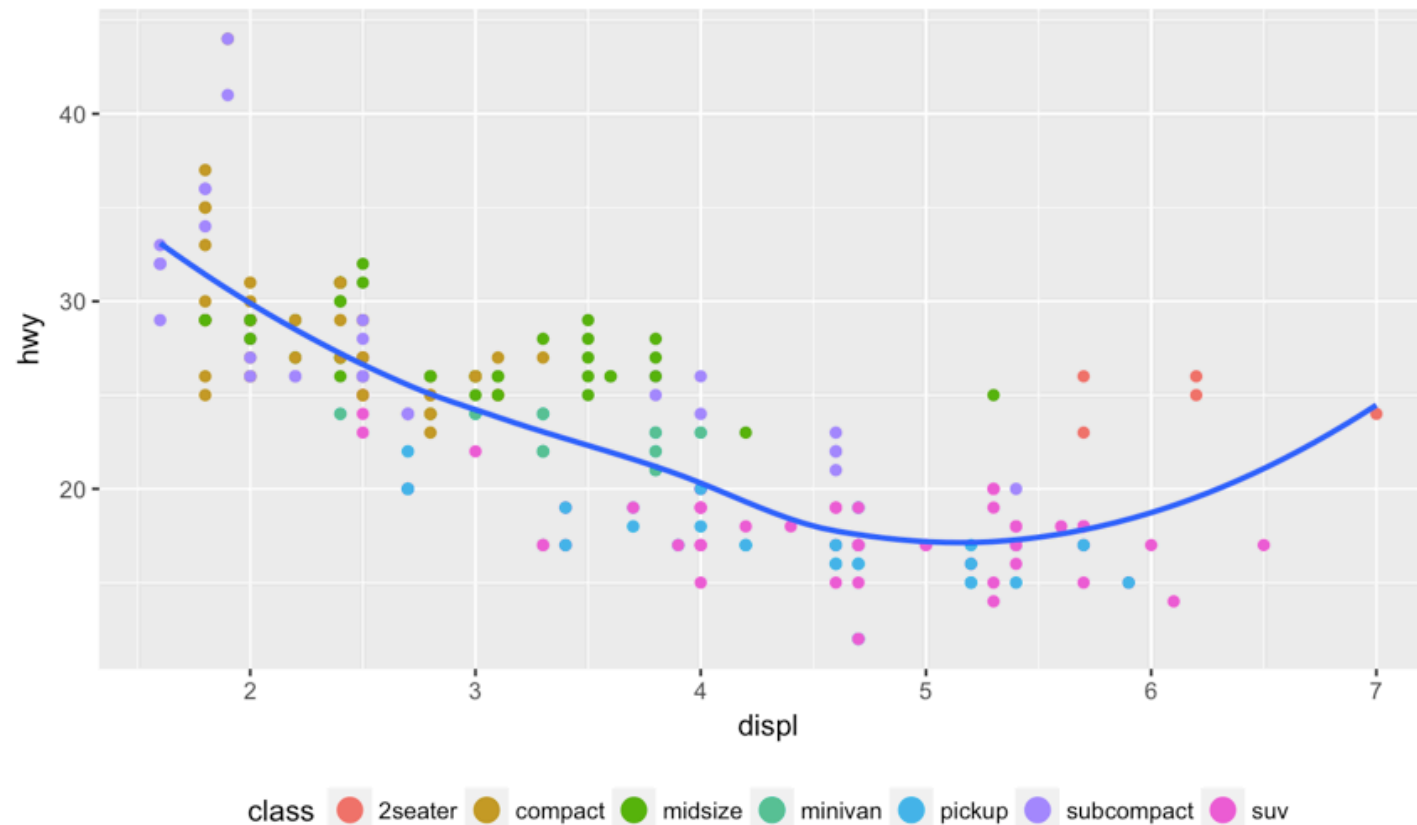
```
ggplot(mpg, aes(displ, hwy)) +  
  geom_point(aes(color = class)) +  
  geom_smooth(se = FALSE) +  
  theme(legend.position = "bottom")
```

theme() controls the non-data
parts of the plot



```
ggplot(mpg, aes(displ, hwy)) +  
  geom_point(aes(color = class)) +  
  geom_smooth(se = FALSE) +  
  theme(legend.position = "bottom") +  
  guides(color = guide_legend(nrow = 1,  
    override.aes = list(size = 4)))
```

