Chapter 1_R4DS

Michael Rose

A data frame with 234 rows and 11 variables

manufacturer model model name

displ engine displacement, in litres

year year of manufacture

cyl number of cylinders

trans type of transmission

drv f = front-wheel drive, r = rear wheel drive, 4 = 4wd

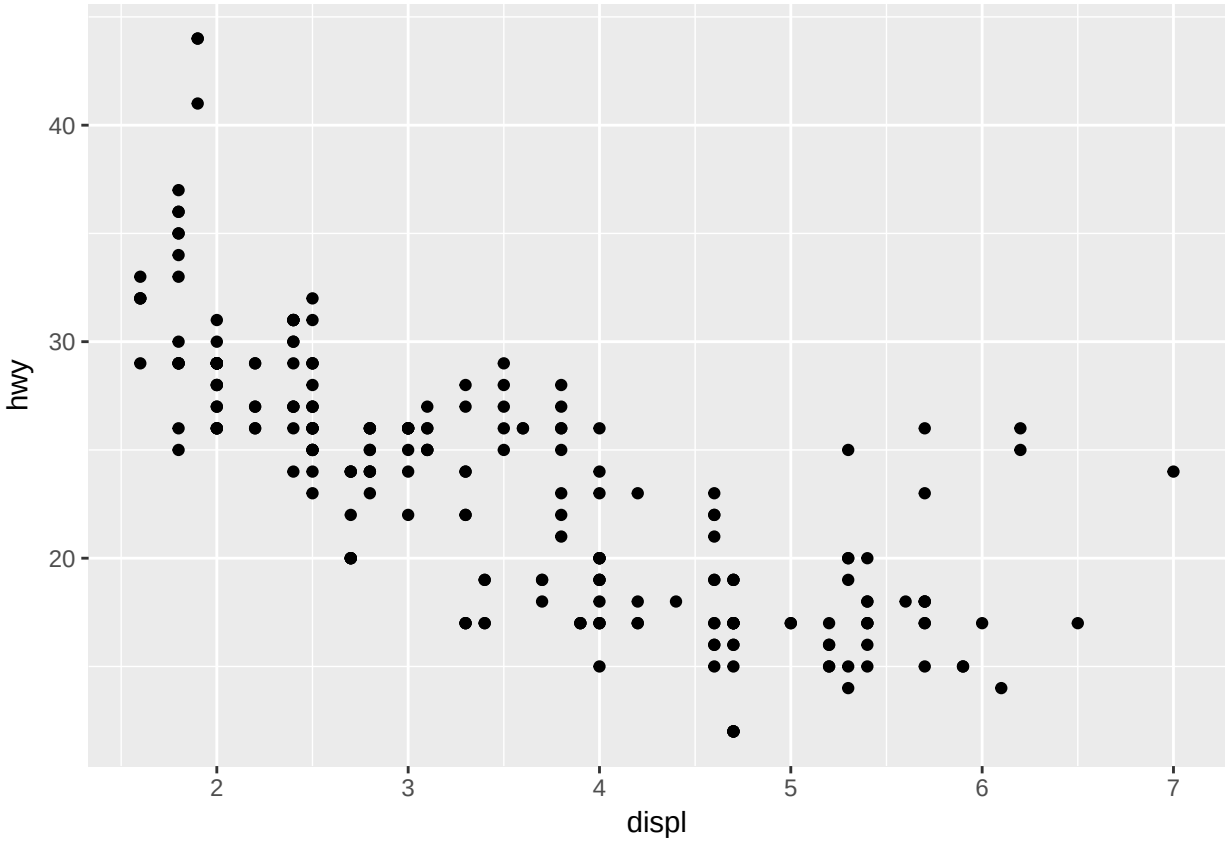
cty city miles per gallon

hwy highway miles per gallon

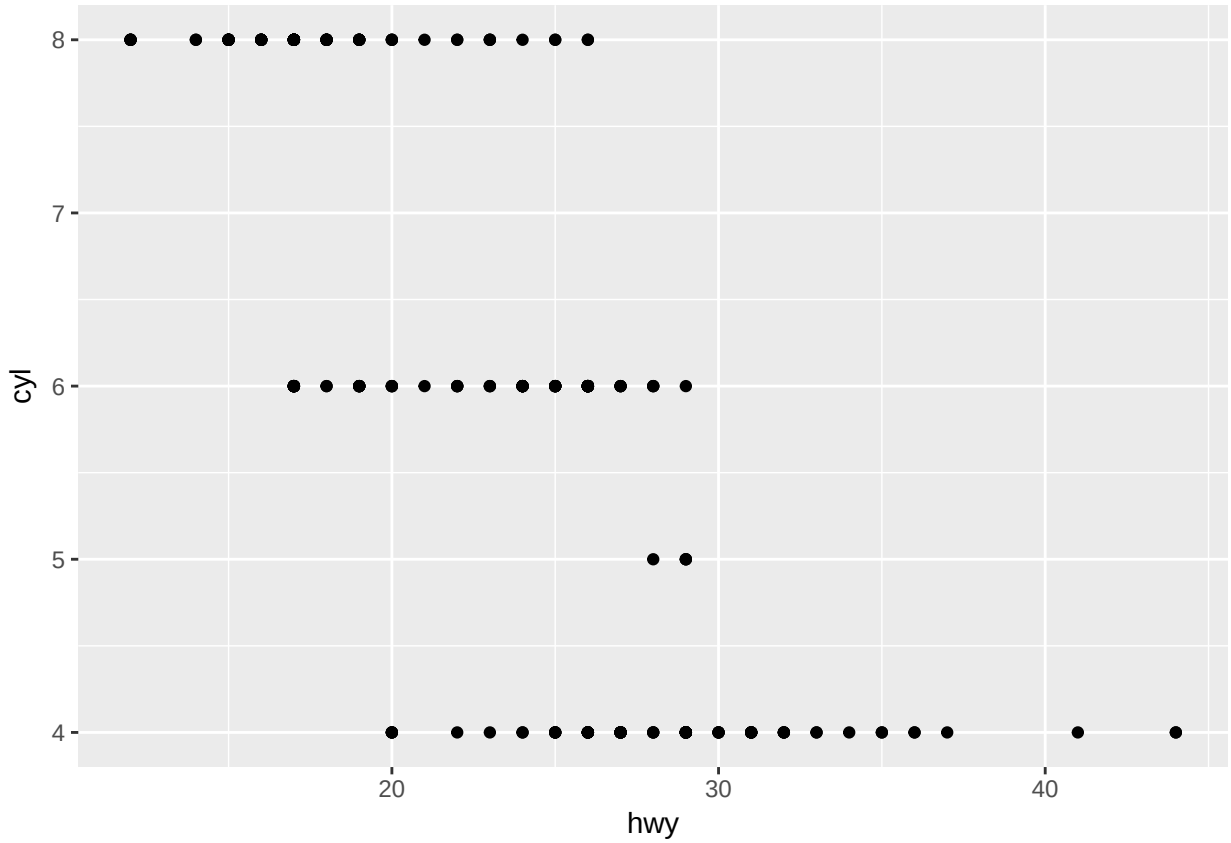
fl fuel type

class "type" of car

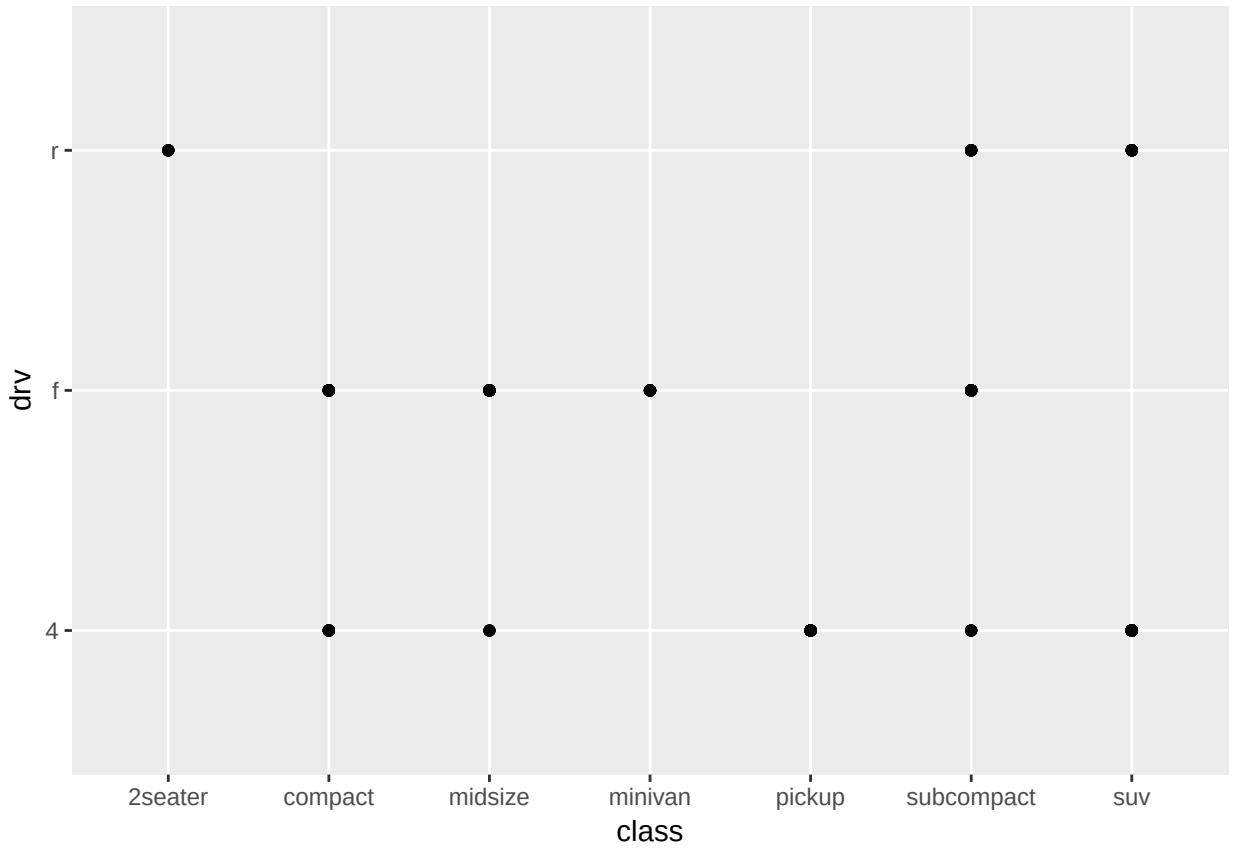
```
# load mpg data set from ggplot2. mpg is a dataframe
mpg <- ggplot2::mpg

#plot displ (litres) vs. highway mpg
ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy))
```



```
# plot highway mpg vs cylinders
ggplot(data = mpg) + geom_point(mapping = aes(x = hwy, y = cyl))
```

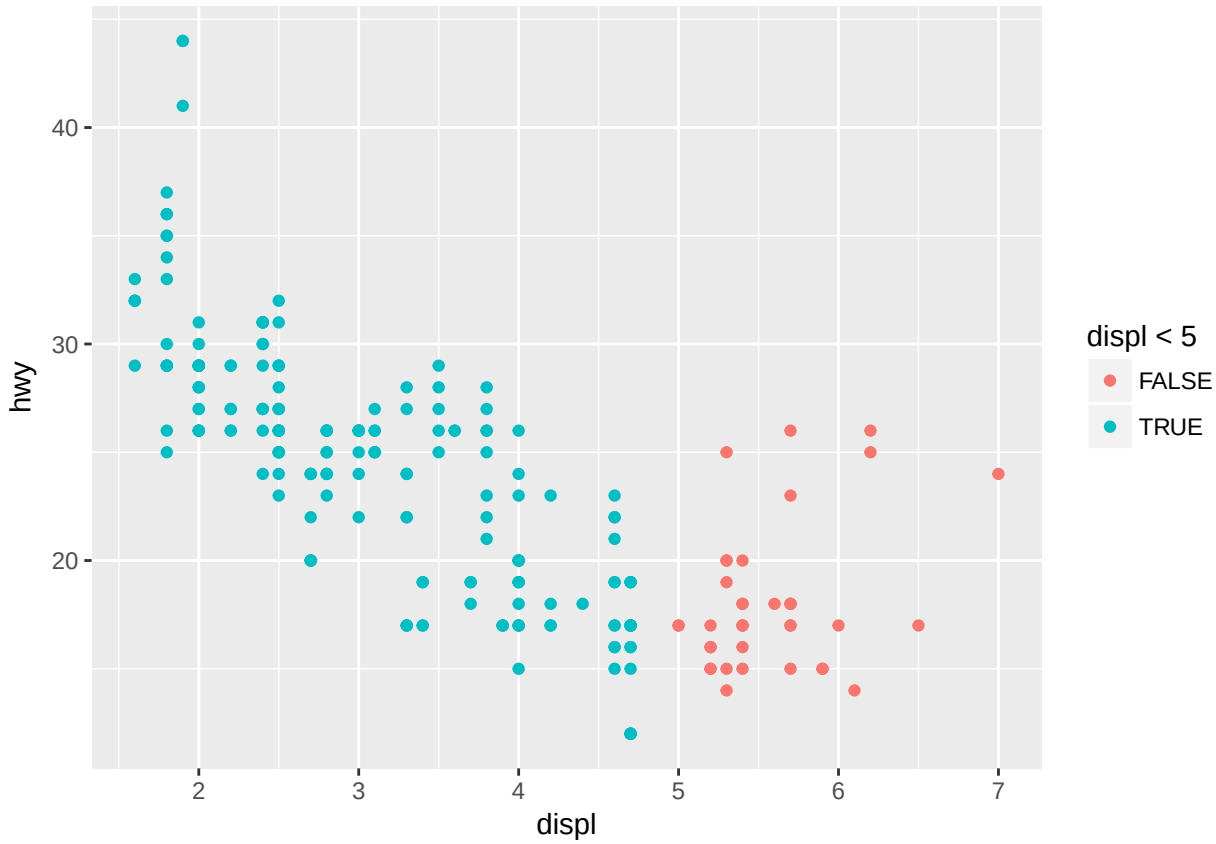



```
# plot unhelpful graph with categorical variables
ggplot(data = mpg) + geom_point(mapping = aes(class, drv))
```

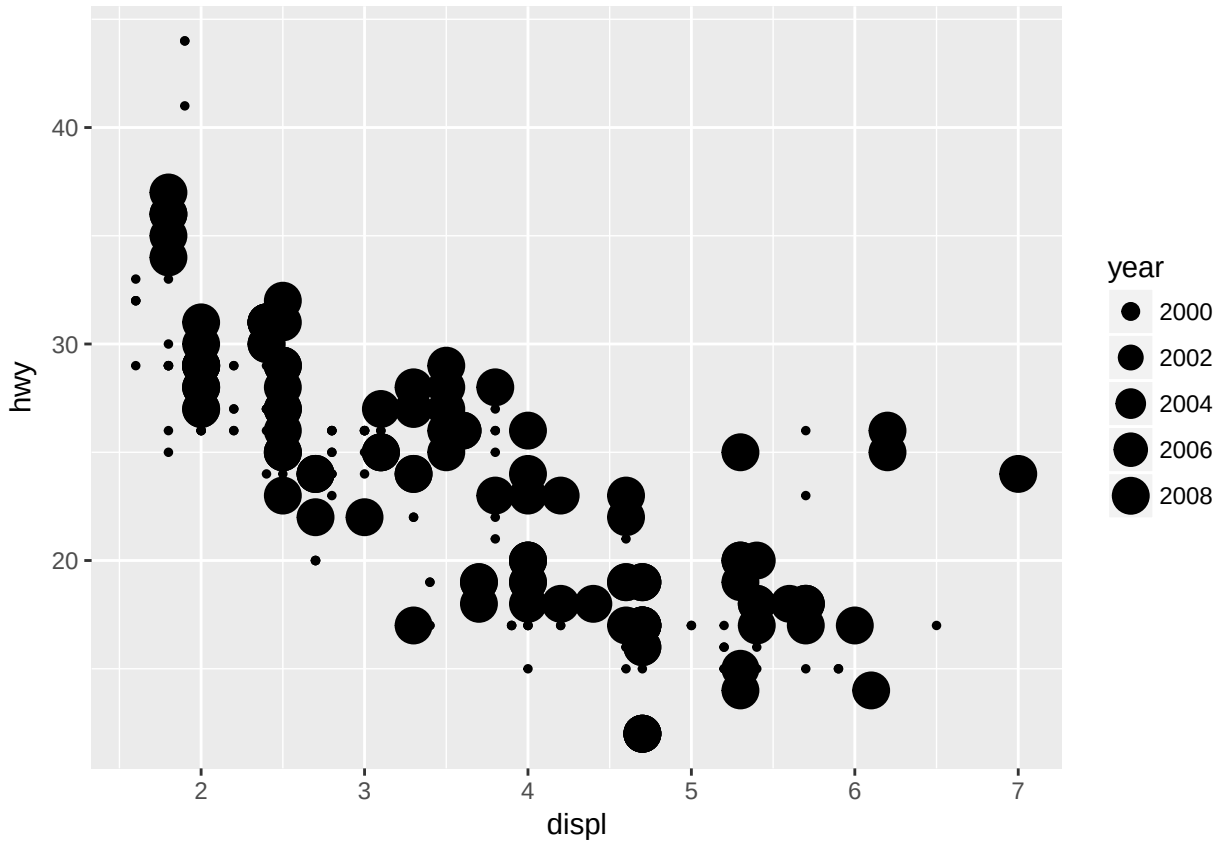


Aesthetic Mappings

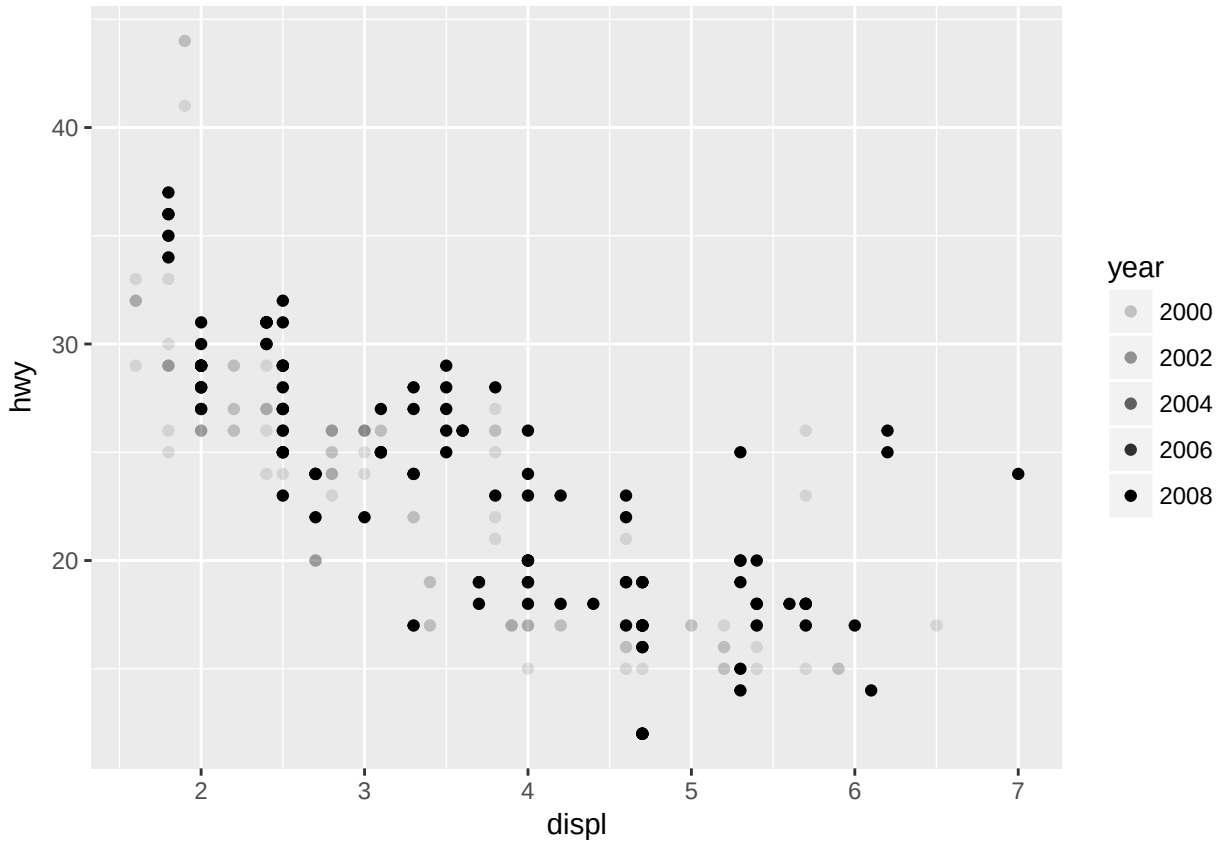
```
# plot with color aesthetic mapped to car class
ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy, color = displ<5))
```




```
# plot with size aesthetic mapped to class
ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy, size = year))
```




```
# plot with alpha aesthetic mapped to class
ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy, alpha = year))
```

