Graphs_In_R.R

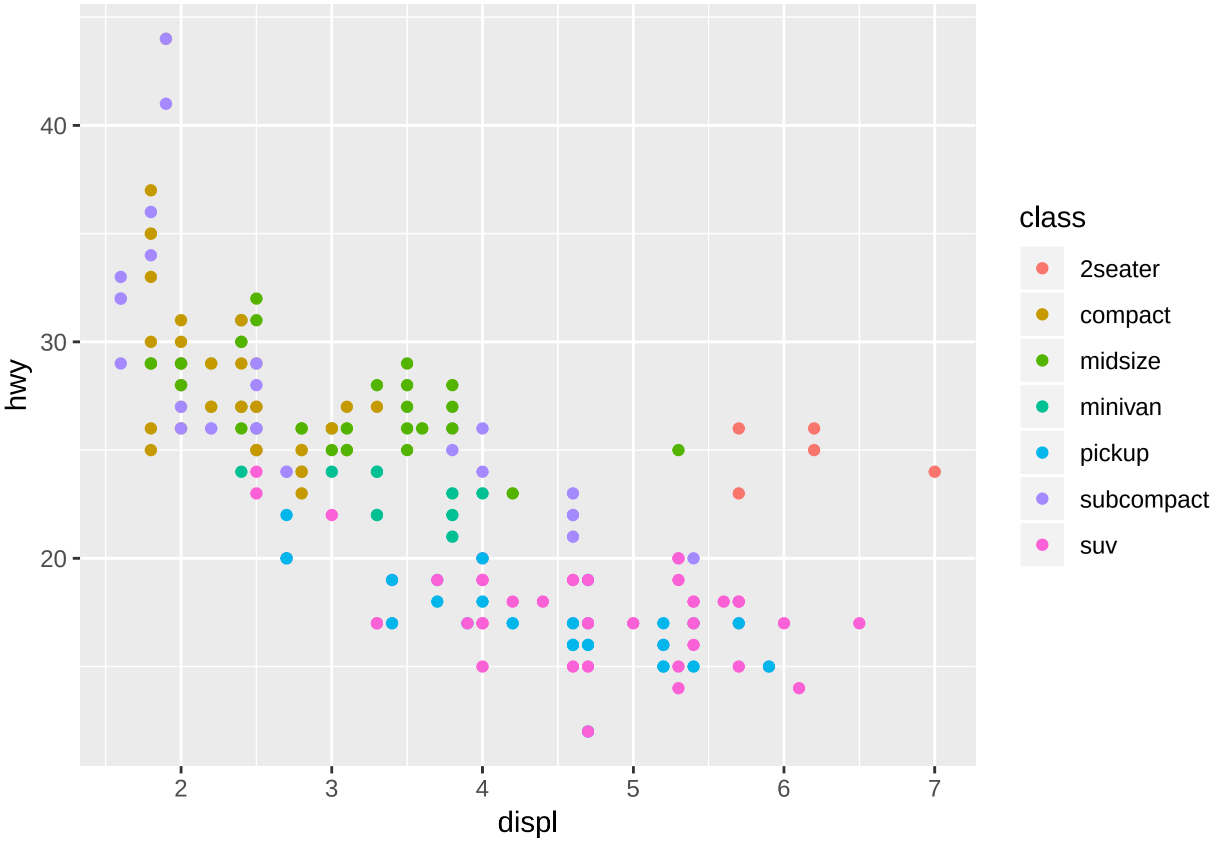
dieudonneouedraogo

Tue Mar 5 17:58:02 2019

```
#################### Welcome to graphs in R, A collection of short codes ###############

########################Dieudonne Ouedraogo  03-05-2019################################

library(tidyverse)
```

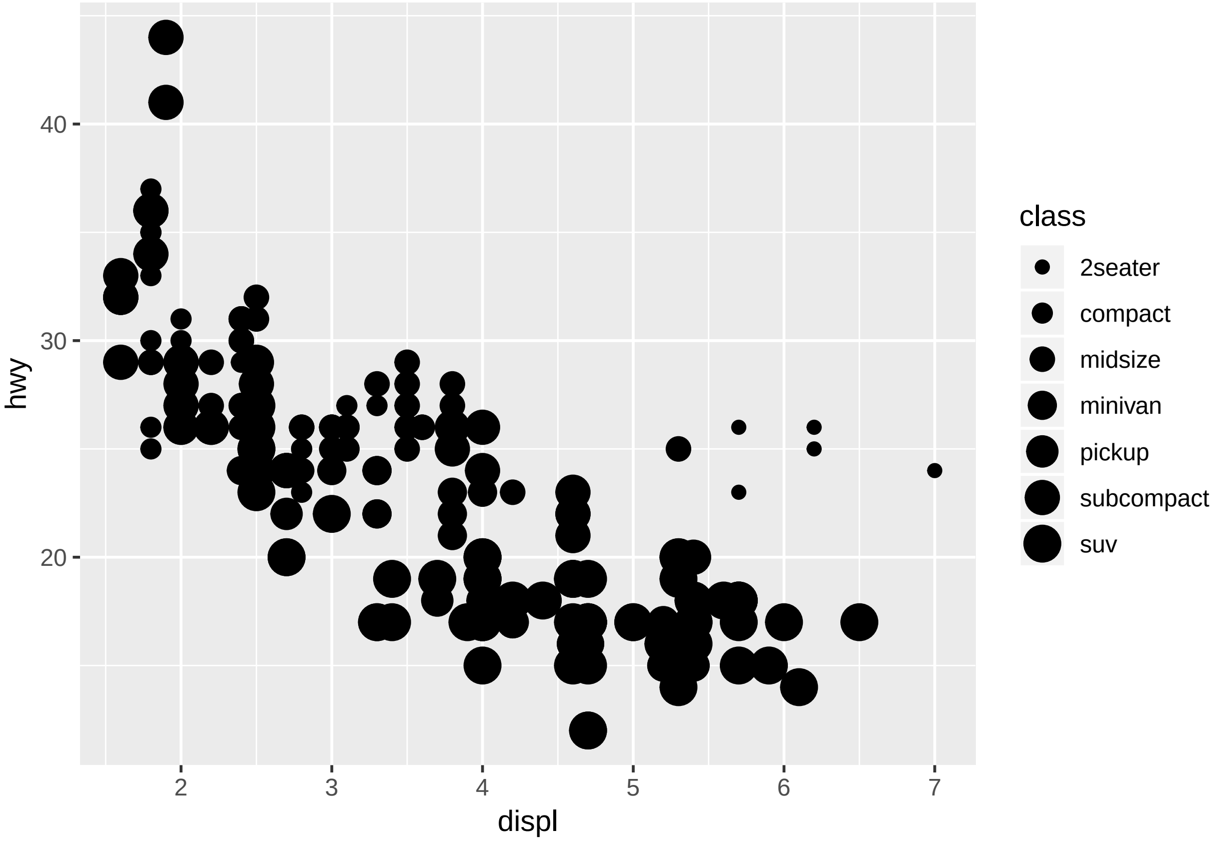
```