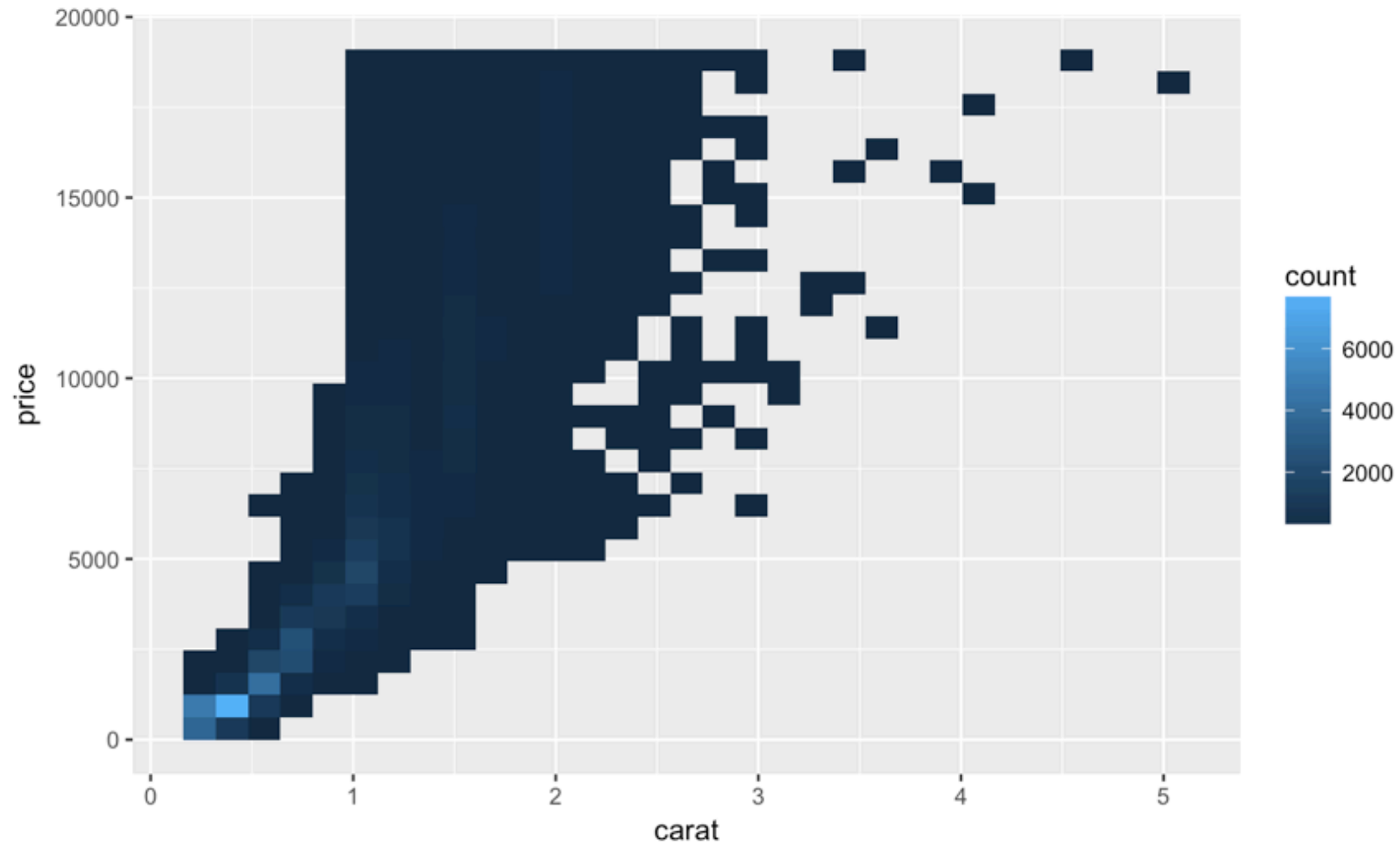
control the display of individual legends with `guides()` and `guide_legend()`



replacing scales


```
diamonds %>%  
  ggplot(aes(carat, price)) +  
  geom_bin2d()
```