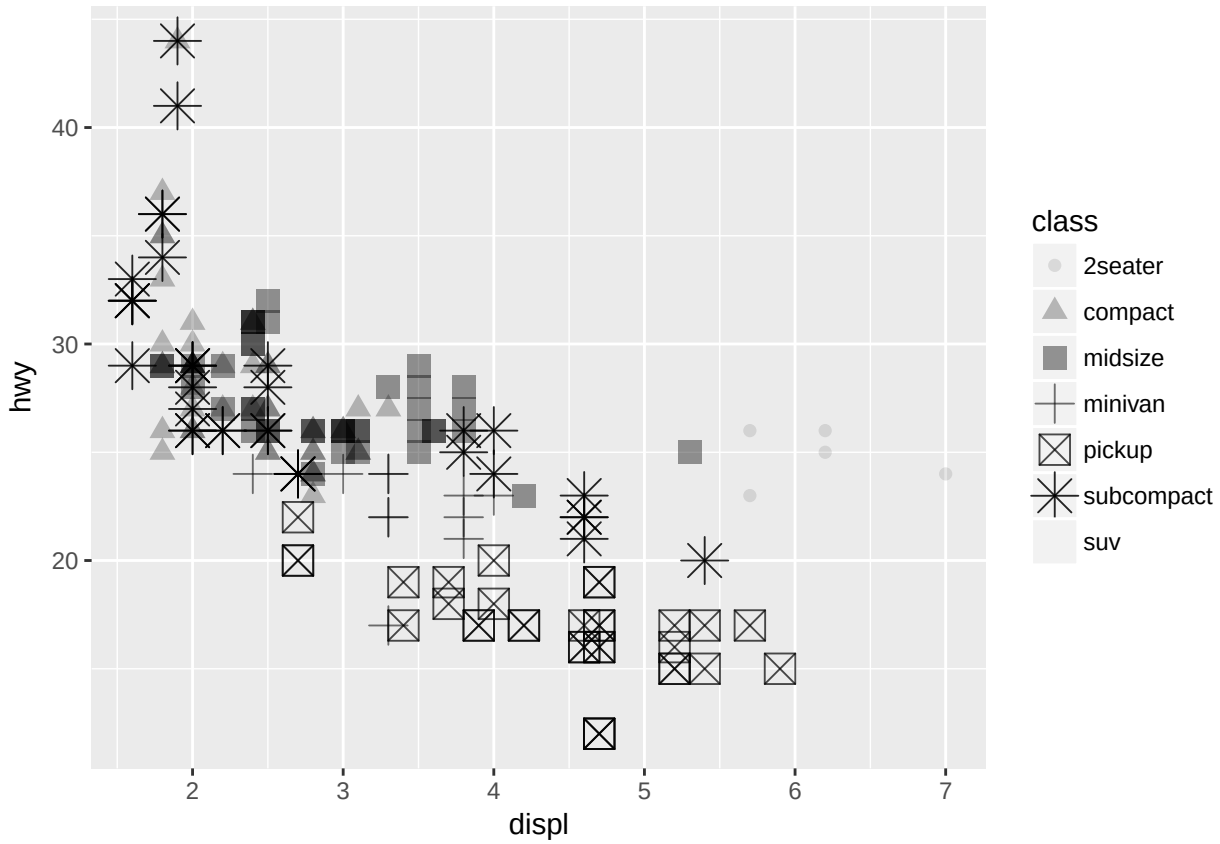



```
# plot with shape aesthetic mapped to class
ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy, shape = class, size = class, alpha = c
```

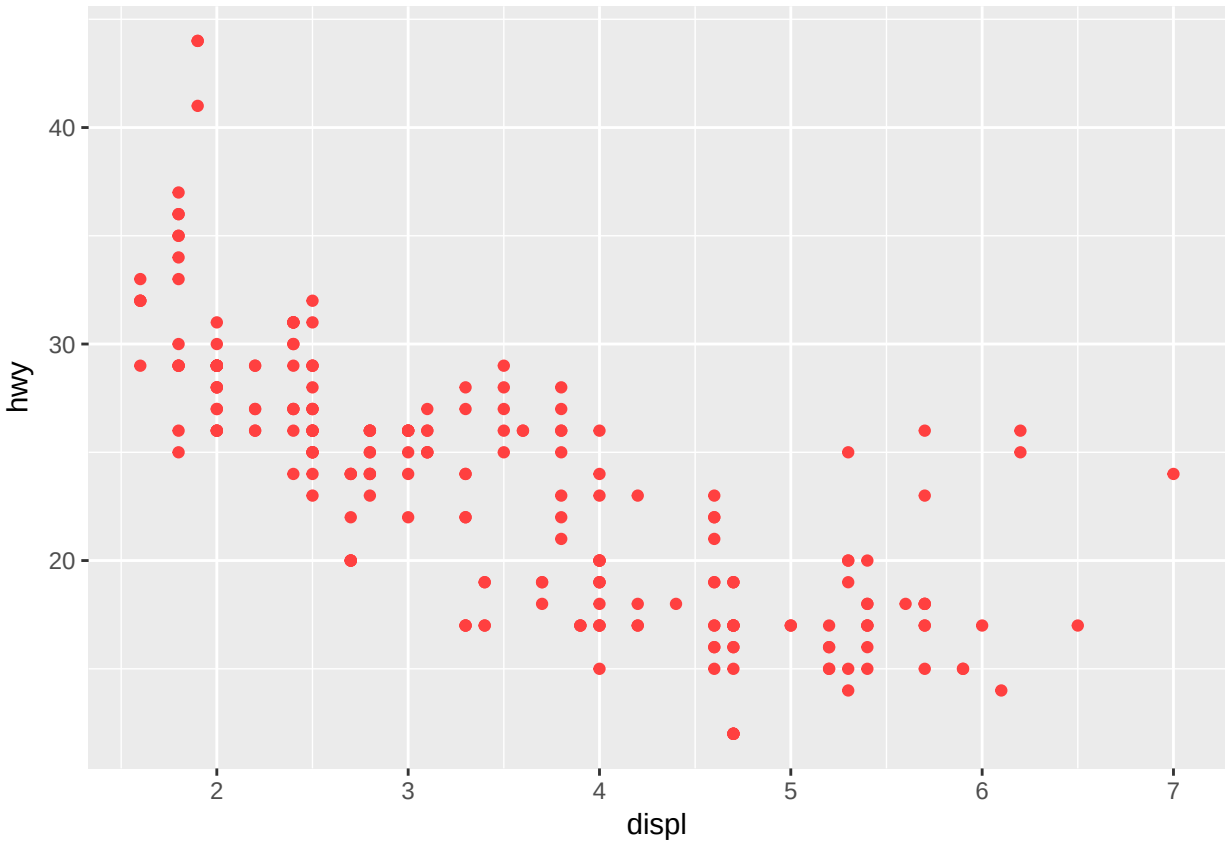
```
## Warning: Using size for a discrete variable is not advised.
```

```
## Warning: The shape palette can deal with a maximum of 6 discrete values
## because more than 6 becomes difficult to discriminate; you have 7.
## Consider specifying shapes manually if you must have them.
```

```
## Warning: Removed 62 rows containing missing values (geom_point).
```




```
# plot and make all data points blue
ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy), color = "brown1")
```

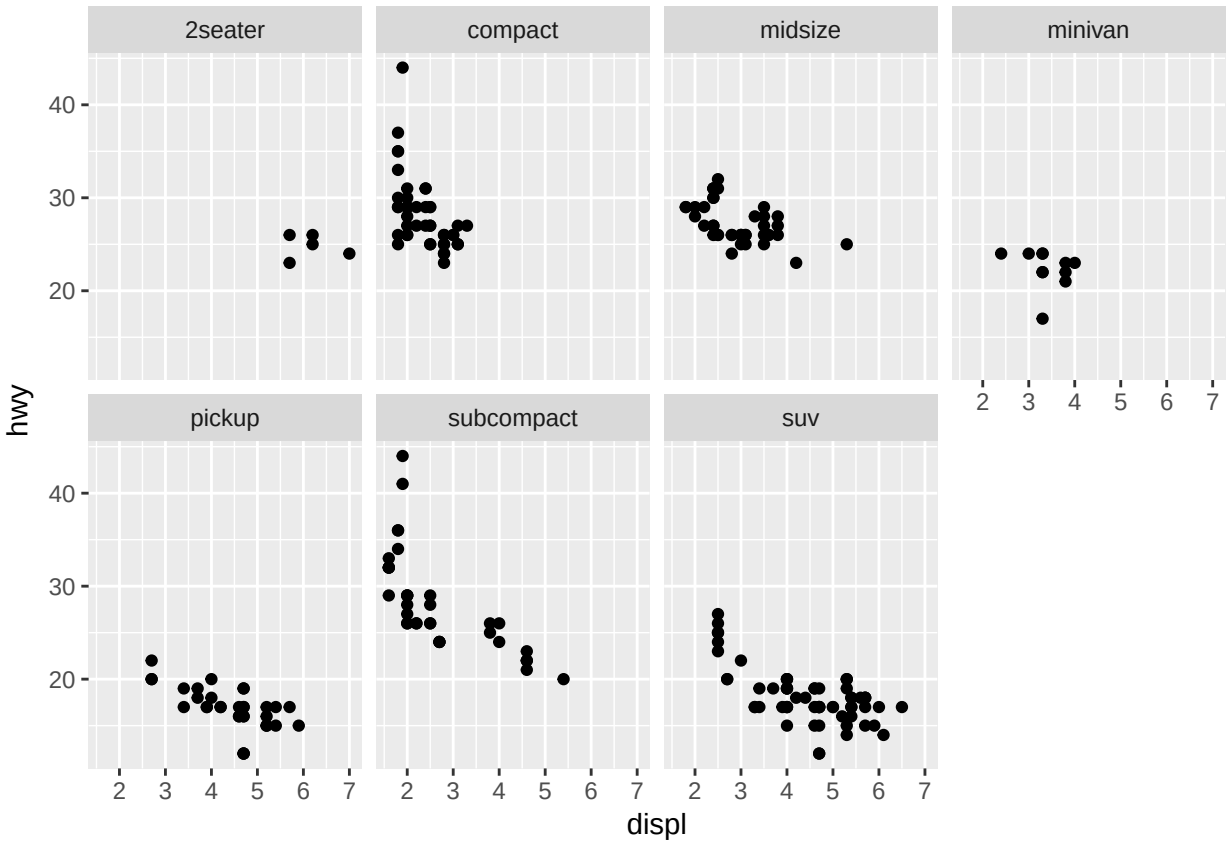



```
# shows which variables are continuous and which are categorical
glimpse(mpg)
```

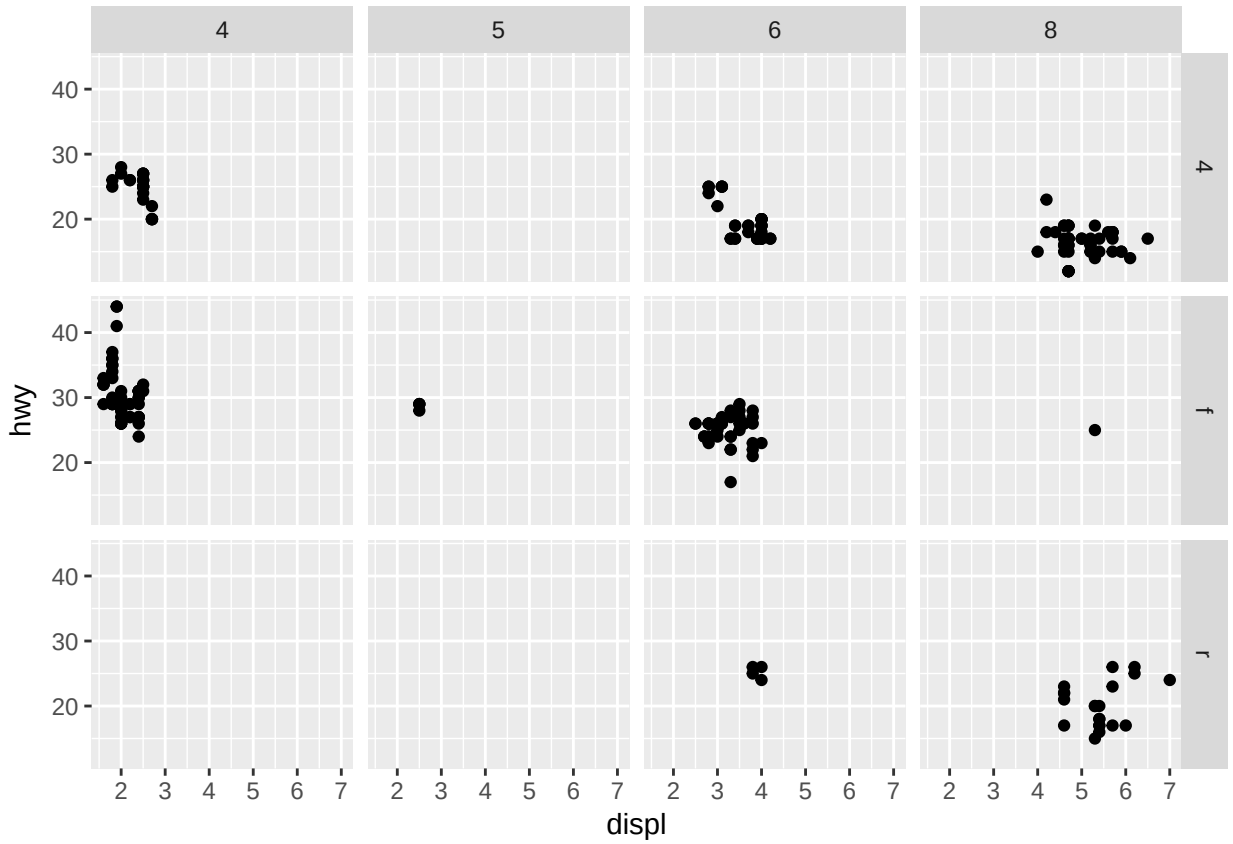
```
## Observations: 234
## Variables: 11
## $ manufacturer <chr> "audi", "audi", "audi", "audi", "audi", "audi", "...
## $ model        <chr> "a4", "a4", "a4", "a4", "a4", "a4", "a4", "a4 qua...
## $ displ        <dbl> 1.8, 1.8, 2.0, 2.0, 2.8, 2.8, 3.1, 1.8, 1.8, 2.0,...
## $ year         <int> 1999, 1999, 2008, 2008, 1999, 1999, 2008, 1999, 1...
## $ cyl          <int> 4, 4, 4, 4, 6, 6, 6, 4, 4, 4, 4, 6, 6, 6, 6, 6, 6...
## $ trans        <chr> "auto(l5)", "manual(m5)", "manual(m6)", "auto(av)...
## $ drv          <chr> "f", "f", "f", "f", "f", "f", "f", "4", "4", "4",...
## $ cty          <int> 18, 21, 20, 21, 16, 18, 18, 18, 16, 20, 19, 15, 1...
## $ hwy          <int> 29, 29, 31, 30, 26, 26, 27, 26, 25, 28, 27, 25, 2...
## $ fl           <chr> "p", "p", "p", "p", "p", "p", "p", "p", "p", "p",...
## $ class        <chr> "compact", "compact", "compact", "compact", "comp...
```

Facets

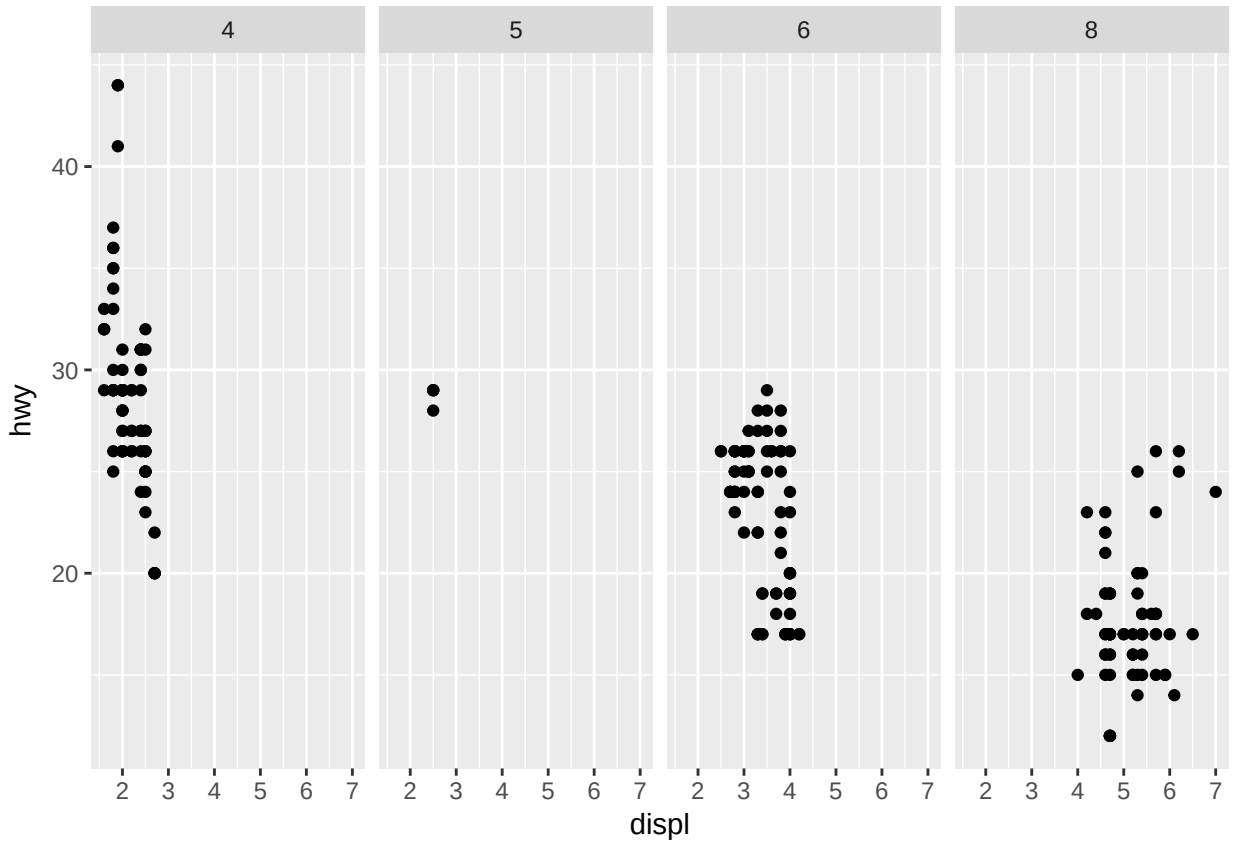
```
# facet wrap by class
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
    facet_wrap(~ class, nrow = 2)
```



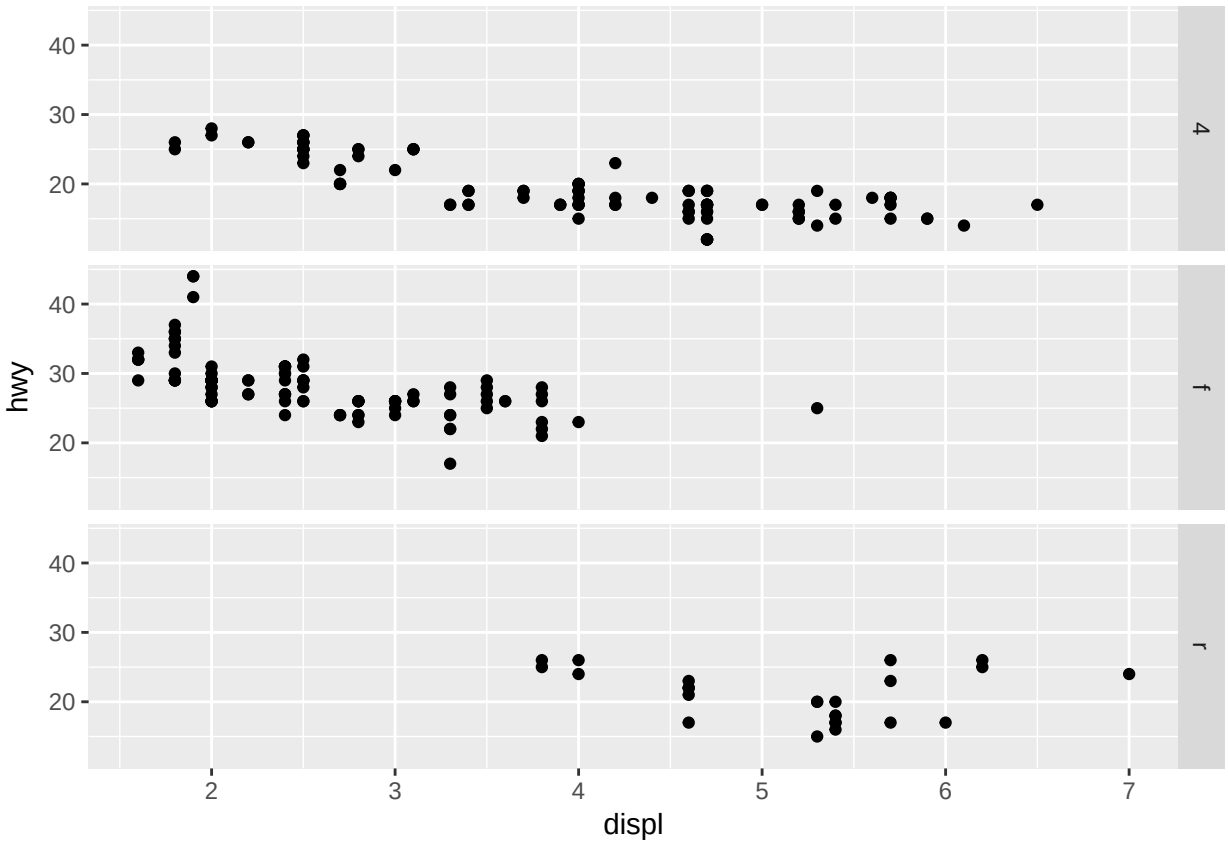

```
# facet grid drive type and cylinders
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
    facet_grid(drv~cyl)
```




```
# facet grid cylinders
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
    facet_grid(.~cyl)
```