## -- Attaching packages --------------------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.1.0     v purrr   0.2.5
## v tibble  2.0.1     v dplyr   0.7.8
## v tidyr   0.8.2     v stringr 1.3.1
## v readr   1.3.1     v forcats 0.3.0

## Warning: package 'tibble' was built under R version 3.5.2

## -- Conflicts ------------------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

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



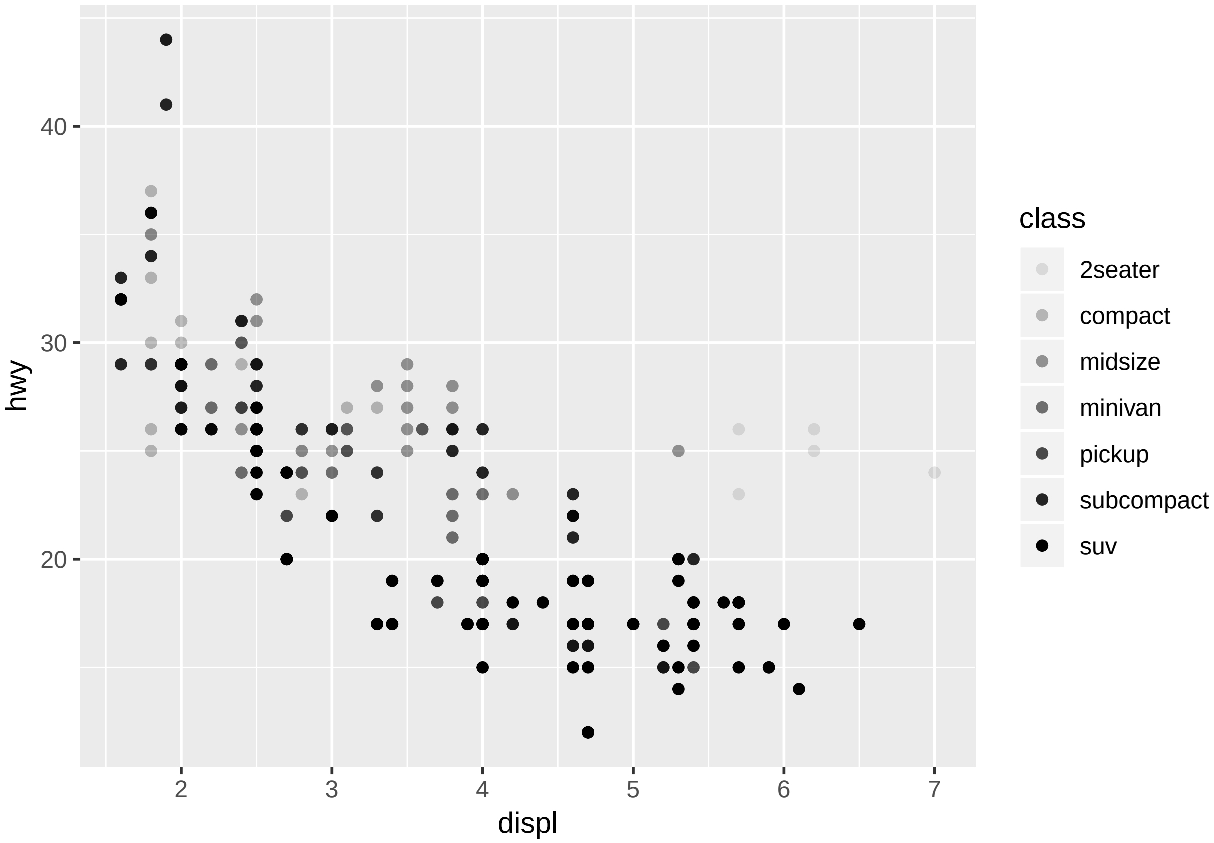
```
################## size by class #################################
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, size = class))
```

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



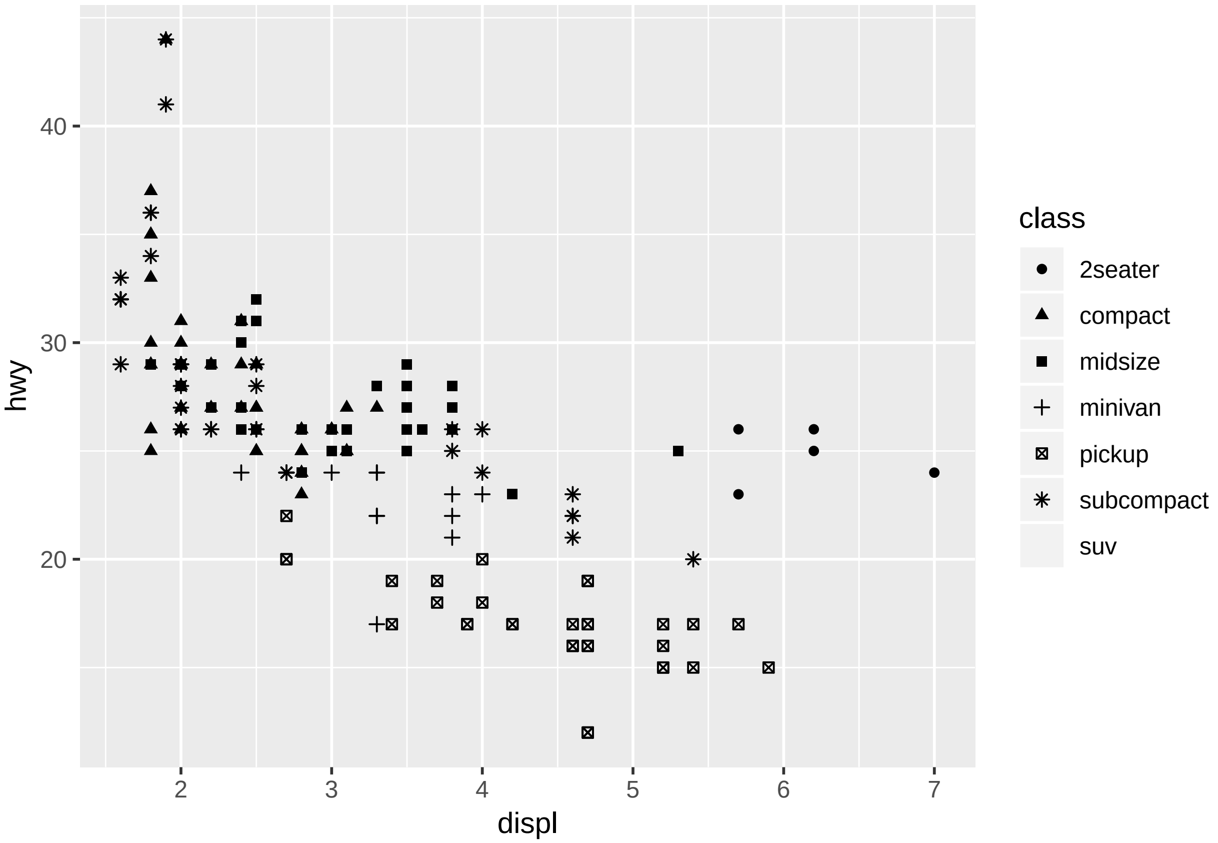