recall carat and price had a
nonlinear relationship

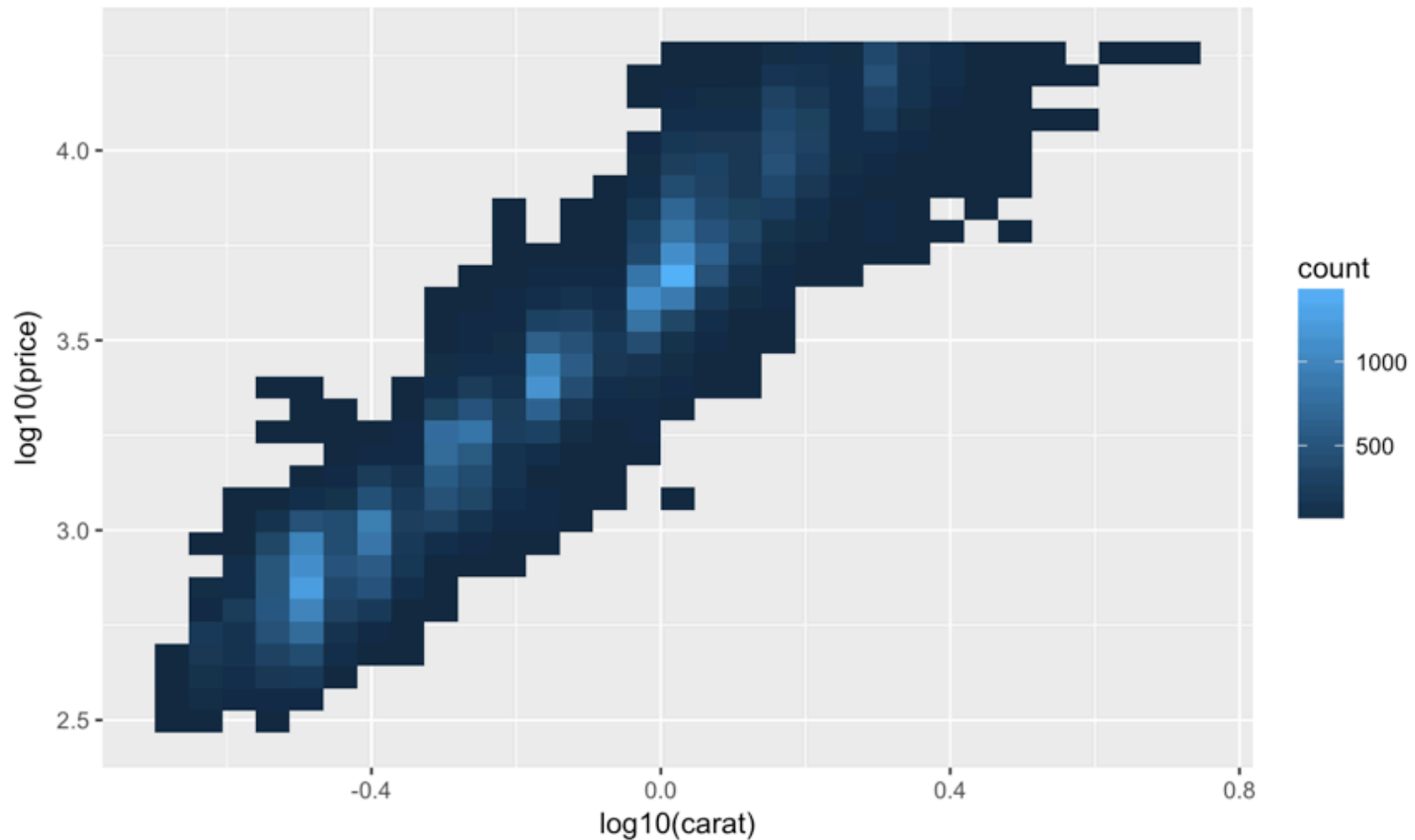


```
diamonds %>%
```

```
  ggplot(aes(log10(carat), log10(price))) +
```