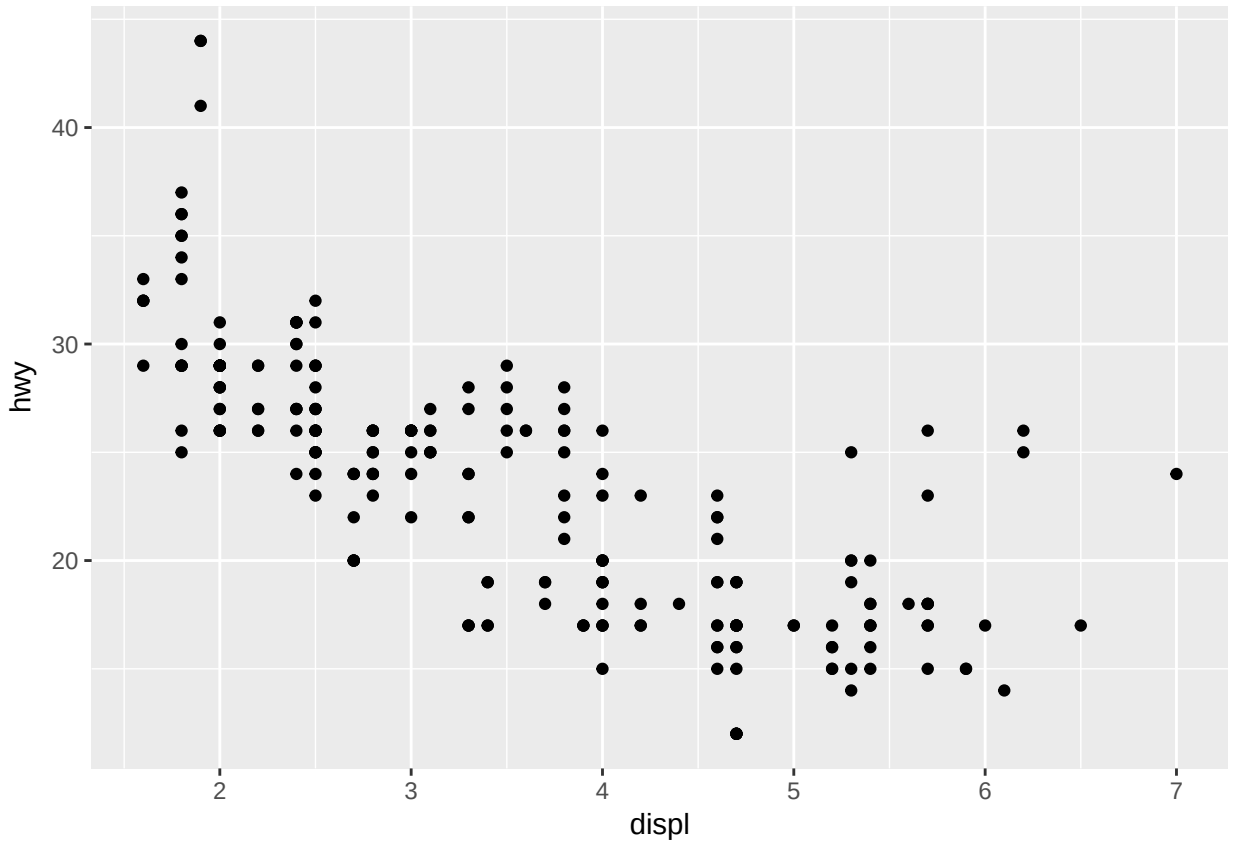



```
# same as above but horizontal orientation
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
    facet_grid(drv ~ .)
```



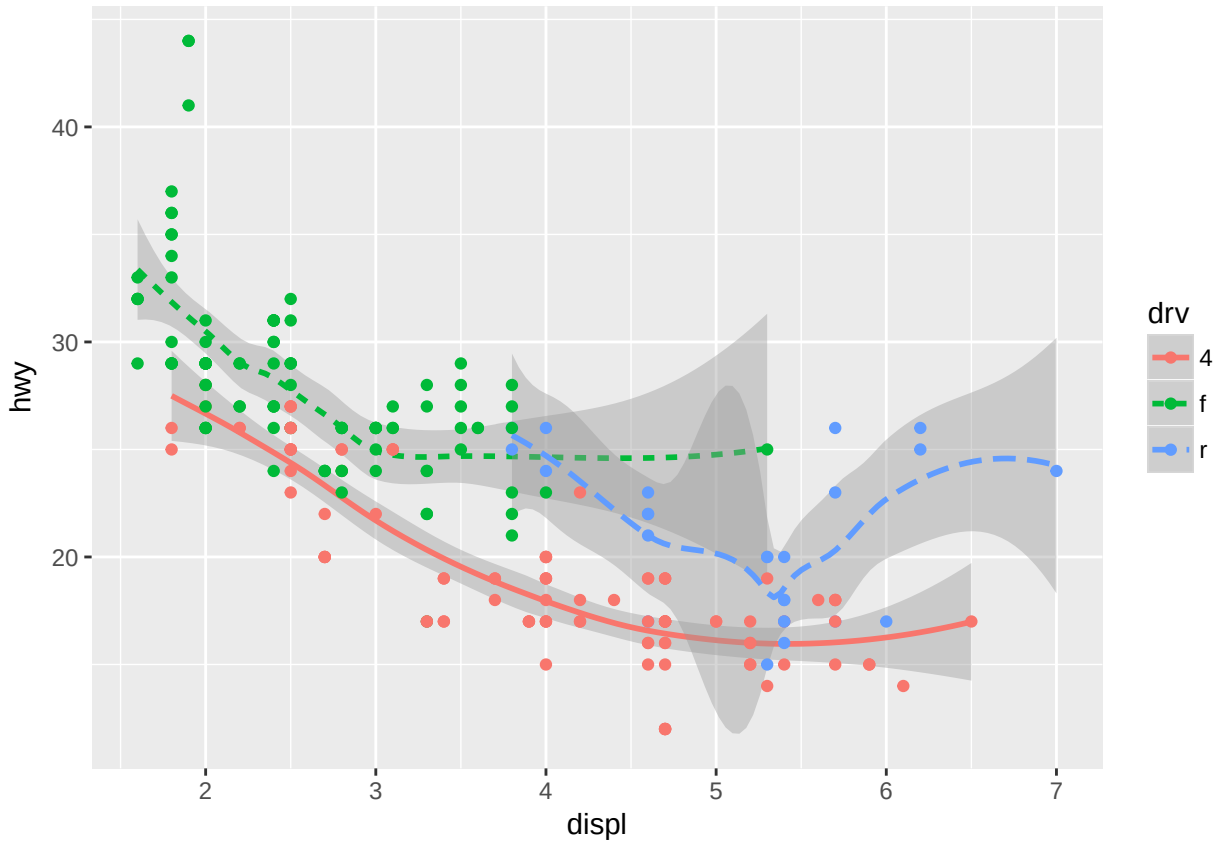
Geometric Objects

```
# left
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy))
```



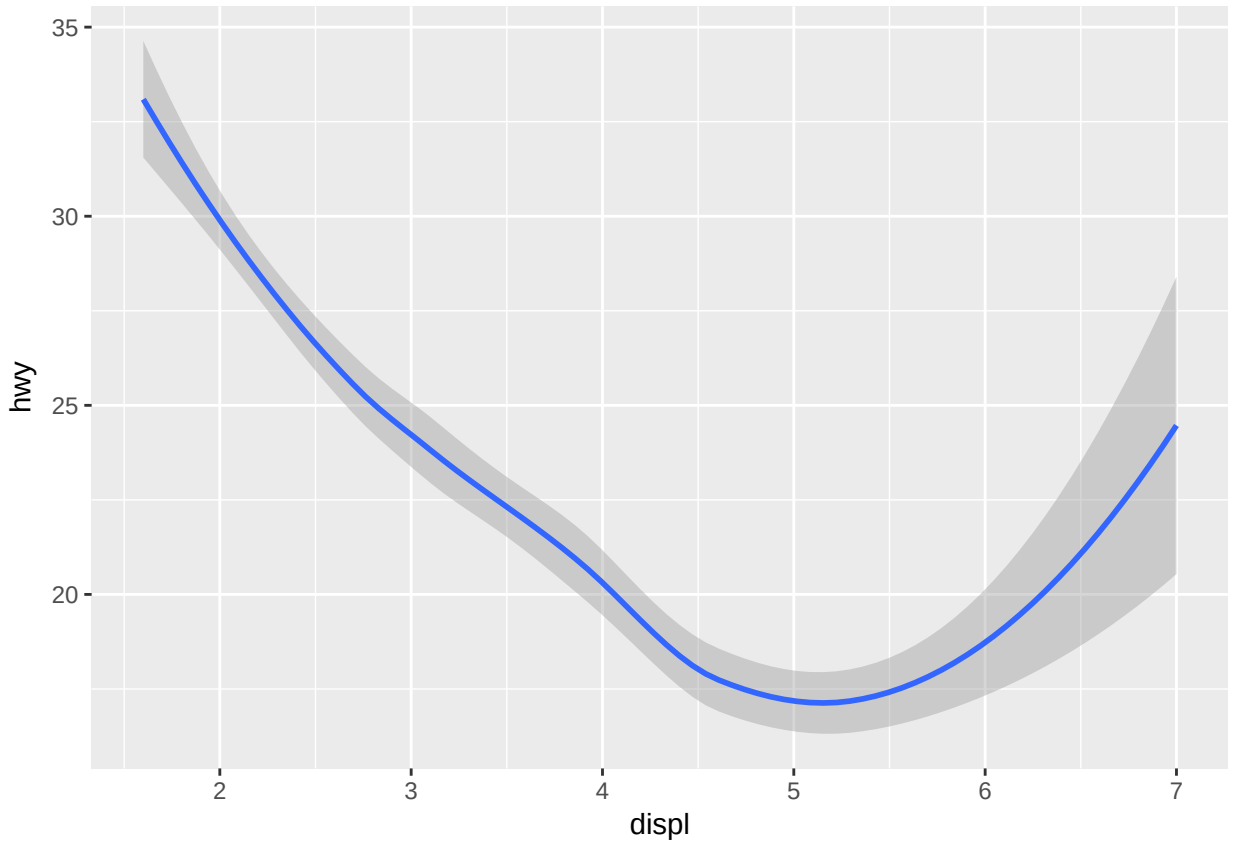

```
# right
ggplot(data = mpg) +
  geom_smooth(mapping = aes(x = displ, y = hwy, linetype = drv, color = drv)) +
    geom_point(mapping = aes(x = displ, y = hwy, color = drv))
```

```
## `geom_smooth()` using method = 'loess'
```



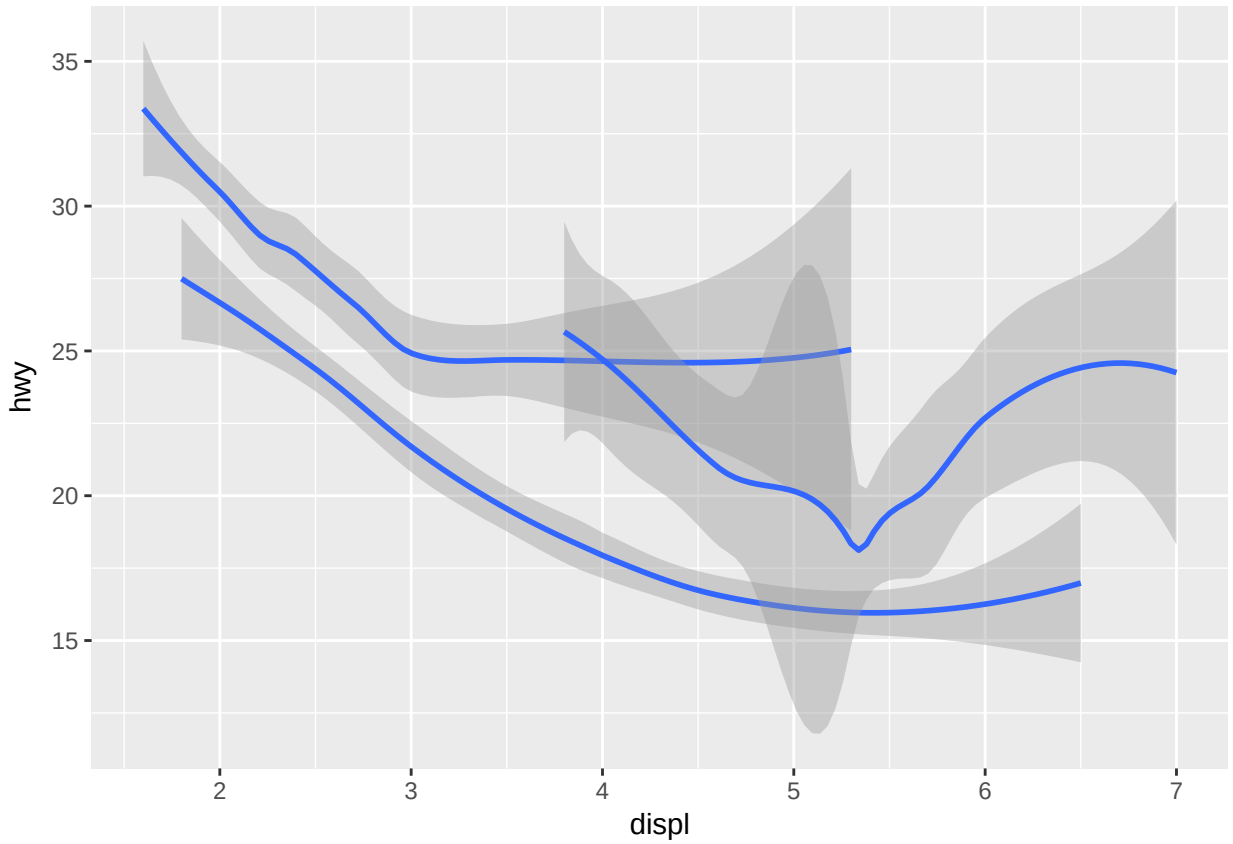

```
ggplot(data = mpg) +
  geom_smooth(mapping = aes(x = displ, y = hwy))
```

```
## `geom_smooth()` using method = 'loess'
```



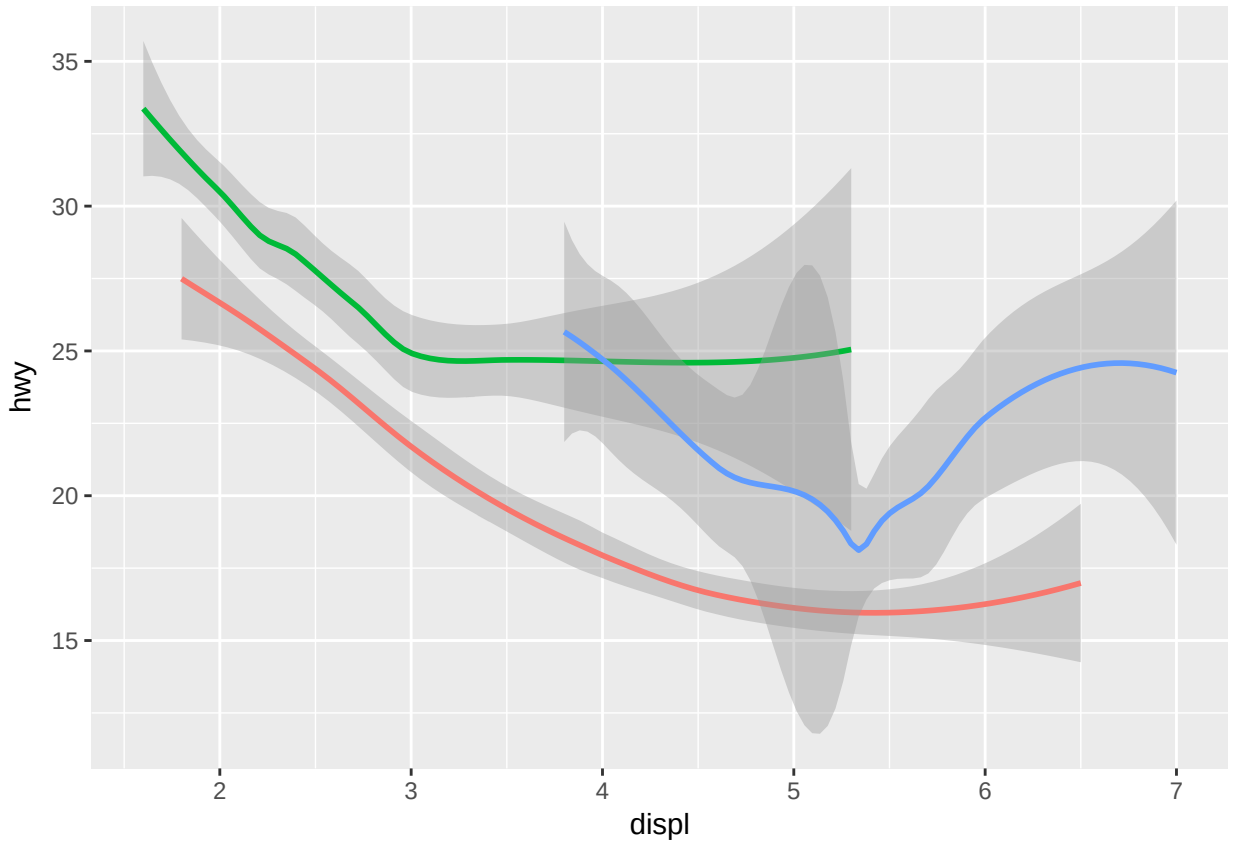

```
ggplot(data = mpg) +
  geom_smooth(mapping = aes(x = displ, y = hwy, group = drv))
```

```
## `geom_smooth()` using method = 'loess'
```



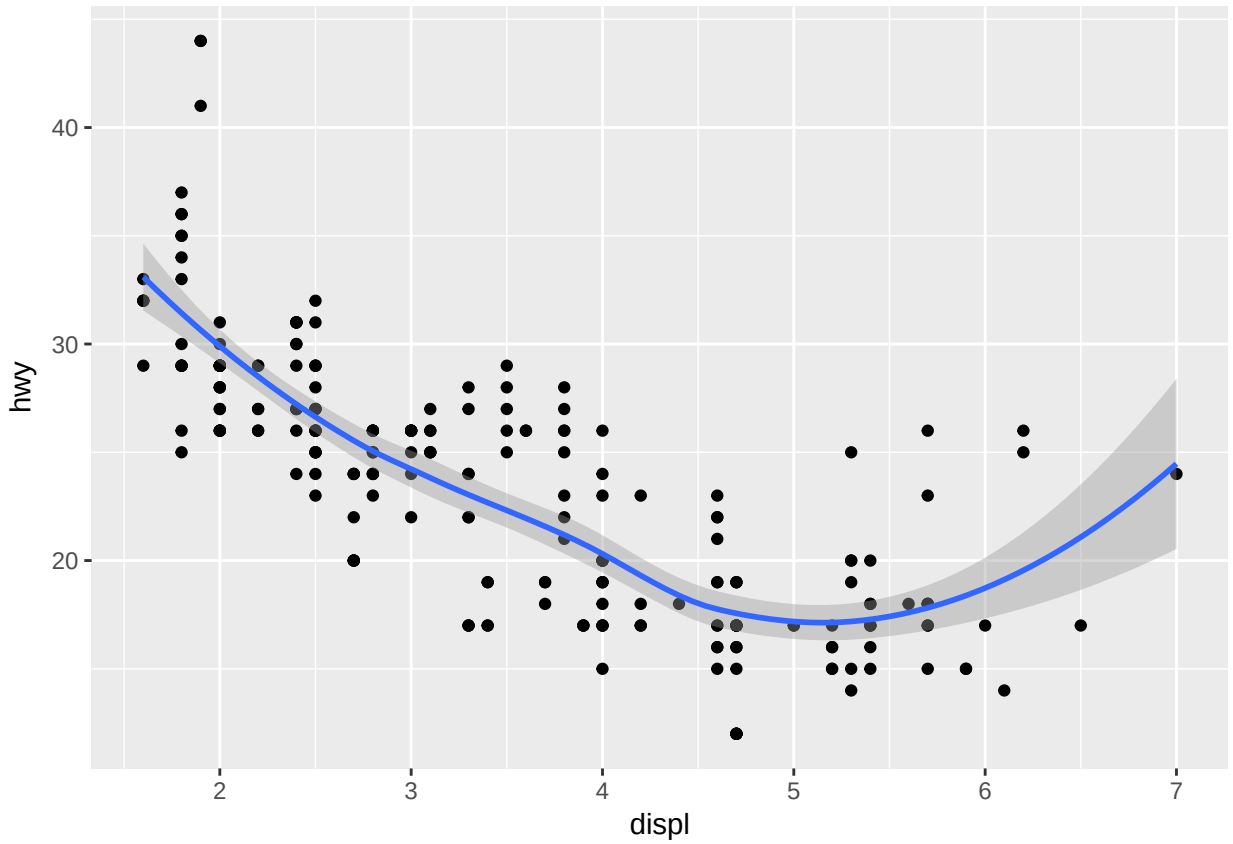

```
ggplot(data = mpg) +
  geom_smooth(mapping = aes(x = displ, y = hwy, color = drv), show.legend = FALSE)
```

```
## `geom_smooth()` using method = 'loess'
```