```
################## Transparency###################################
#
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, alpha = class))
```

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

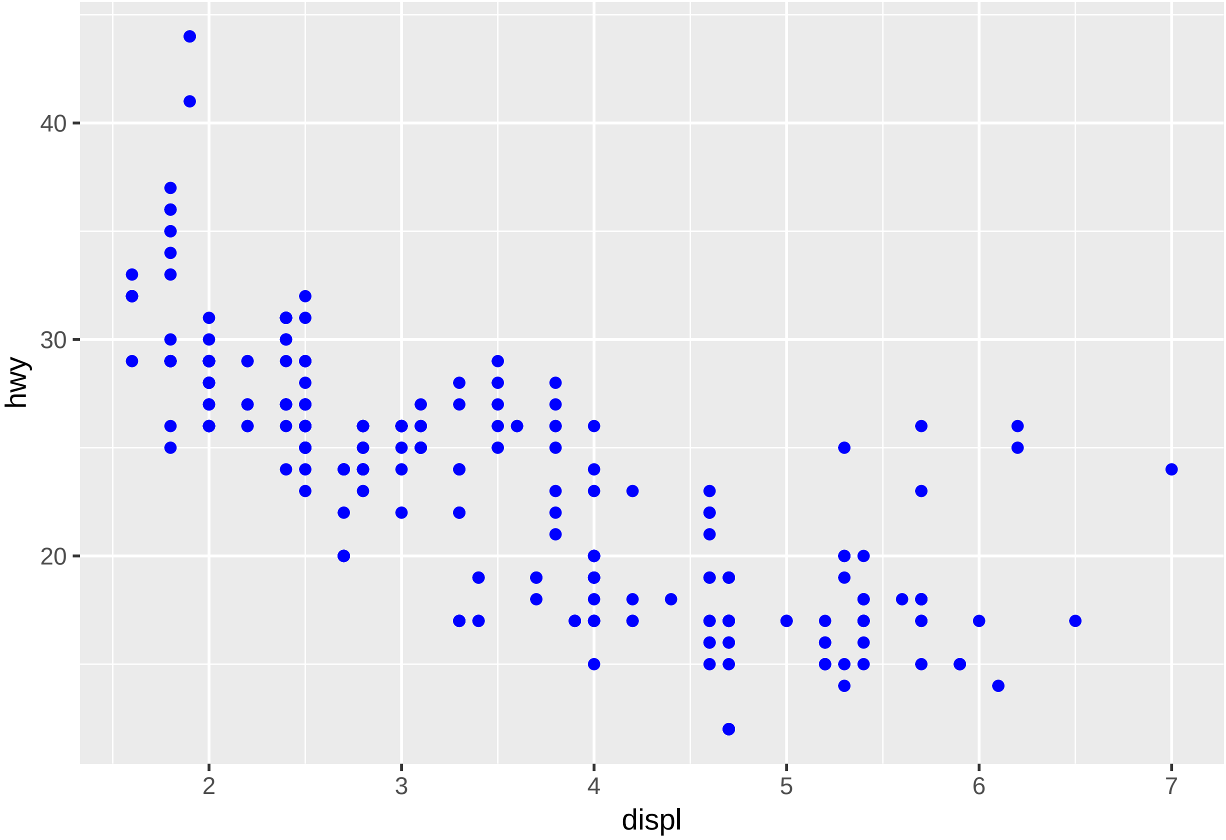


```
#################### Shape ########################################