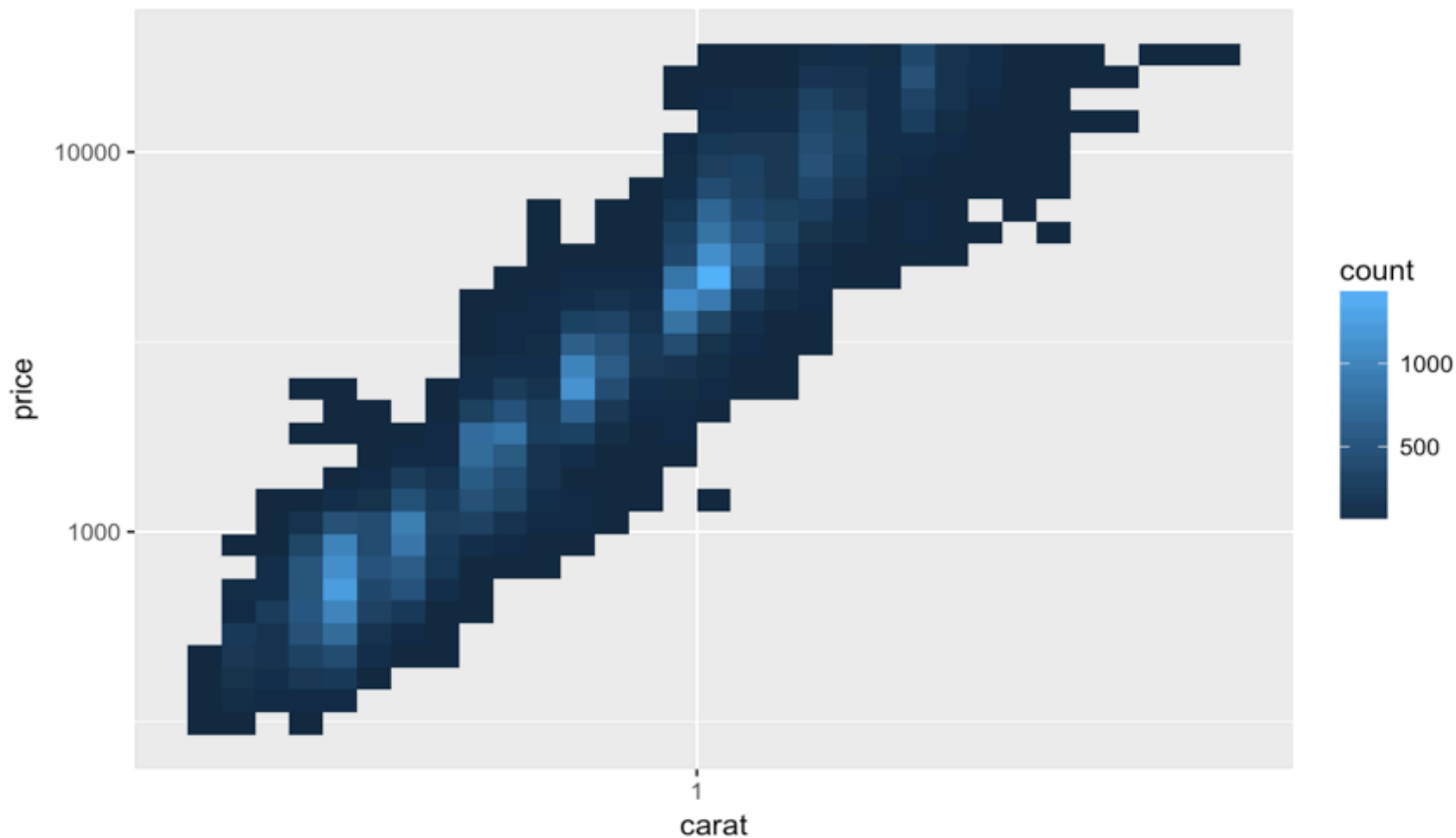
```
  geom_bin2d()
```

transforming the data creates a
more linear relationship