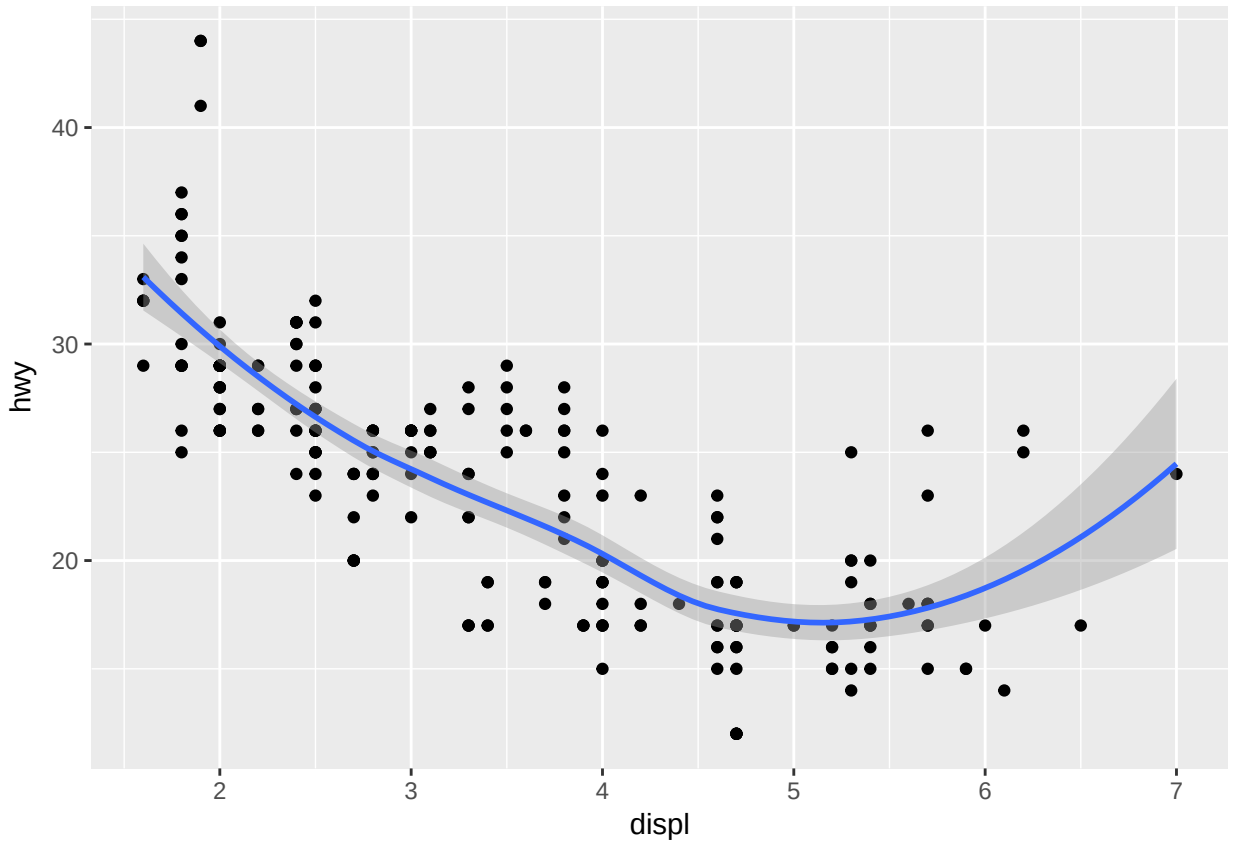

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
    geom_smooth(mapping = aes(x = displ, y = hwy))
```

```
## `geom_smooth()` using method = 'loess'
```



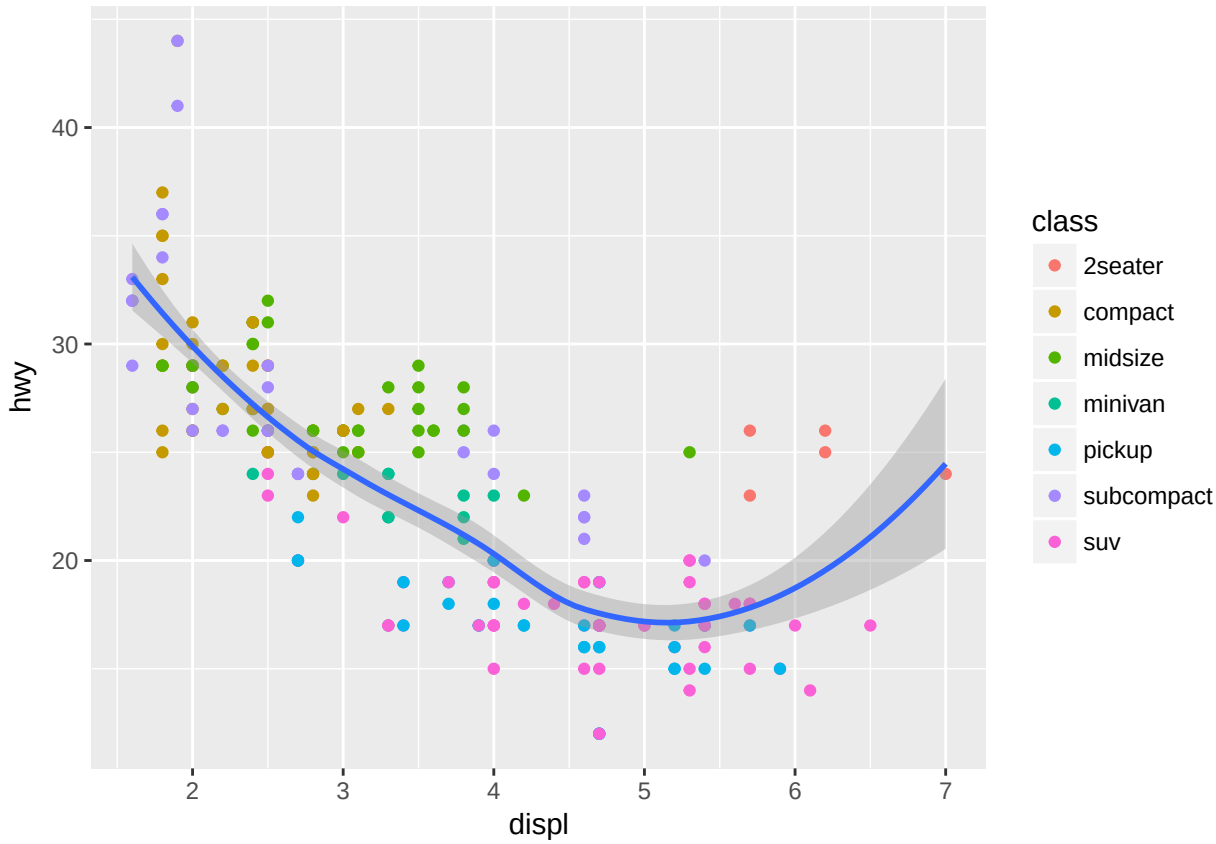

```
# cleaner
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) + geom_point() + geom_smooth()
```

```
## `geom_smooth()` using method = 'loess'
```



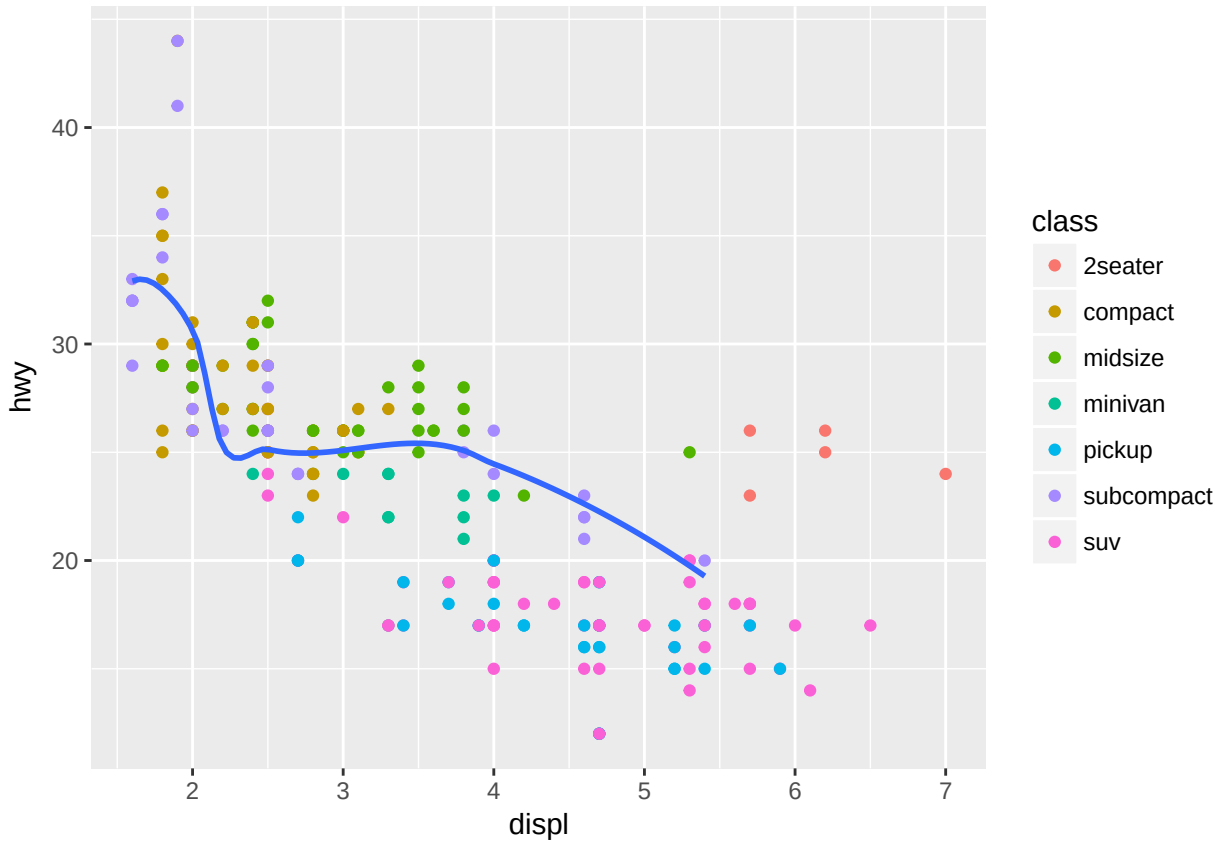

```
# over ride for 1 layer only
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) + geom_point(mapping = aes(color = class)) + geom_
```

```
## `geom_smooth()` using method = 'loess'
```



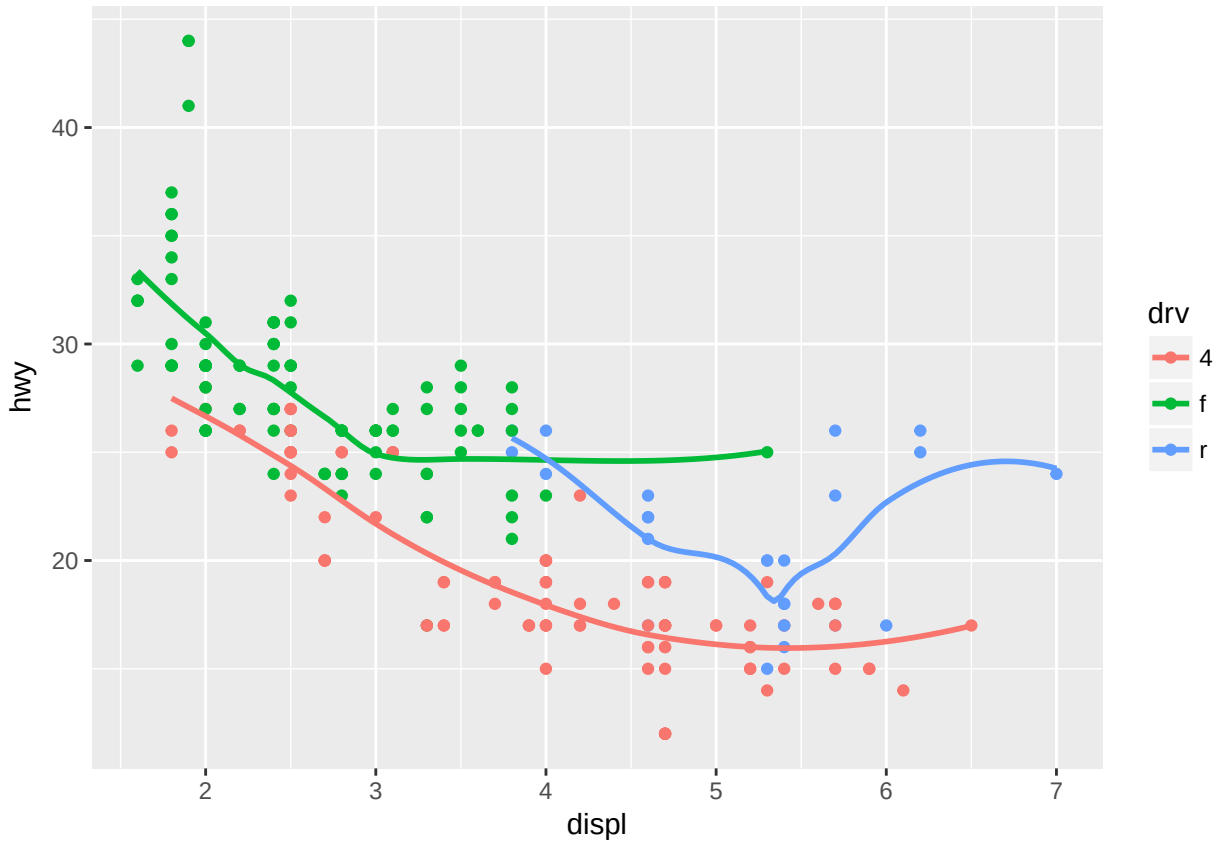

```
# geom smooth for subcompact only
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point(mapping = aes(color = class)) +
    geom_smooth(data = filter(mpg, class == "subcompact"), se = FALSE)
```

```
## `geom_smooth()` using method = 'loess'
```



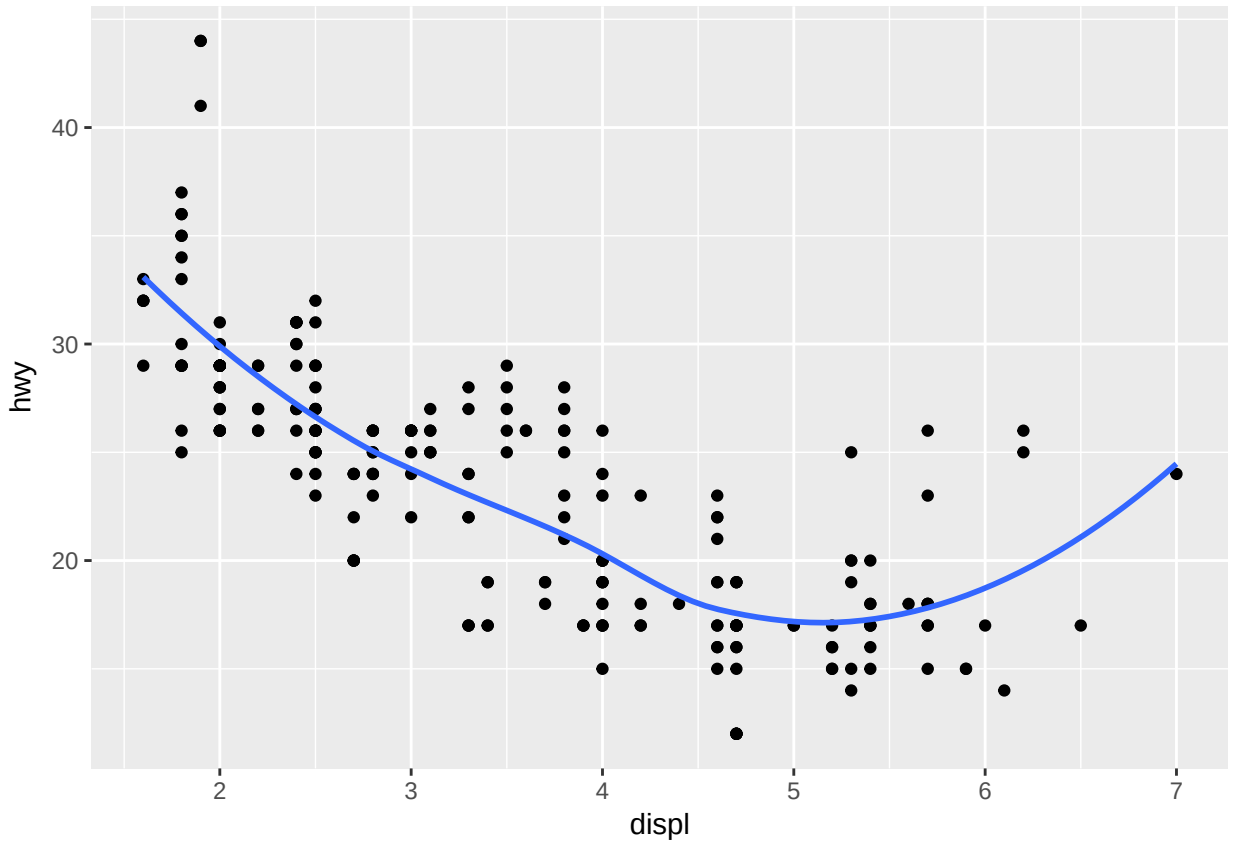

```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy, color = drv)) +
  geom_point() + geom_smooth(se = FALSE)
```

```
## `geom_smooth()` using method = 'loess'
```



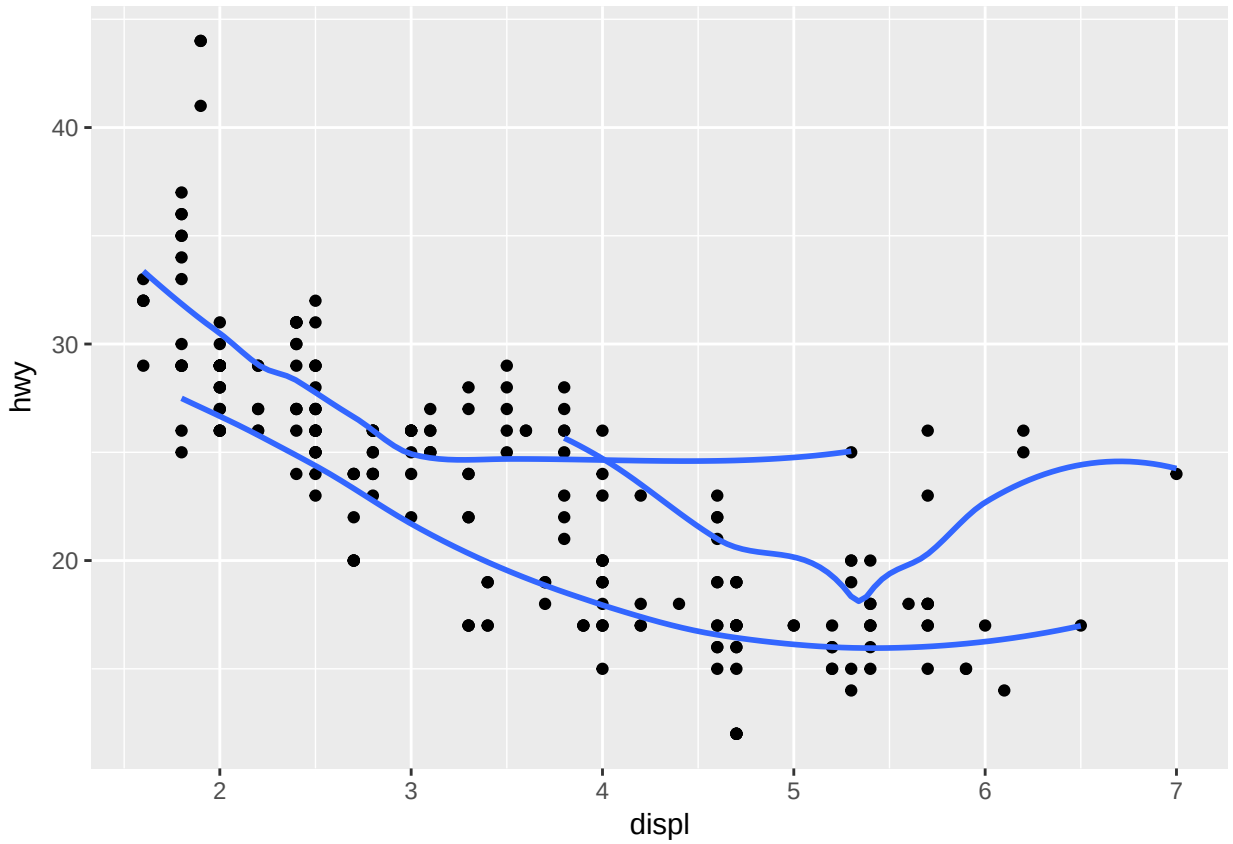

```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) + geom_point() + geom_smooth(se = FALSE)
```