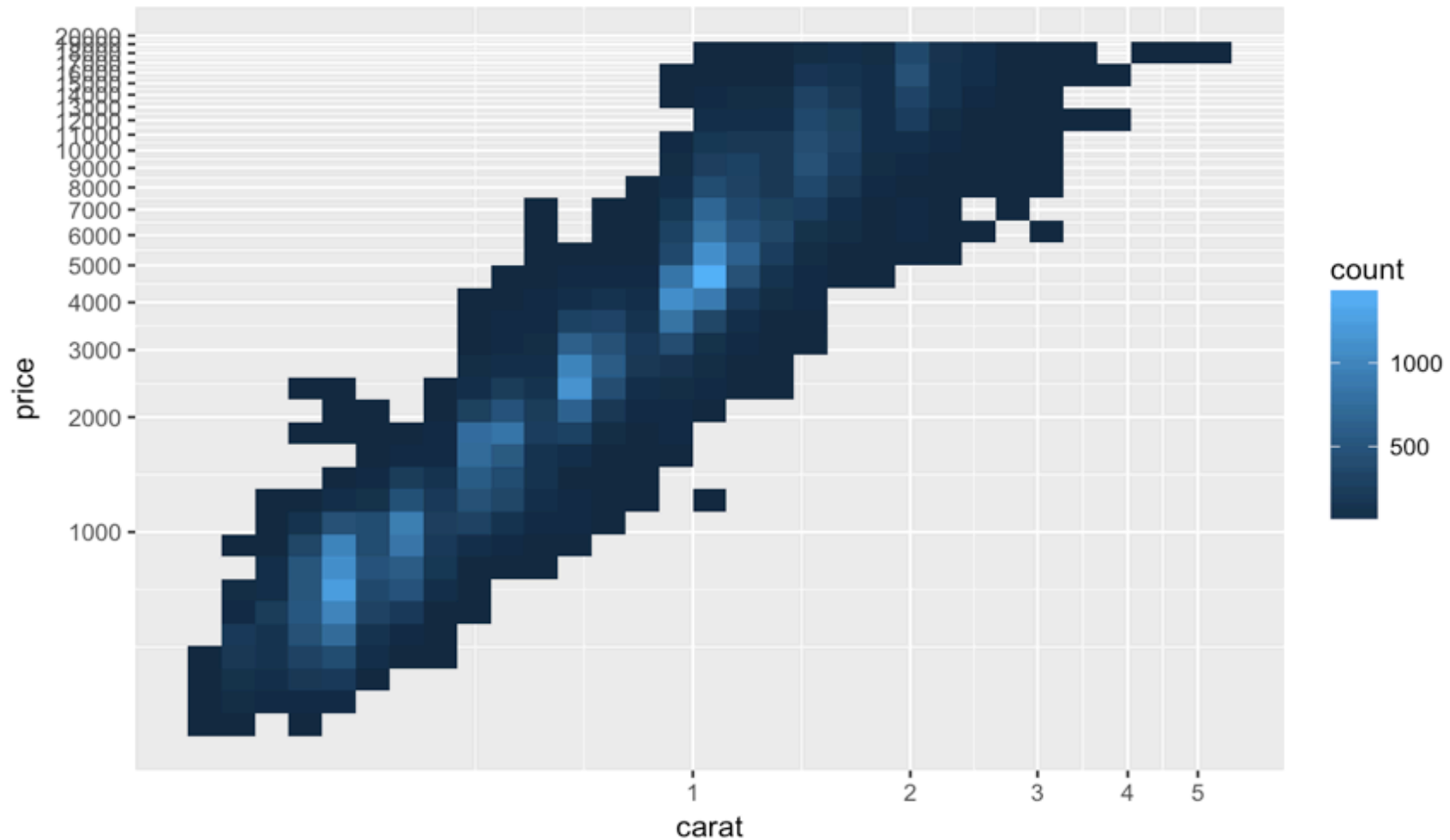


```
ggplot(diamonds, aes(carat, price)) +  
  geom_bin2d() +  
  scale_x_log10() +  
  scale_y_log10()
```

changing the scales instead of the
variables can help with
interpretation

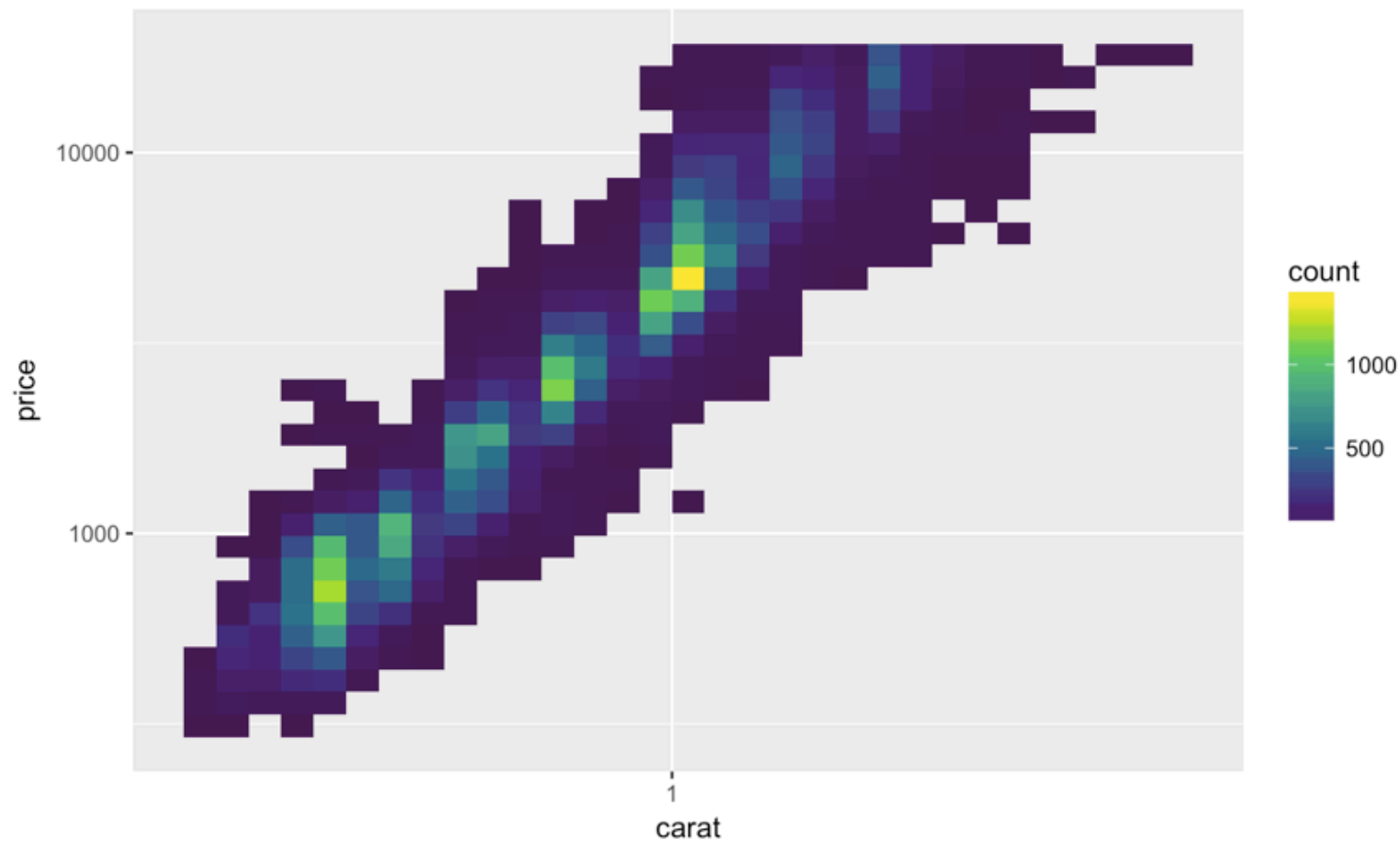


```
ggplot(diamonds, aes(carat, price)) +  
  geom_bin2d() +  
  scale_x_log10(breaks = seq(0, 5, by = 1)) +  
  scale_y_log10(breaks = seq(1000, 20000, by = 1000))
```