```
## `geom_smooth()` using method = 'loess'
```



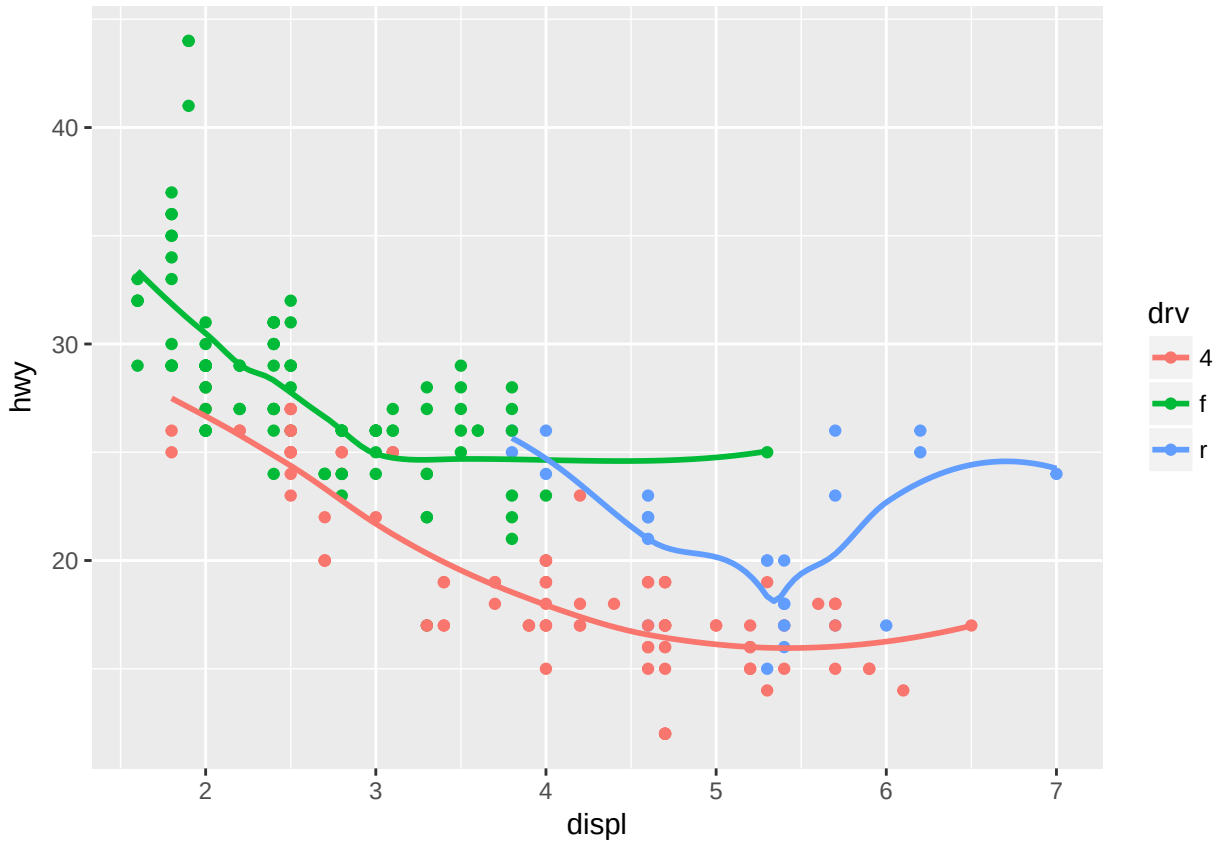

```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy, class = drv)) + geom_point() + geom_smooth(se = FA
```

```
## `geom_smooth()` using method = 'loess'
```



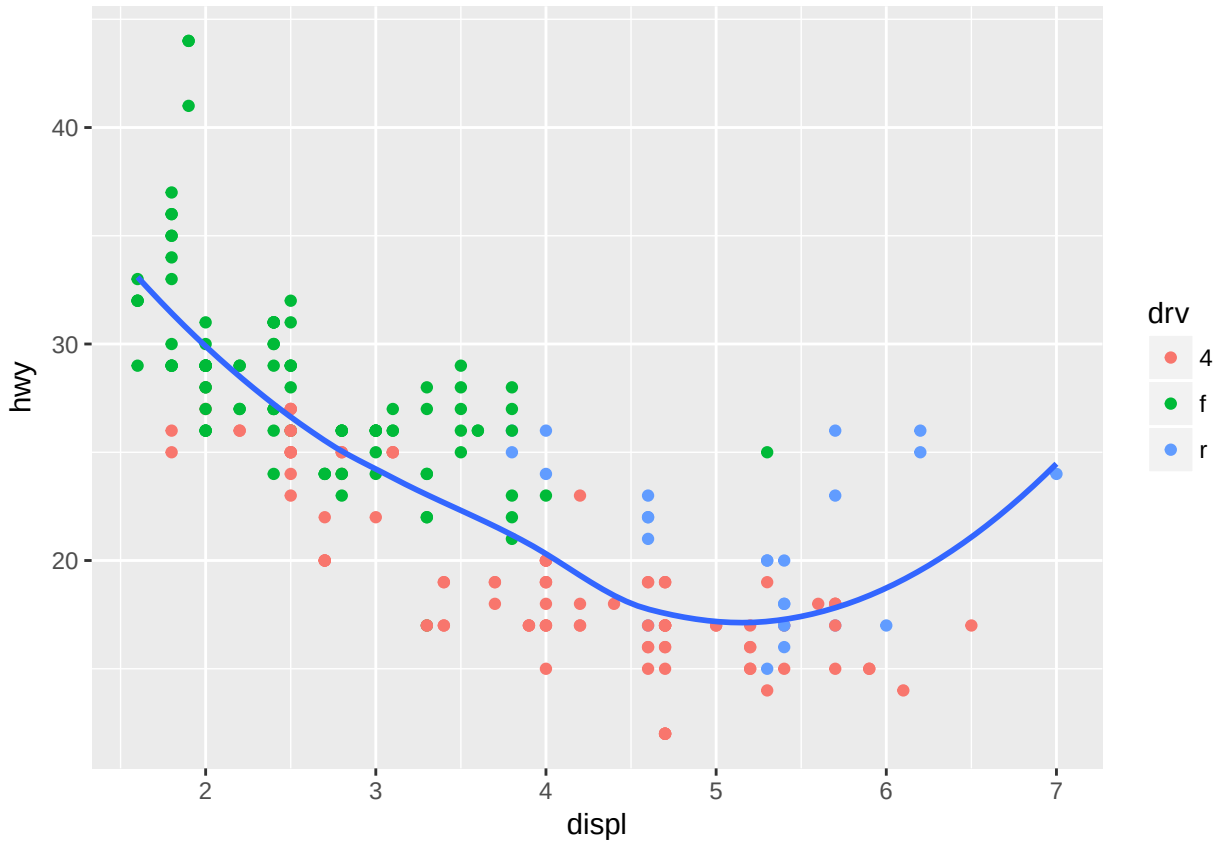

```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy, col = drv)) + geom_point() + geom_smooth(se = FALSE
```

```
## `geom_smooth()` using method = 'loess'
```



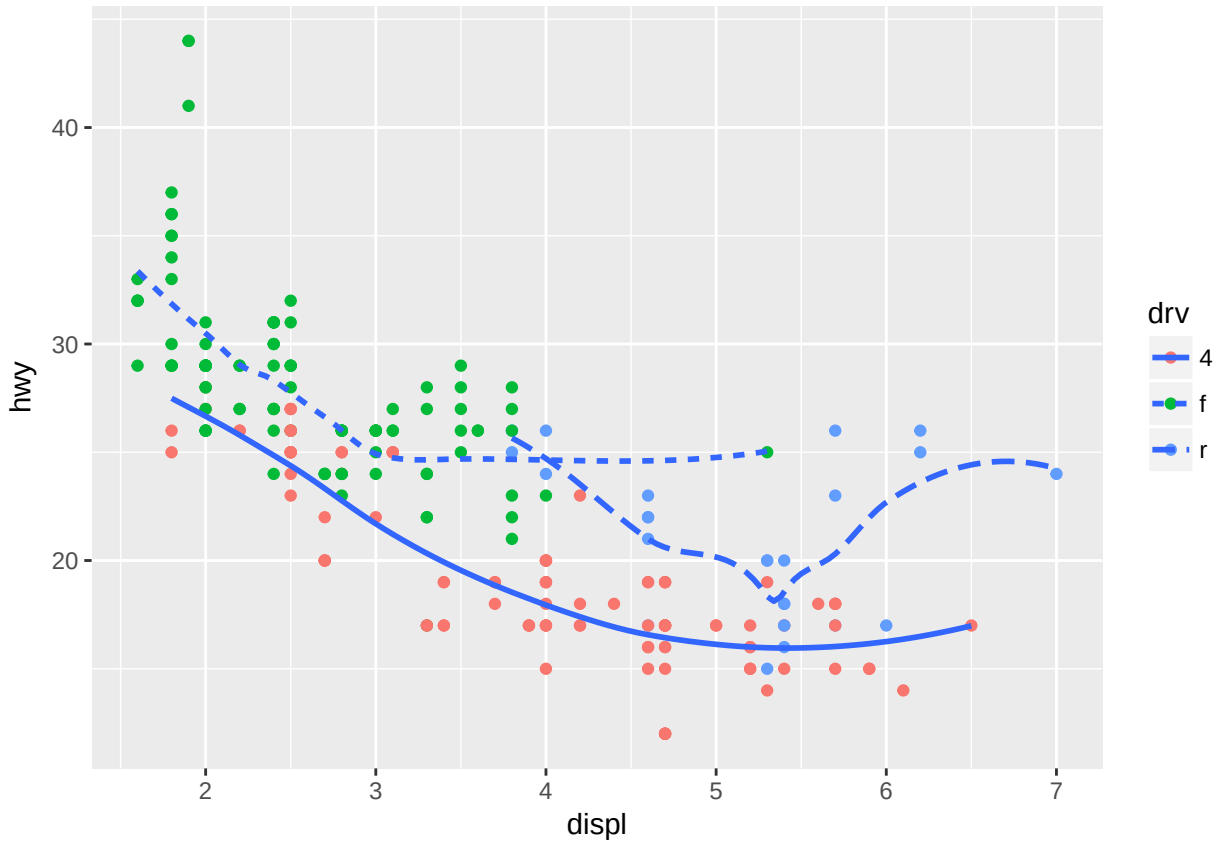

```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) + geom_point(mapping = aes(color = drv)) + geom_sm
```

```
## `geom_smooth()` using method = 'loess'
```

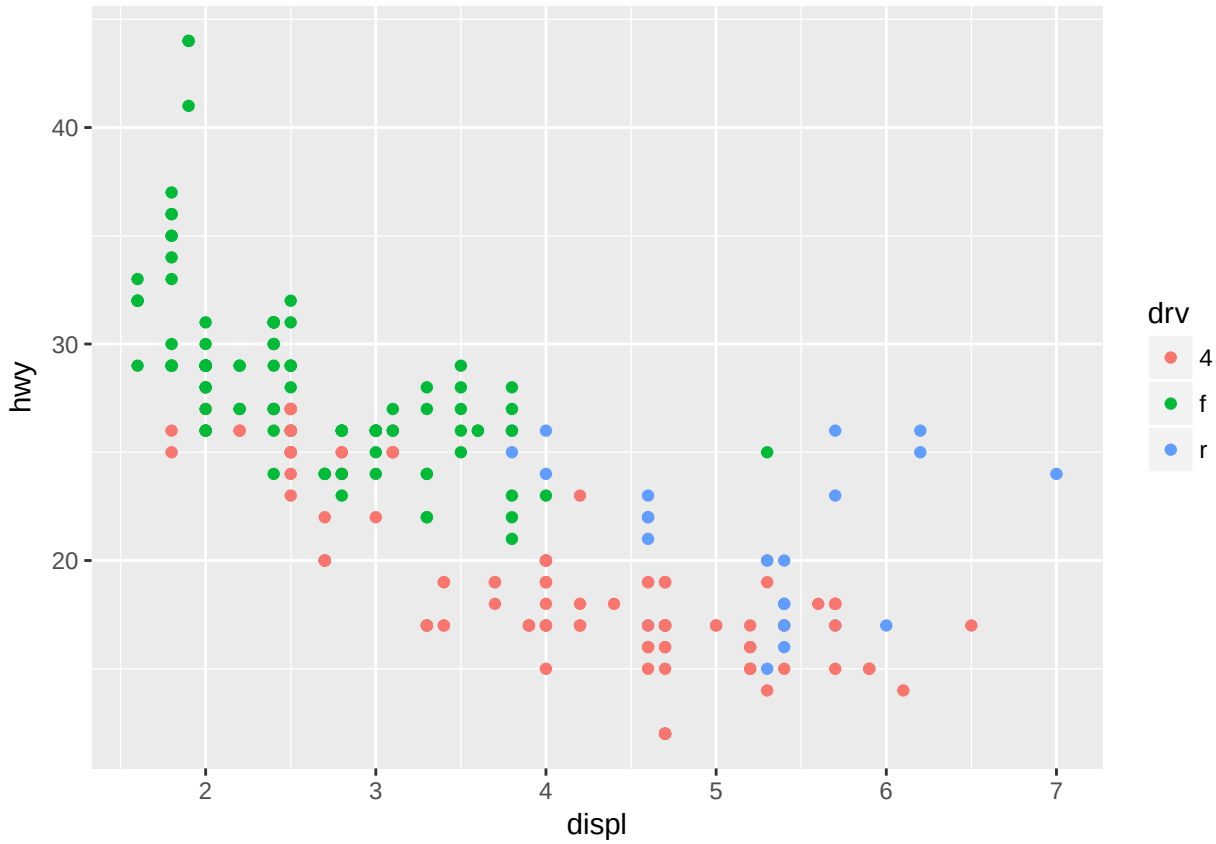



```
ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy, color = drv)) + geom_smooth(mapping =
```

```
## `geom_smooth()` using method = 'loess'
```

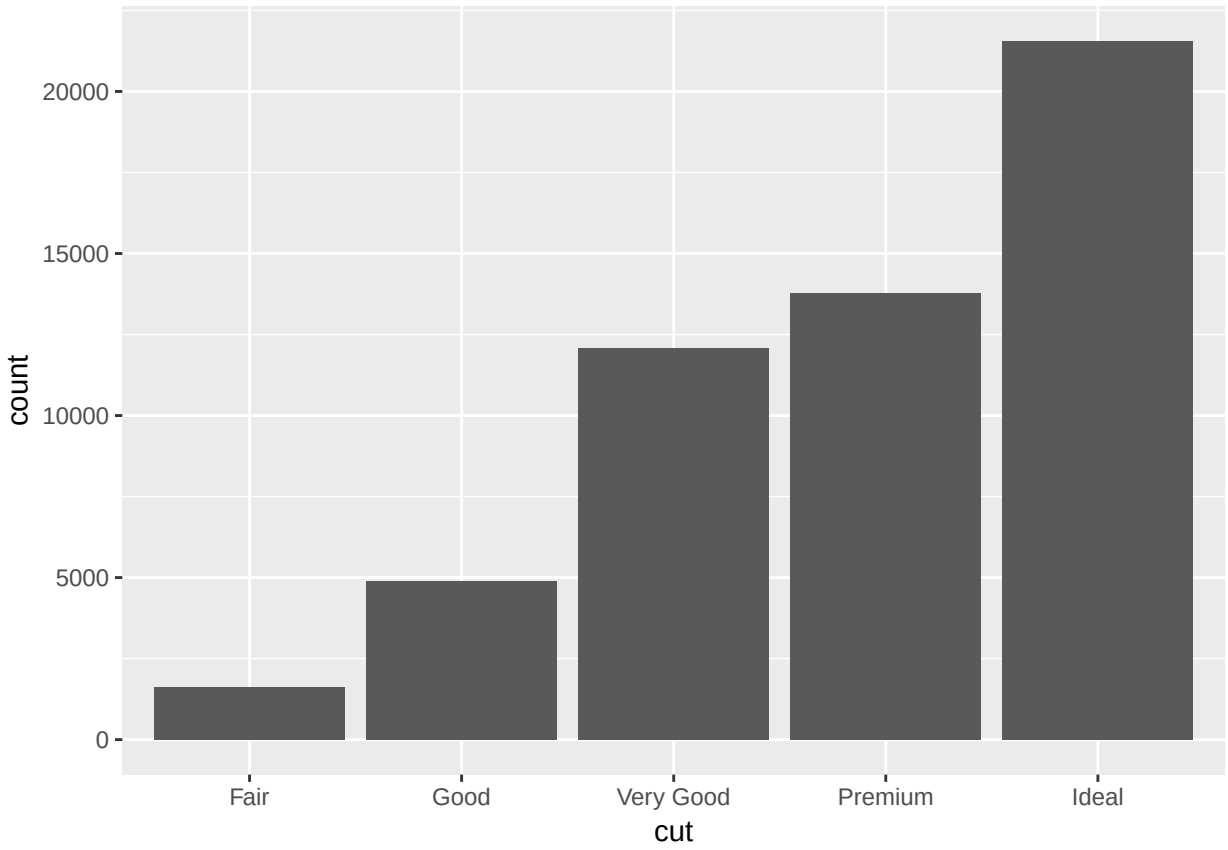



```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy, col = drv)) + geom_point()
```

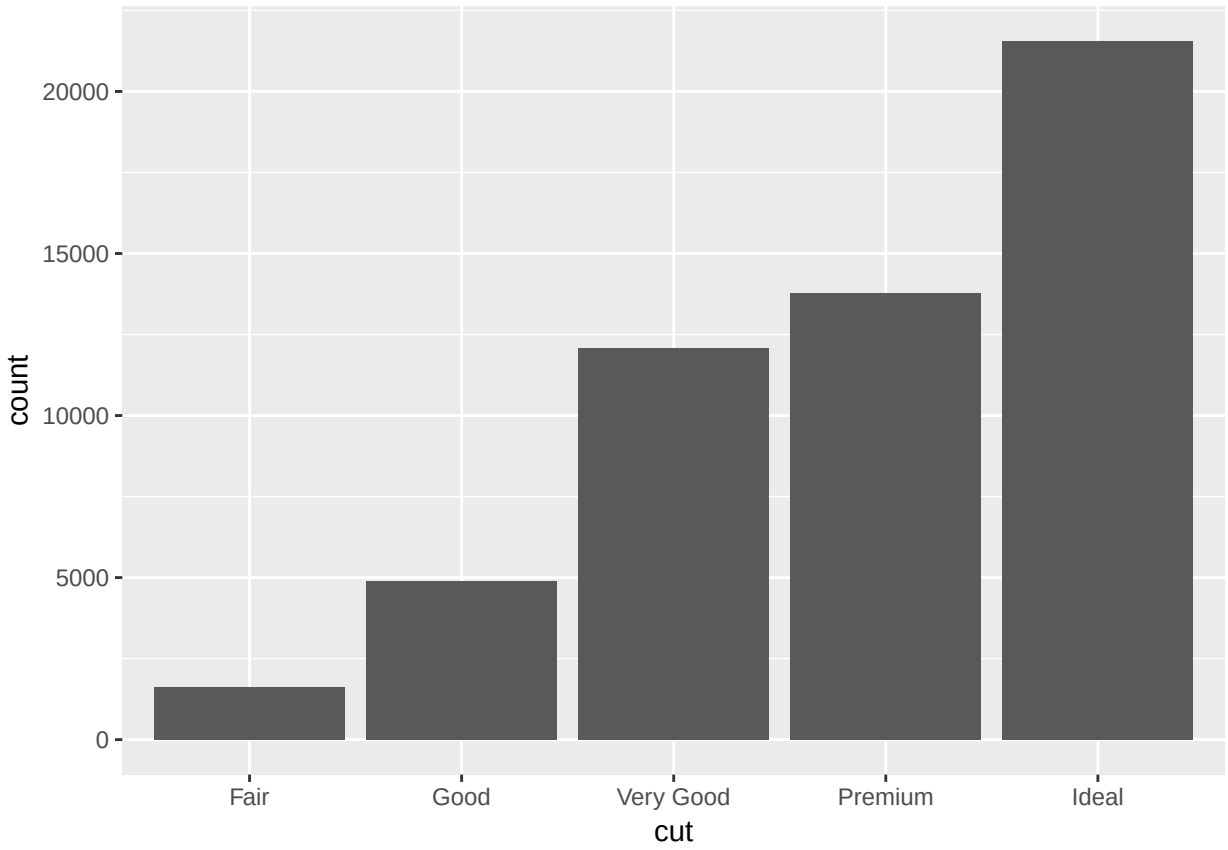


Statistical Transformations

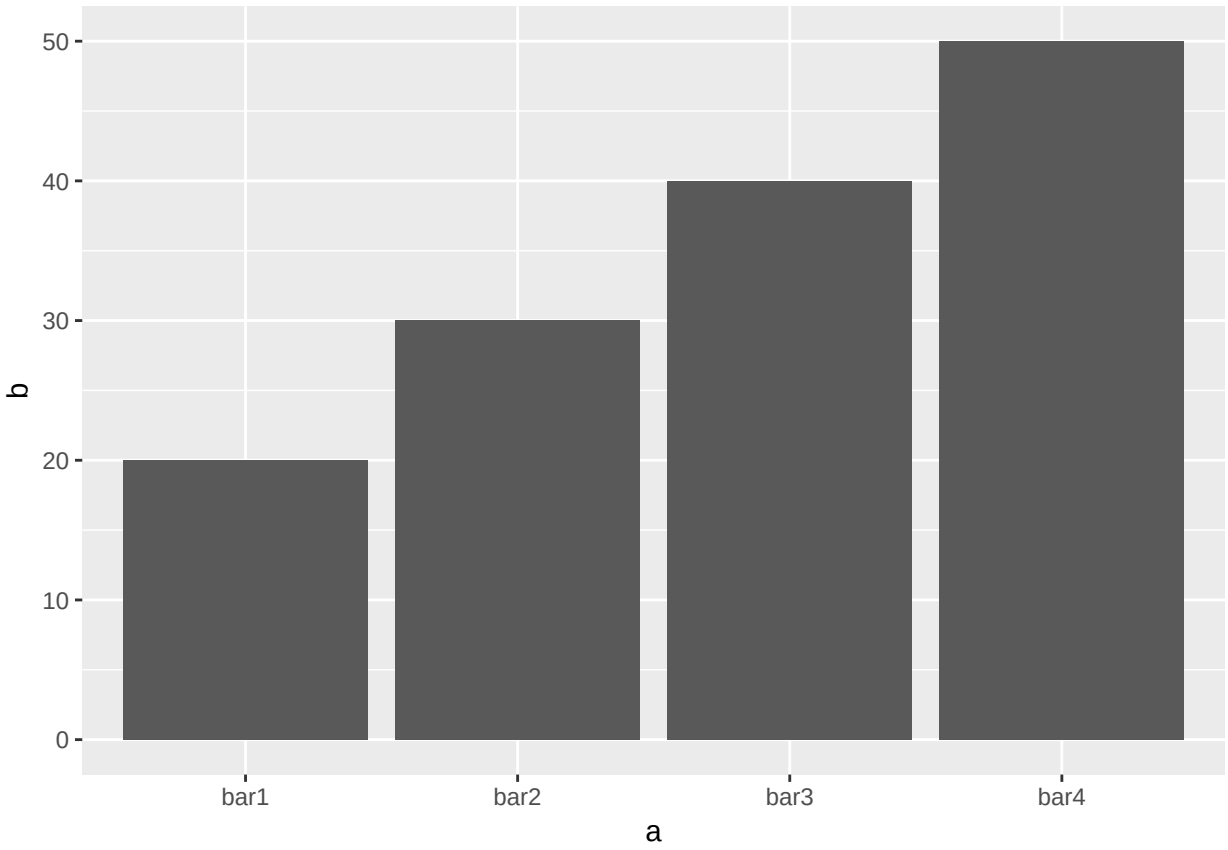
```
diamonds <- ggplot2::diamonds

ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut))
```



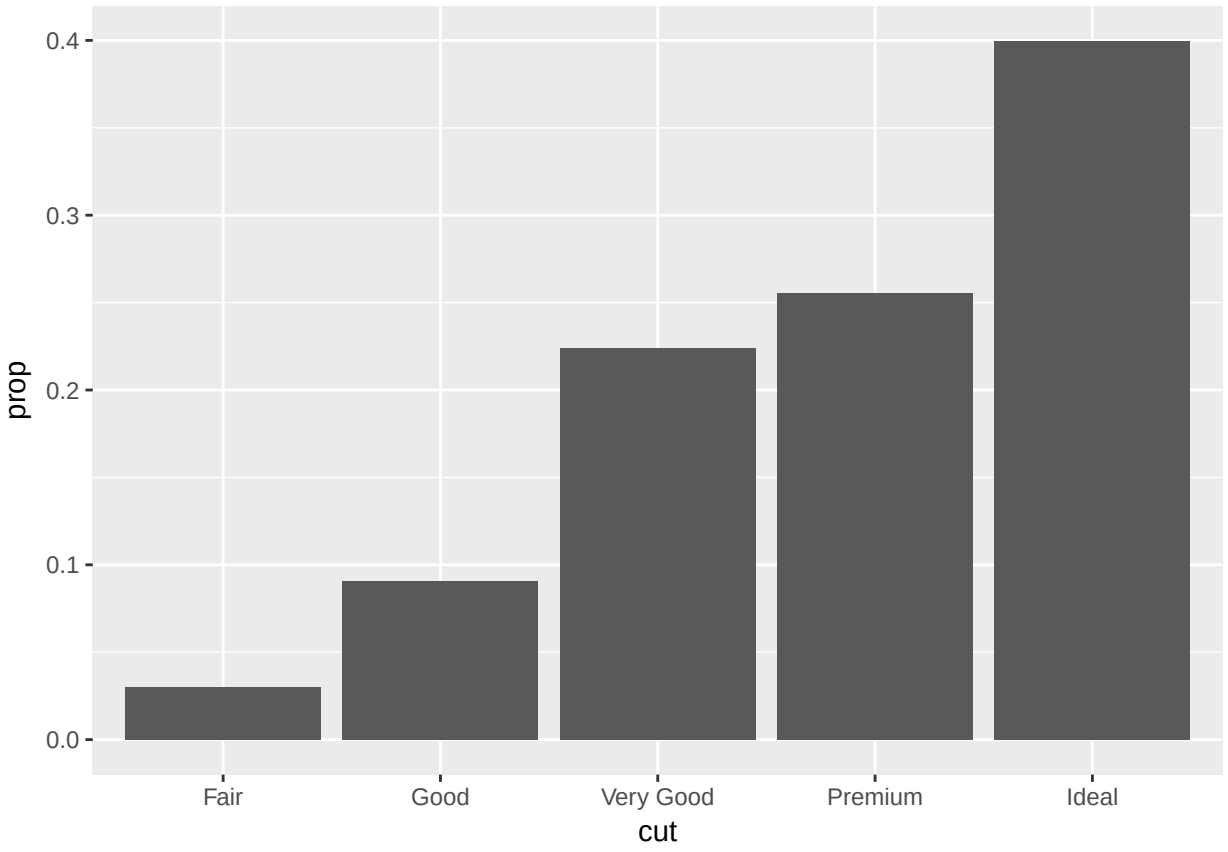

```
# create using statcount instead of geom_bar
ggplot(data = diamonds) +
  stat_count(mapping = aes(x = cut))
```



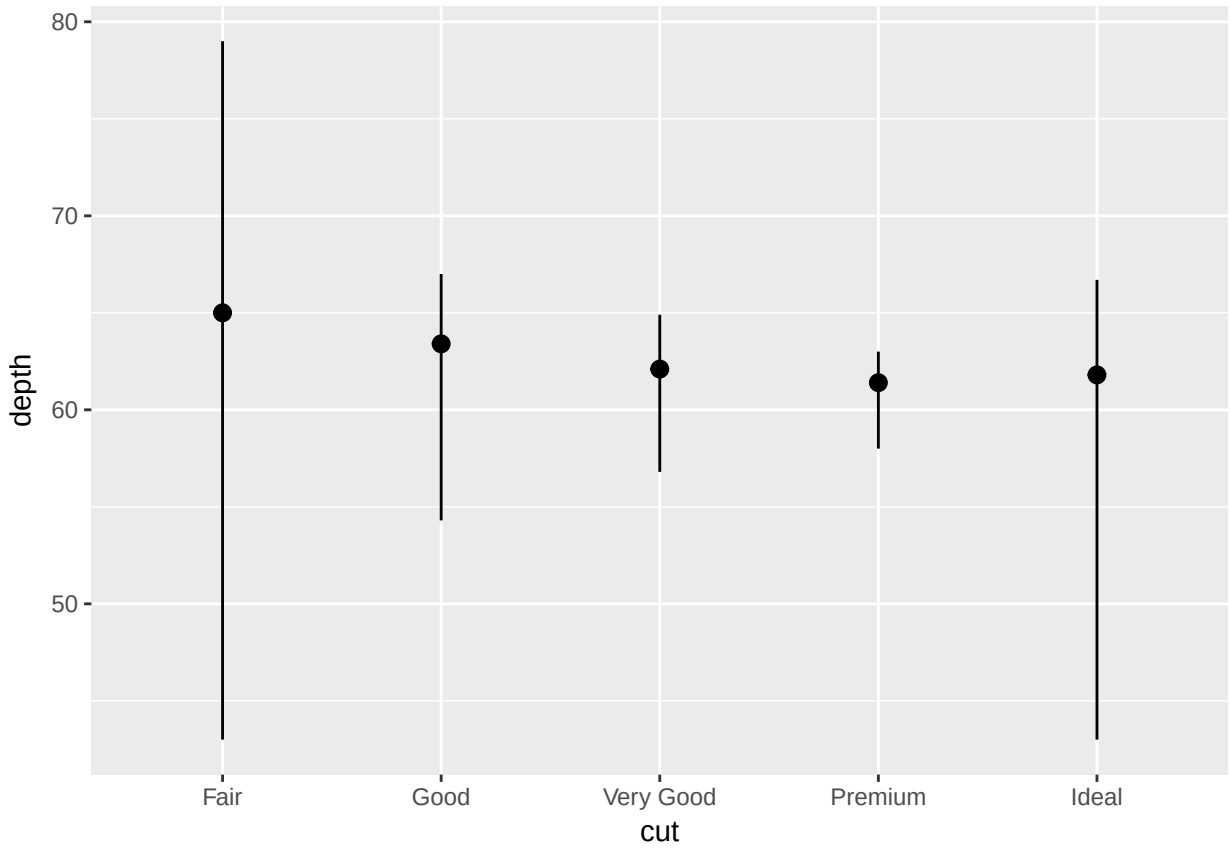

```
# override the default stat. Change stat from count to identity

demo <- tribble(
  ~a, ~b,
  "bar1", 20,
  "bar2", 30,
  "bar3", 40,
  "bar4", 50
)

ggplot(data = demo) +
  geom_bar(mapping = aes(x = a, y = b), stat = "identity")
```




```
# display a bar chart of proportion rather than count
ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut, y = ..prop.., group = 1))
```

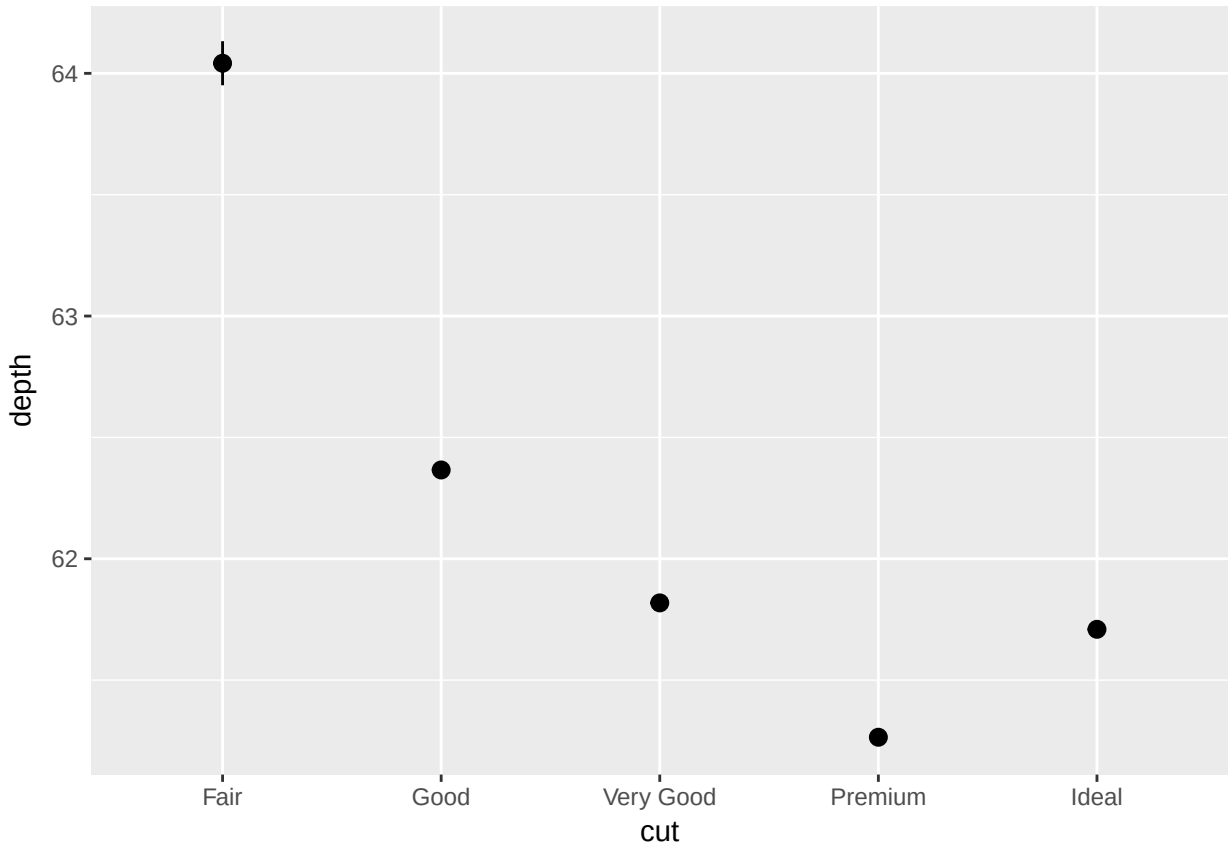



```
# use stat summary to give information about statistical transformations in code
ggplot(data = diamonds) +
  stat_summary(
    mapping = aes(x = cut, y = depth),
    fun.ymin = min,
    fun.ymax = max,
    fun.y = median
  )
```

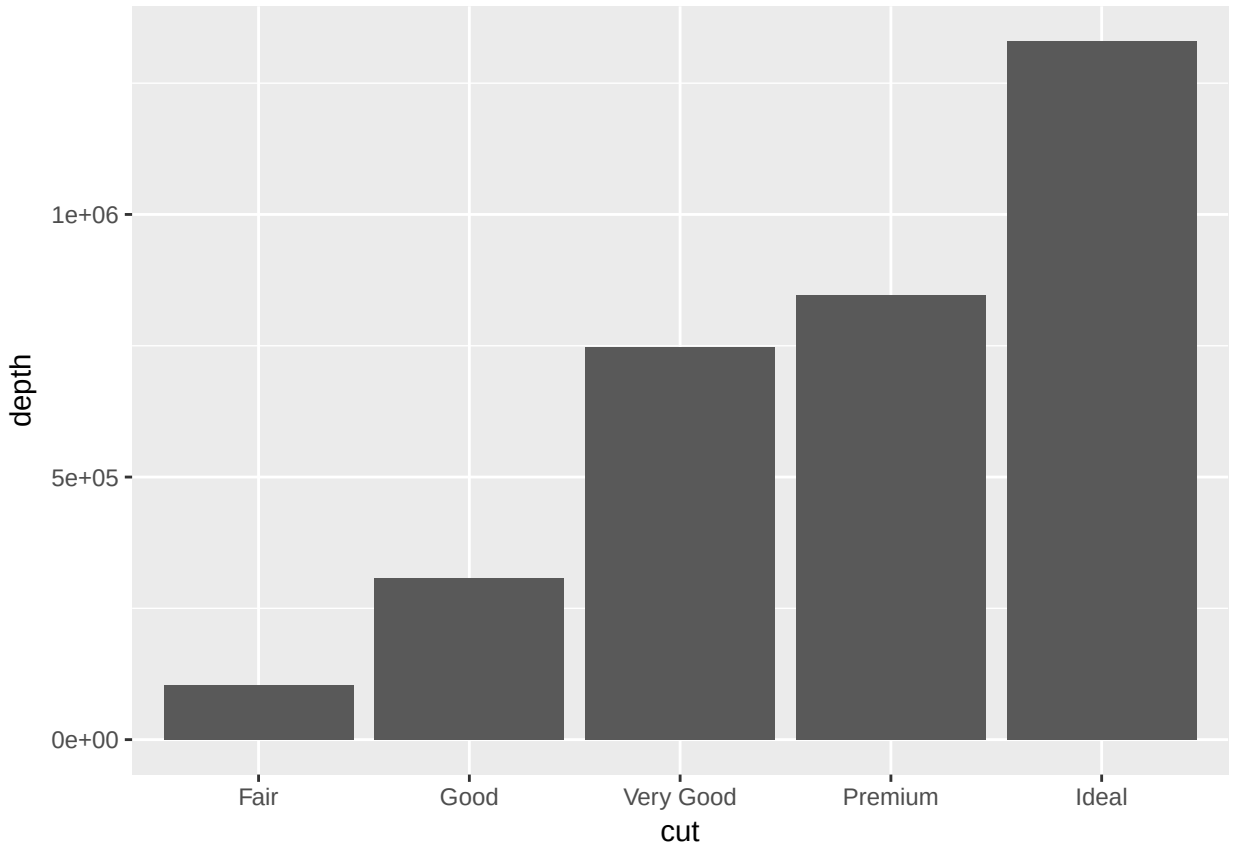



```
#geom_pointrange
ggplot(data = diamonds) +
  geom_pointrange(mapping = aes(x = cut, y = depth), stat = "summary")
```

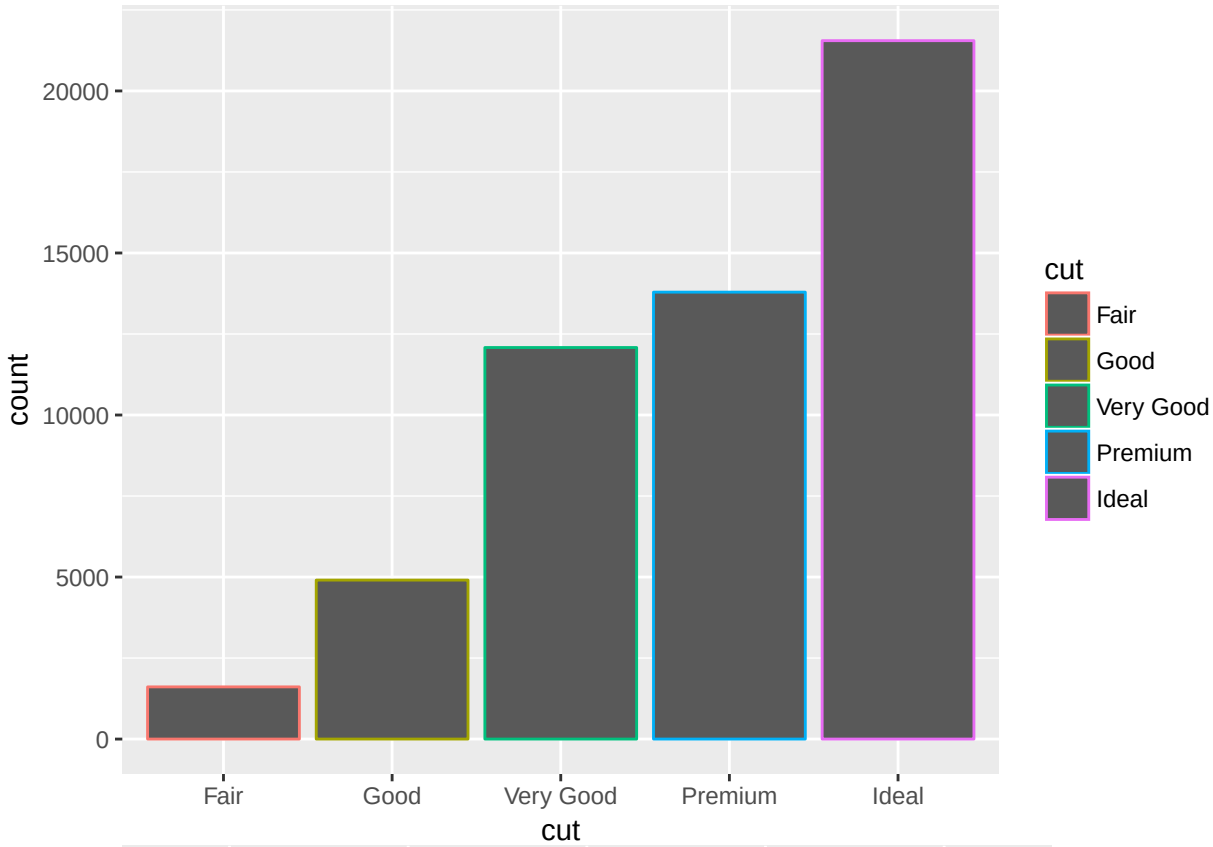
```
## No summary function supplied, defaulting to `mean_se()
```

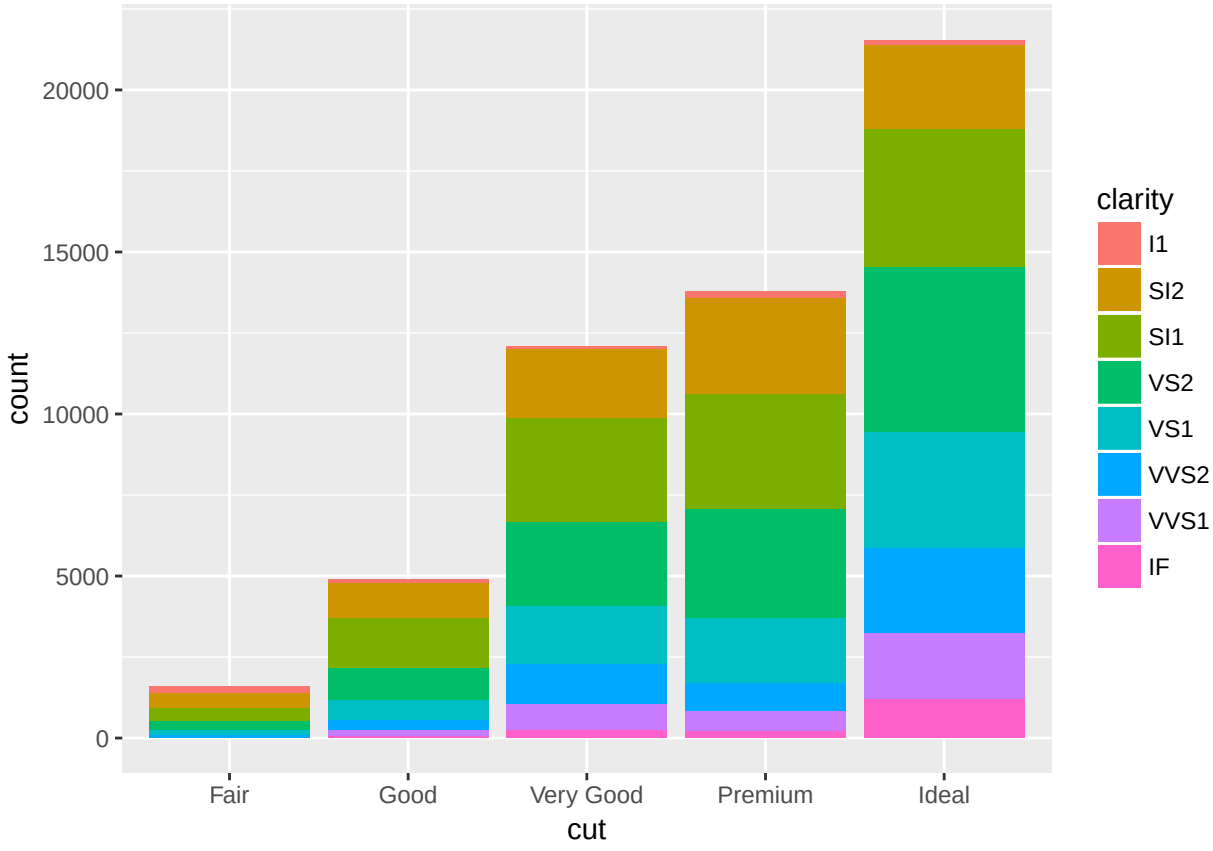


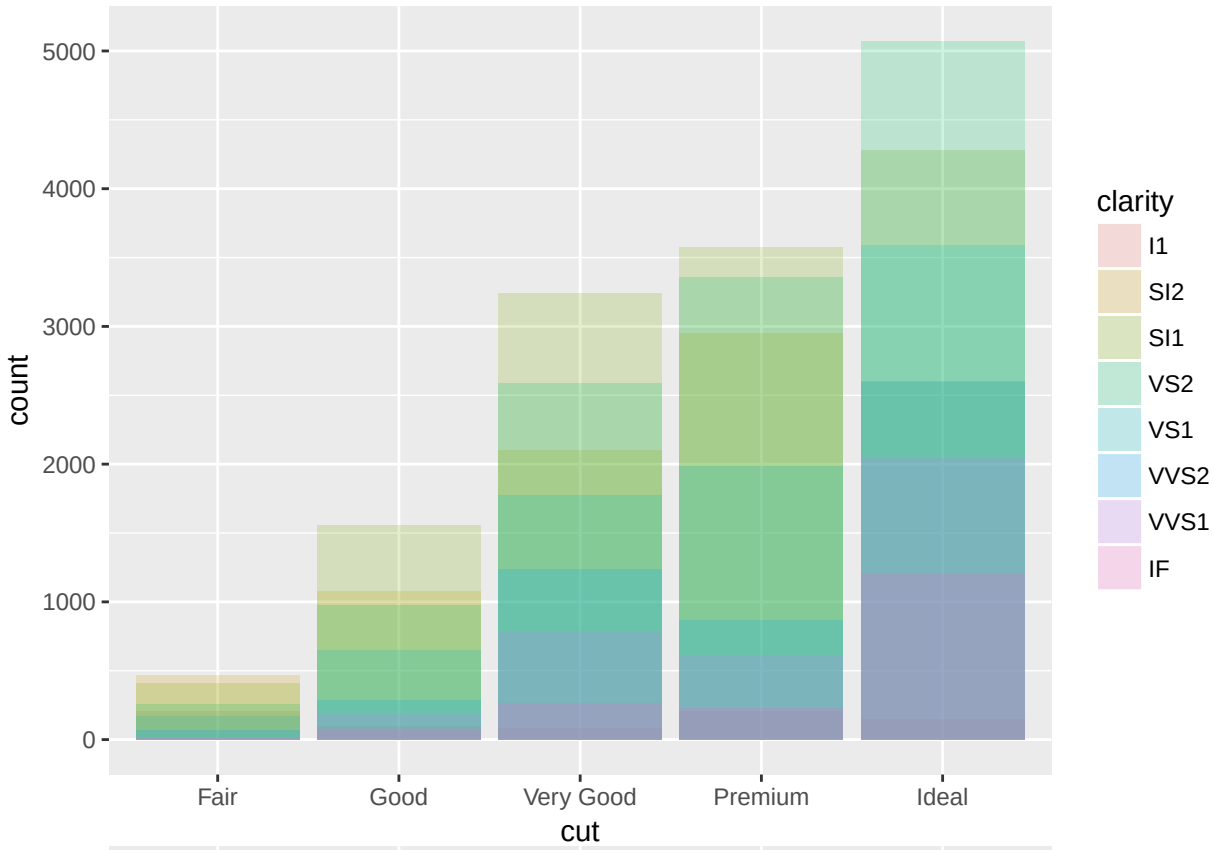