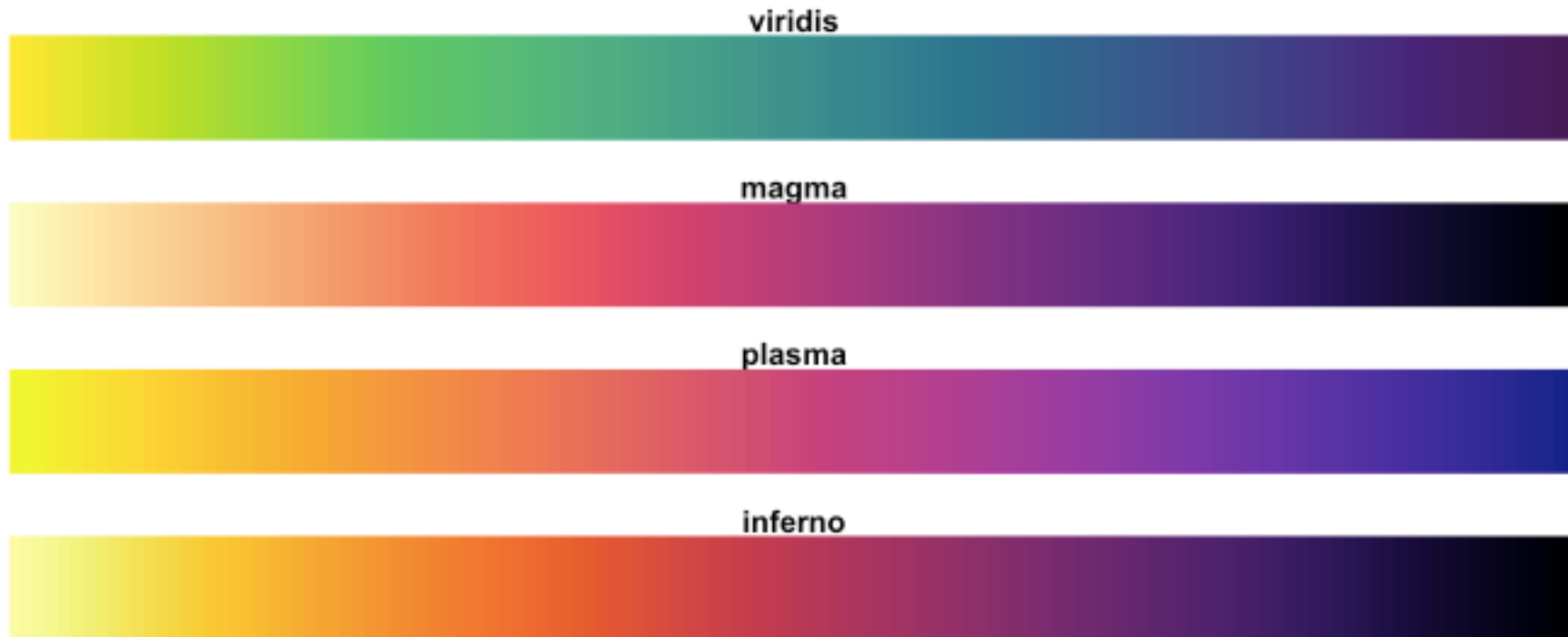
```
ggplot(diamonds, aes(carat, price)) +  
  geom_bin2d() +  
  scale_x_log10() +  
  scale_y_log10() +  
  scale_fill_viridis_c()
```

scale_fill_viridis_c() provides
carefully chosen color scales for
continuous variables.



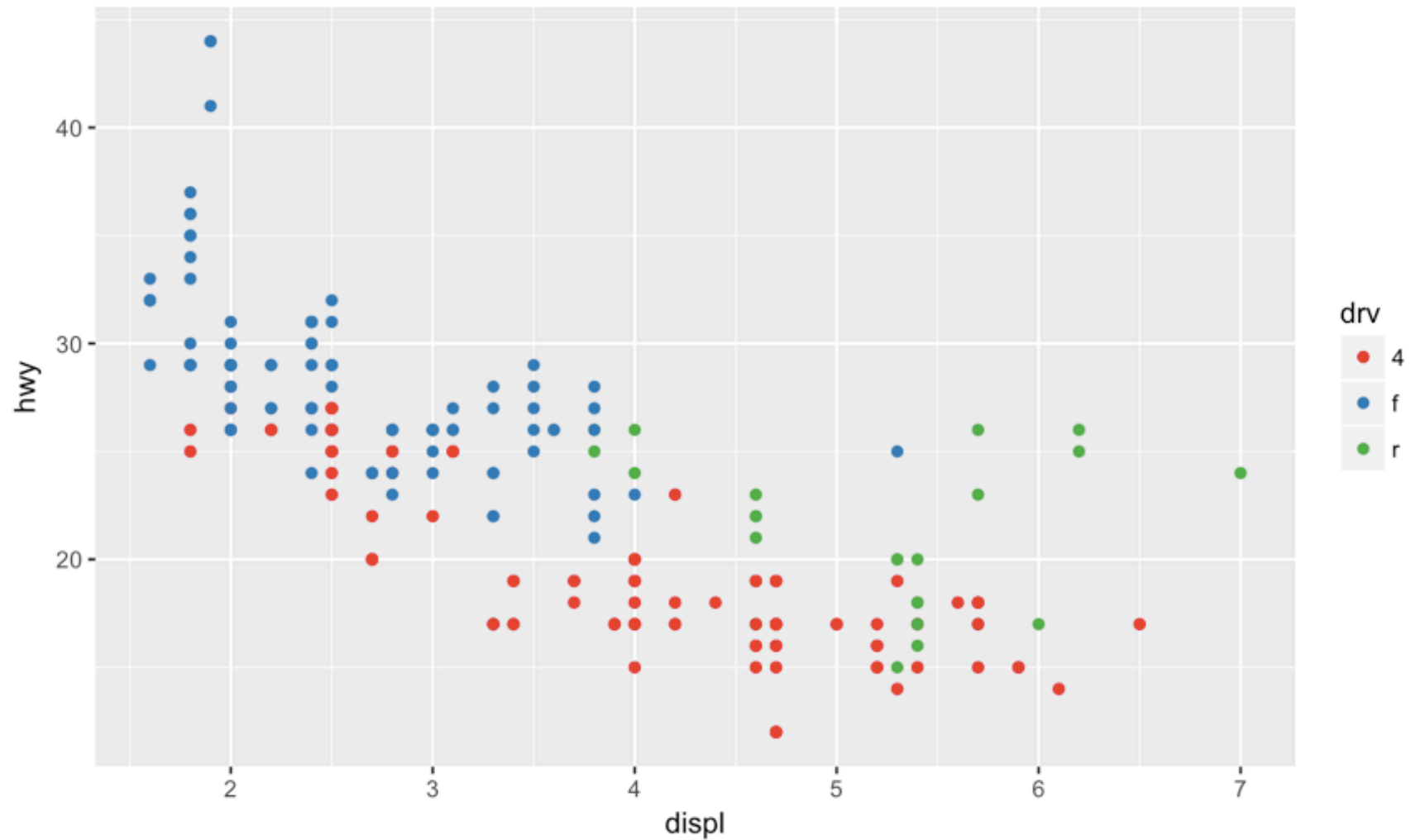
Viridis color scales

```
scale_fill_viridis(option = "magma")
```



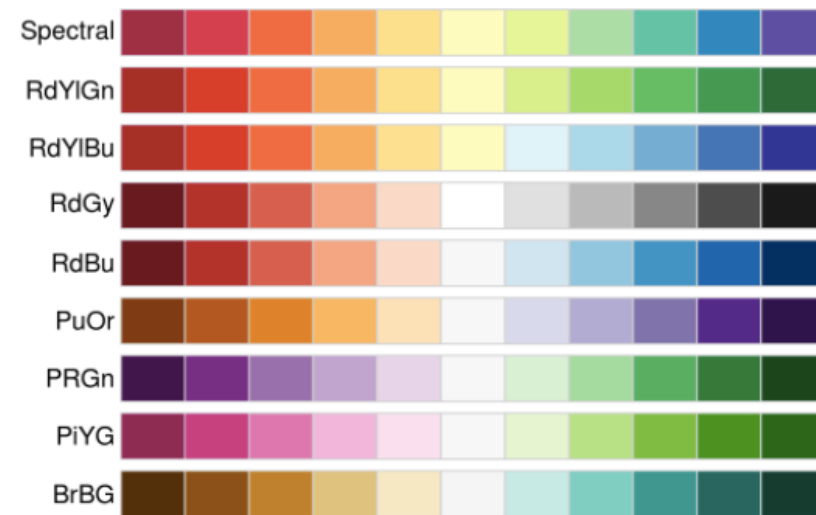
```
ggplot(mpg, aes(displ, hwy)) +  
  geom_point(aes(color = drv)) +  
  scale_color_brewer(palette = "Set1")
```

`scale_fill_brewer()` provides
carefully chosen color palettes for
discrete variables.



Brewer color palettes

```
scale_color_brewer(palette = "Set1")
```




```
presidential %>%
```

```
  mutate(id = row_number()) %>%
```

```
  ggplot(aes(start, id, color = party)) +
```

```
  geom_point() +
```

```
  geom_segment(aes(xend = end, yend = id)) +
```

```
  scale_color_manual(values = c(Republican = "red", Democratic = "blue"))
```