```
#geom_col()
ggplot(data = diamonds) +
  geom_col(mapping = aes(x = cut, y = depth))
```

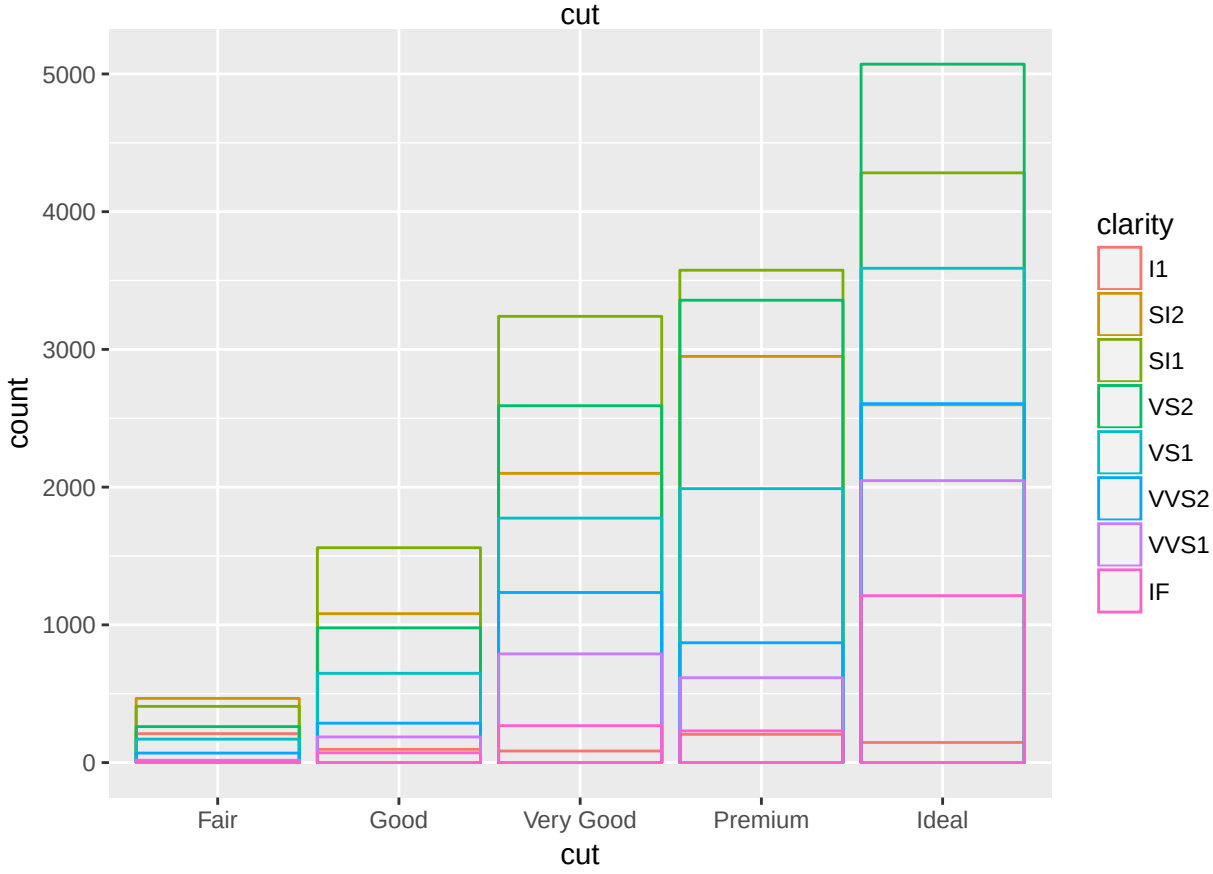

depth
1e+06
5e+05
0e+00
Fair
Good
Very Good
Premium
Ideal
cut

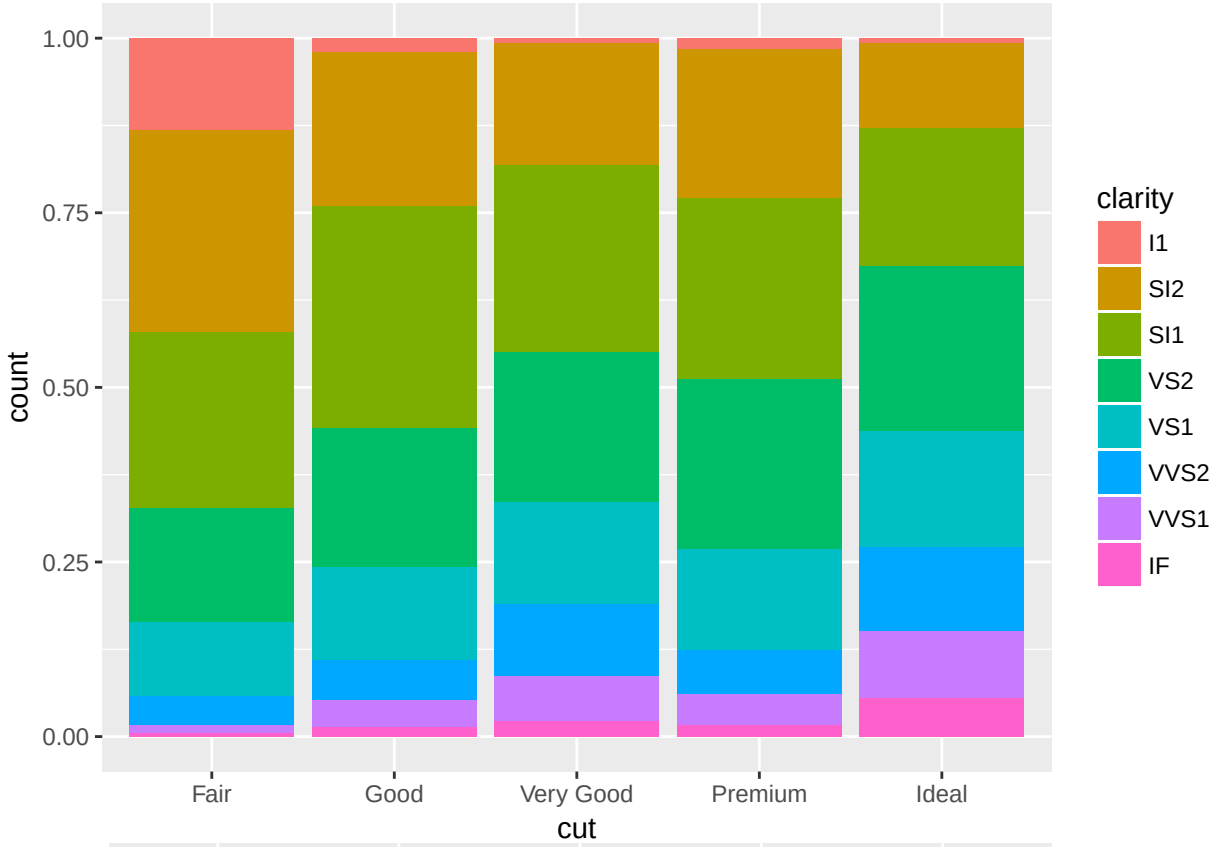
Position Adjustments

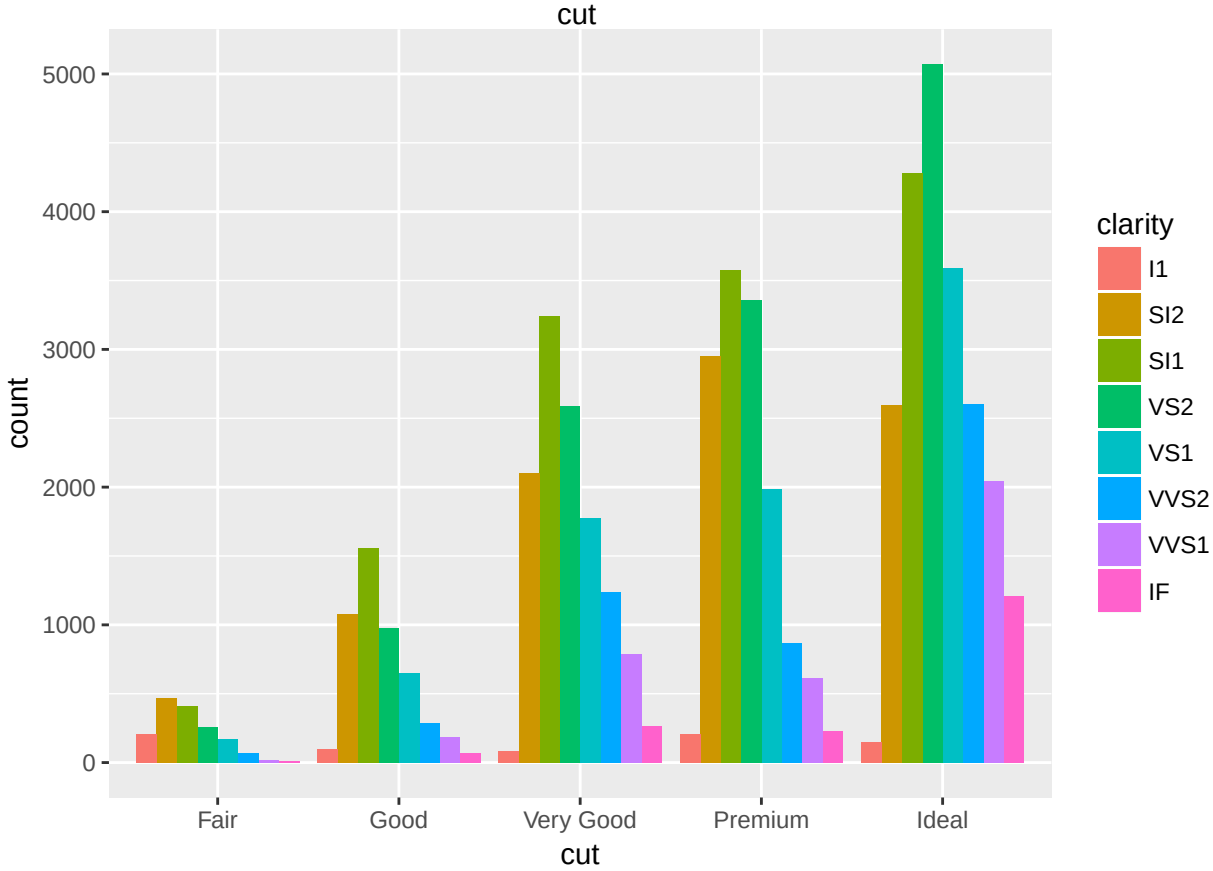


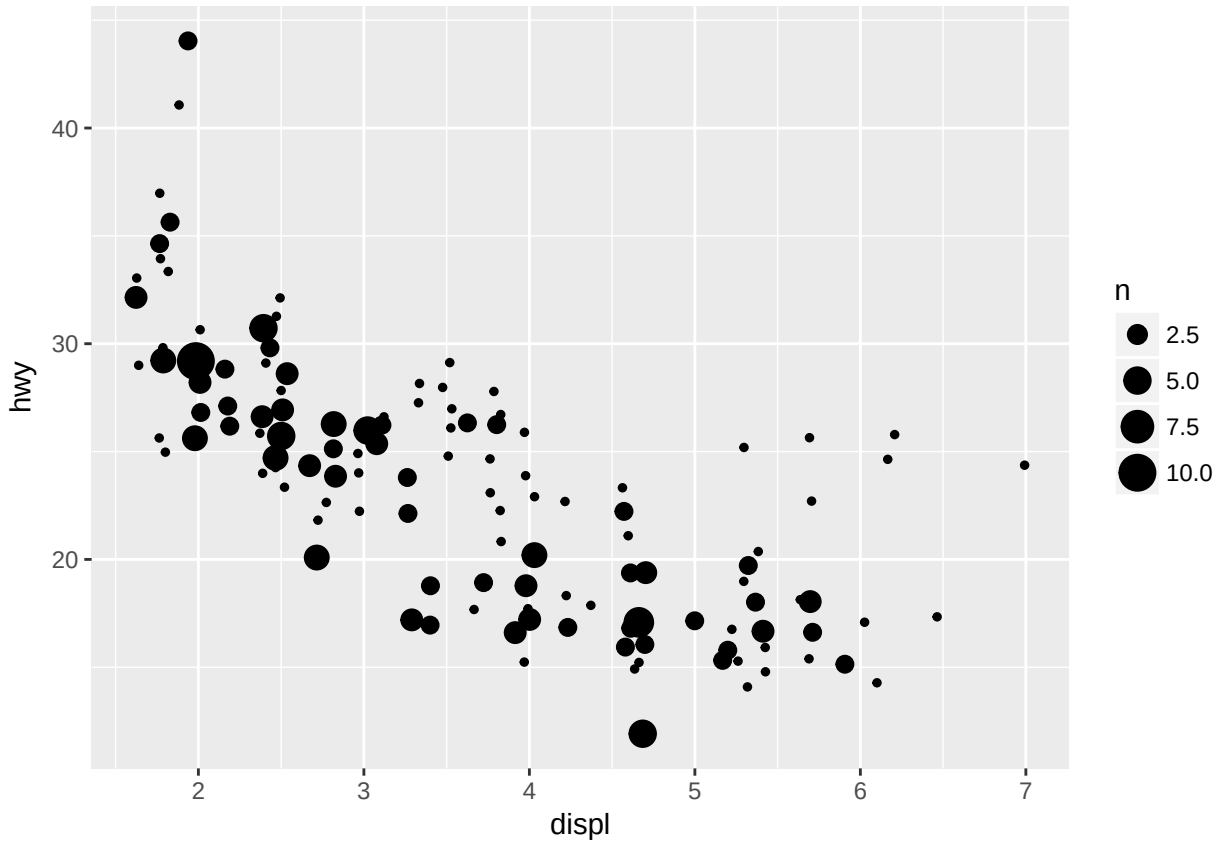


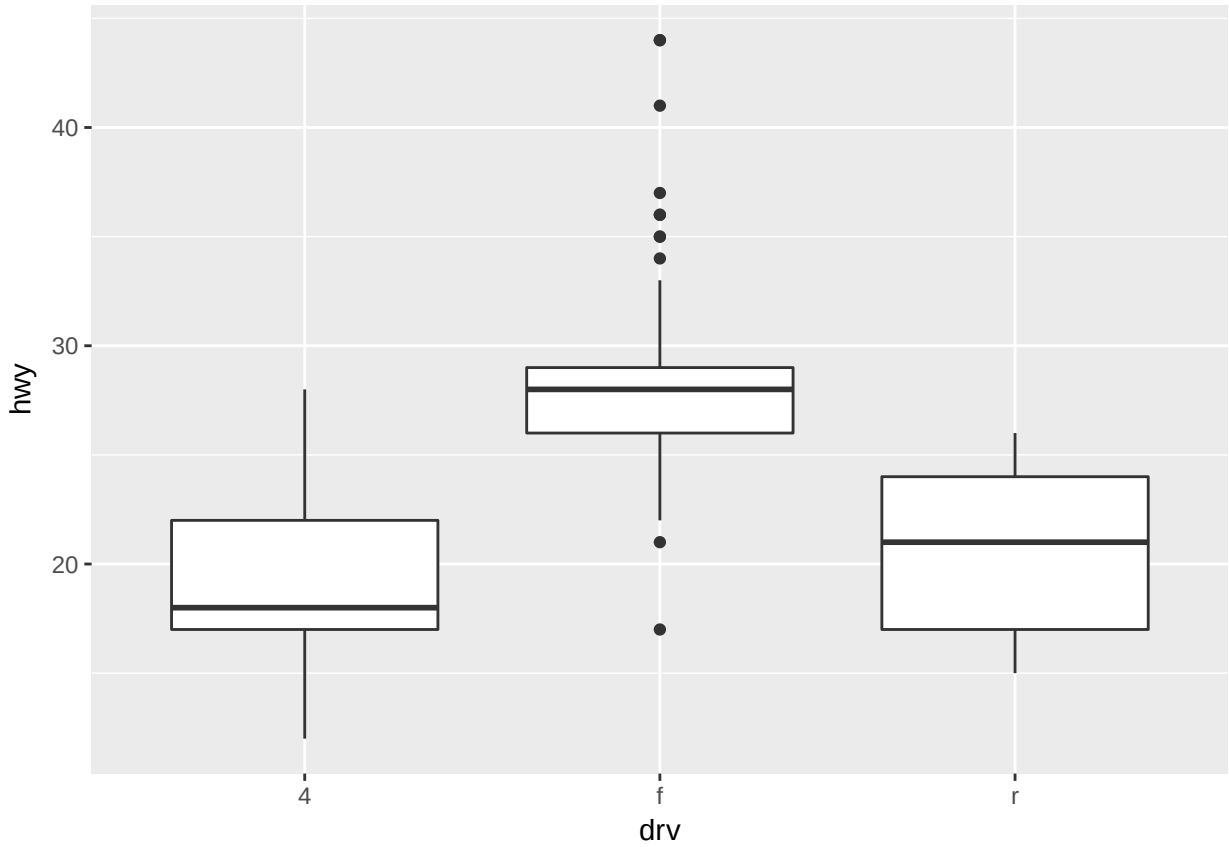

5000
4000
3000
2000
1000
0
count
Fair
Good
Very Good
Premium
Ideal
cut
clarity
I1
SI2
SI1
VS2
VS1
VVS2
VVS1
IF


5000
4000
3000
2000
1000
0
count
Fair
Good
Very Good
Premium
Ideal
cut
clarity
I1
SI2
SI1
VS2
VS1
VVS2
VVS1
IF


1.00
0.75
0.50
0.25
0.00
count
Fair
Good
Very Good
Premium
Ideal
cut
clarity
I1
SI2
SI1
VS2
VS1
VVS2
VVS1
IF

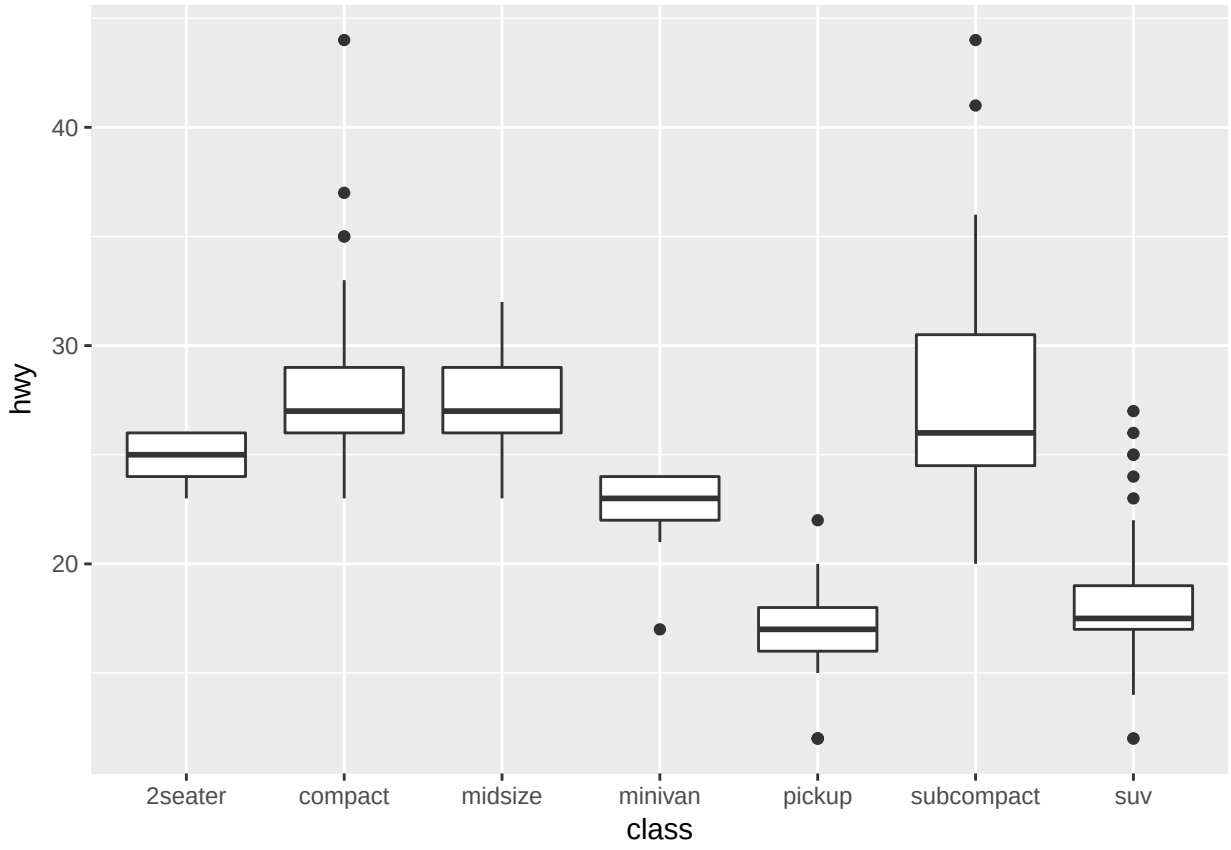

5000
4000
3000
2000
1000
0
count
Fair
Good
Very Good
Premium
Ideal
cut
clarity
I1
SI2
SI1
VS2
VS1
VVS2
VVS1
IF


hwy
40
30
20
2
3
4
5
6
7
displ
n
2.5
5.0
7.5
10.0

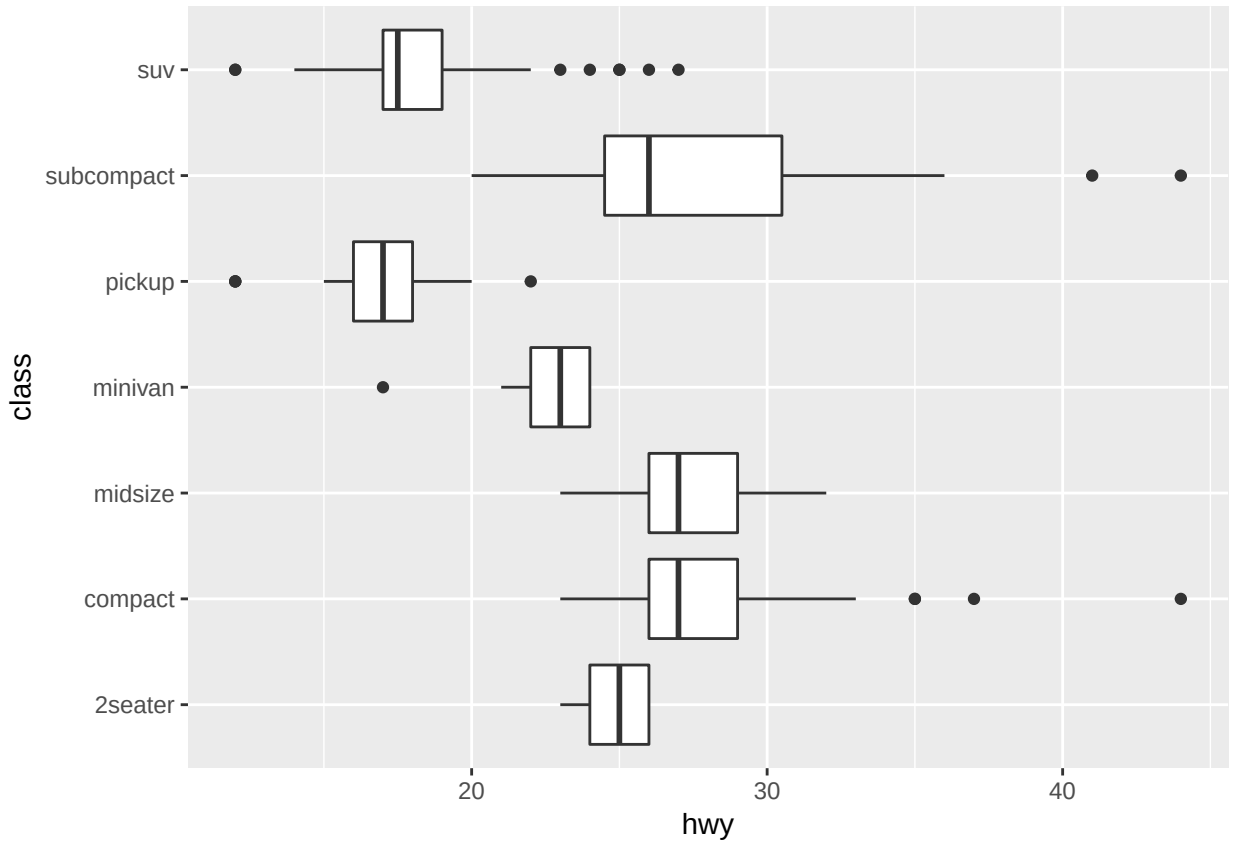

hwy
40
30
20
4
f
r
drv

Coordinate Systems

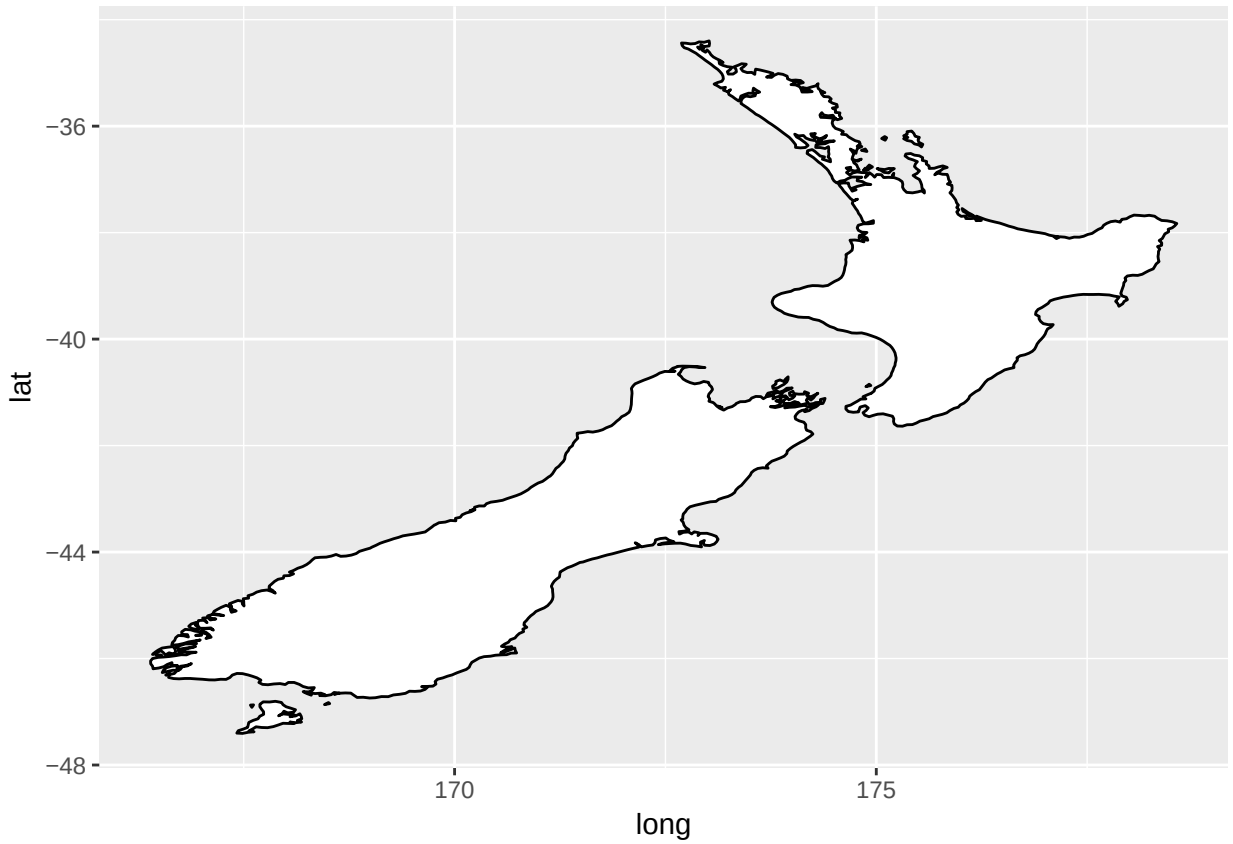
```
#coord_flip
ggplot(data = mpg, mapping = aes(x = class, y = hwy)) + geom_boxplot()
```



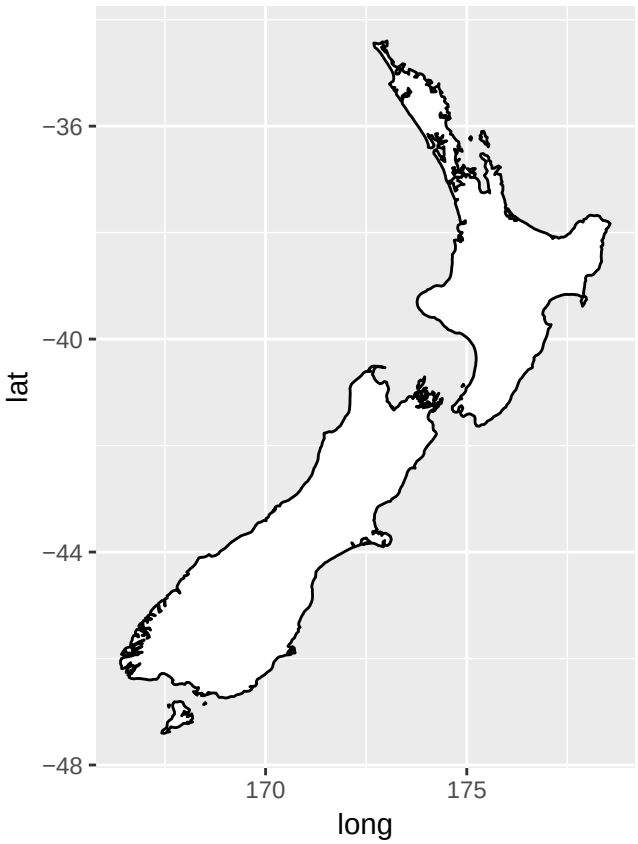

```
ggplot(data = mpg, mapping = aes(x = class, y = hwy)) + geom_boxplot() + coord_flip()
```



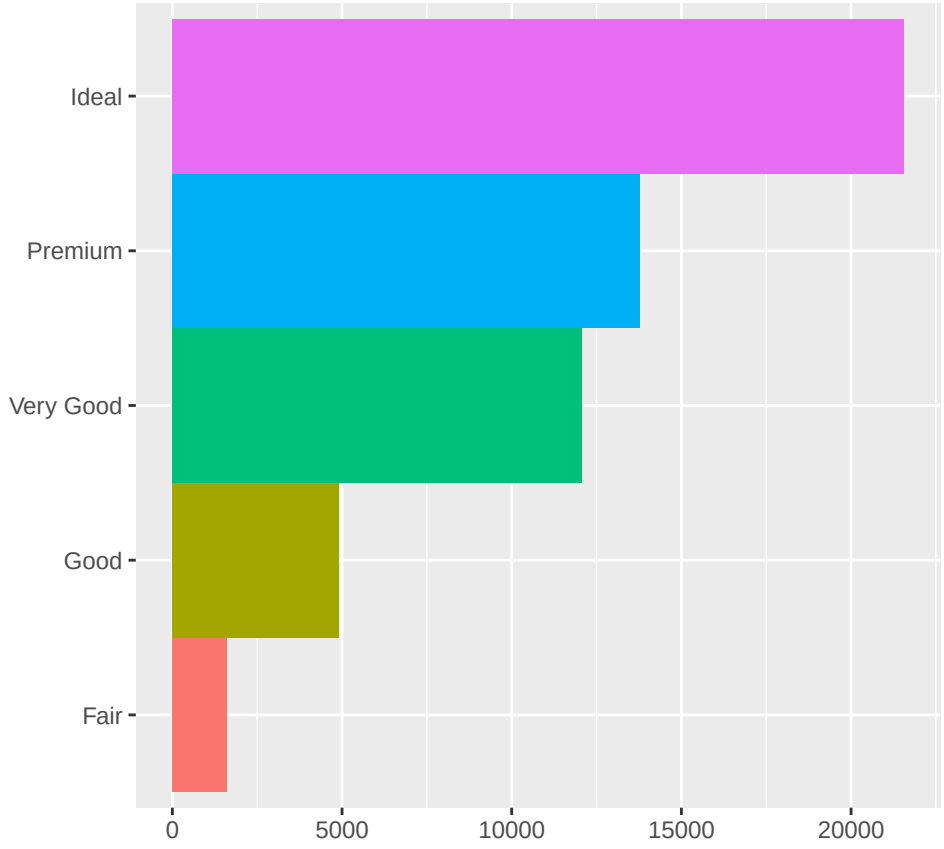

```
#coord_quickmap
nz <- map_data("nz")

ggplot(nz, aes(long, lat, group = group)) +
  geom_polygon(fill = "white", color = "black")
```



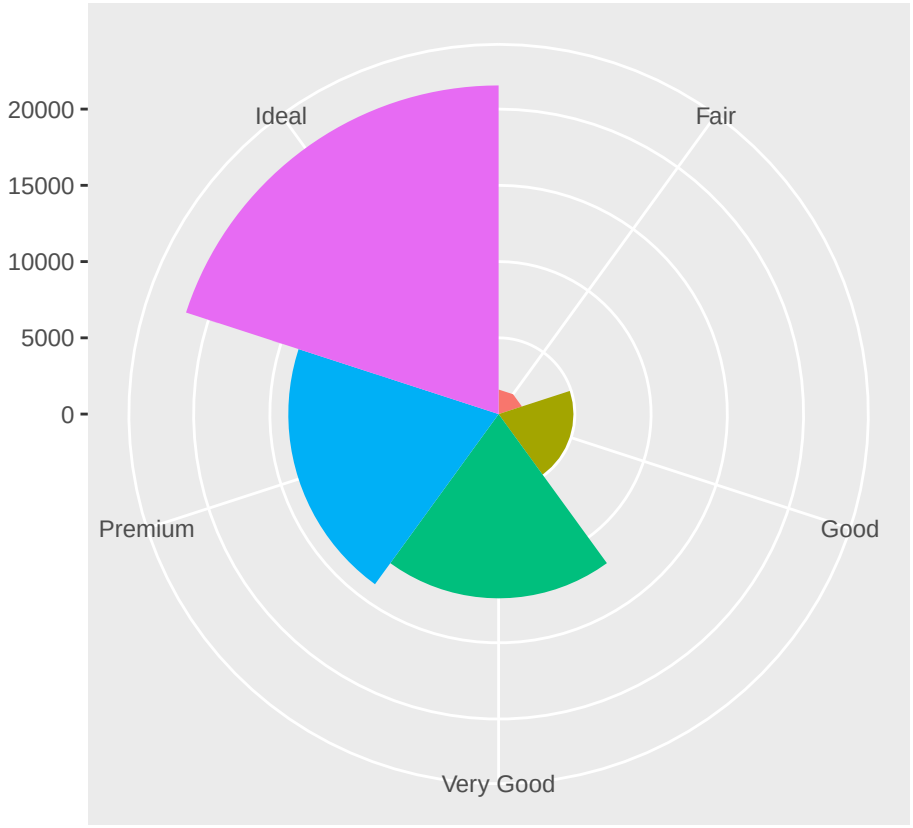

```
ggplot(nz, aes(long, lat, group = group)) +
  geom_polygon(fill = "white", color = "black") +
    coord_quickmap()
```



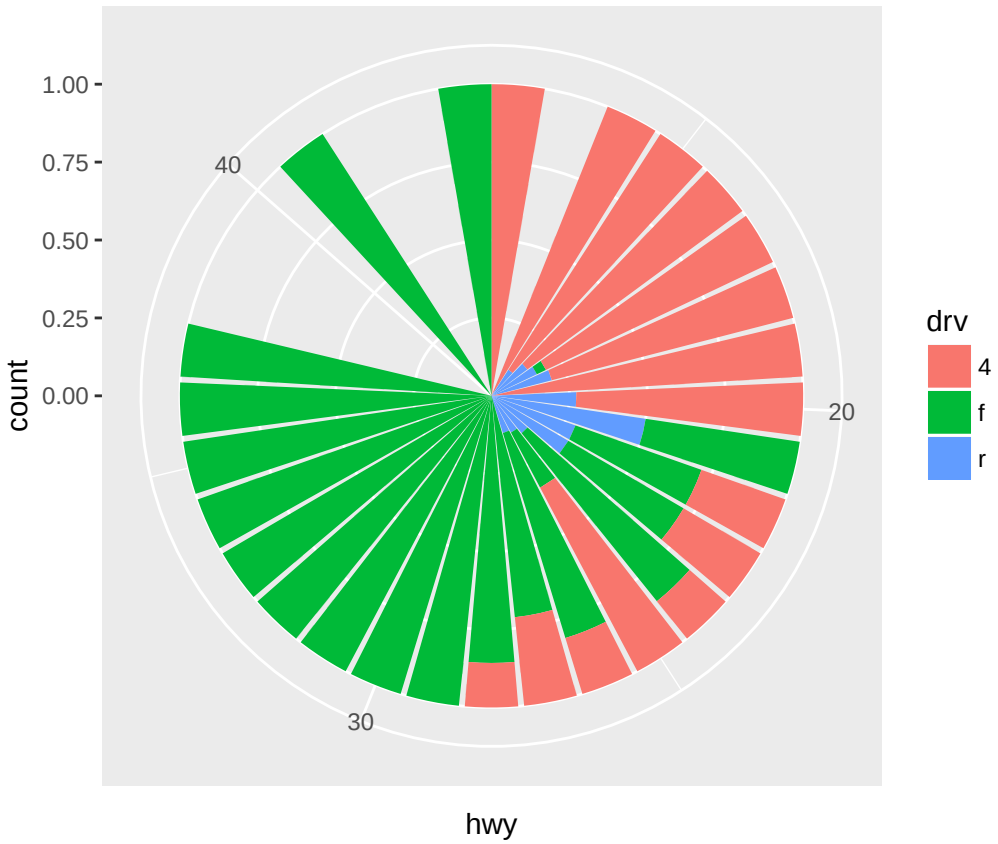

```
#coord_polar
bar <- ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut, fill = cut), show.legend = FALSE, width = 1) +
    theme(aspect.ratio = 1) +
      labs(x = NULL, y = NULL)

bar + coord_flip()
```




```
bar + coord_polar()
```




```
ggplot(data = mpg) +
  geom_bar(mapping = aes(x = hwy, fill = drv), position = "fill") +
    coord_polar()
```


1.00
0.75
0.50
0.25
0.00
count
40
20
30
hwy
drv
4
f
r