#
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, shape = class))
```

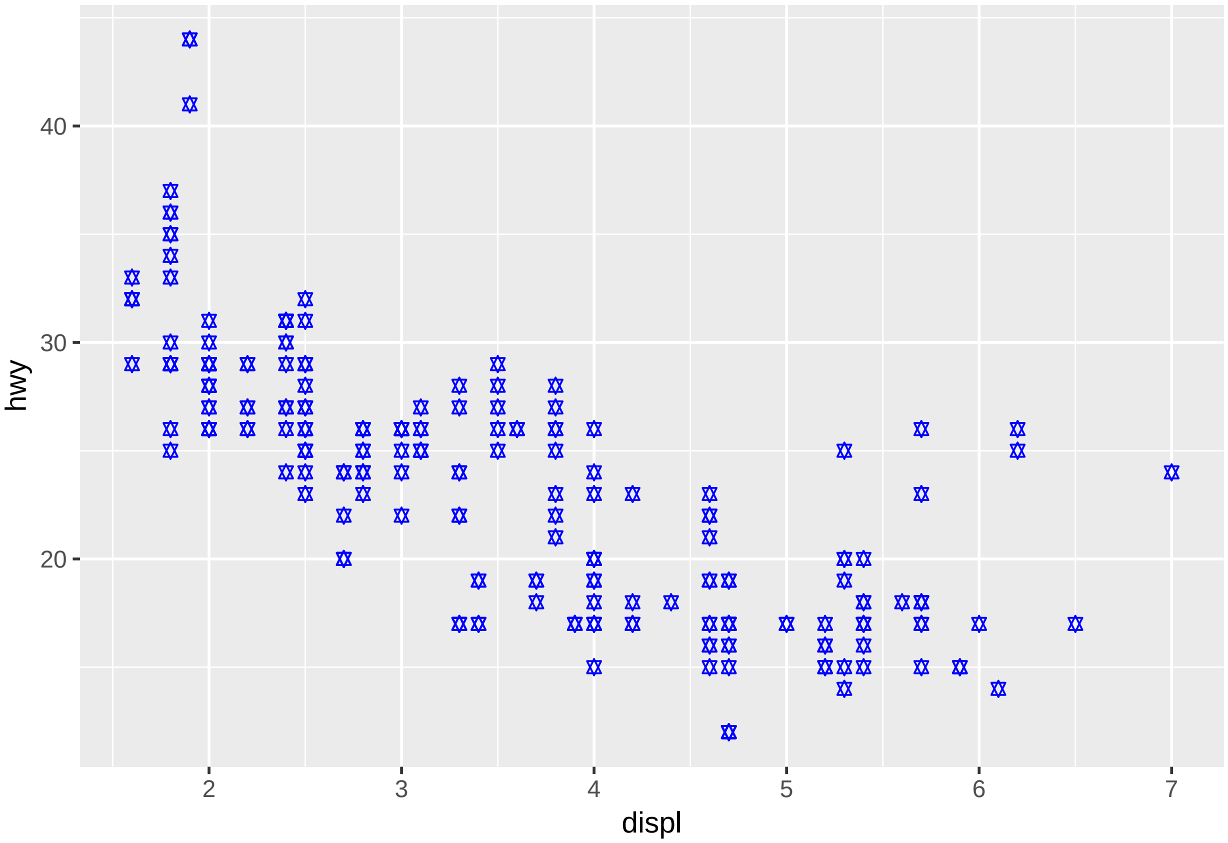
```
## Warning: The shape palette can deal with a maximum of 6 discrete values
## because more than 6 becomes difficult to discriminate; you have 7.
## Consider specifying shapes manually if you must have them.

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



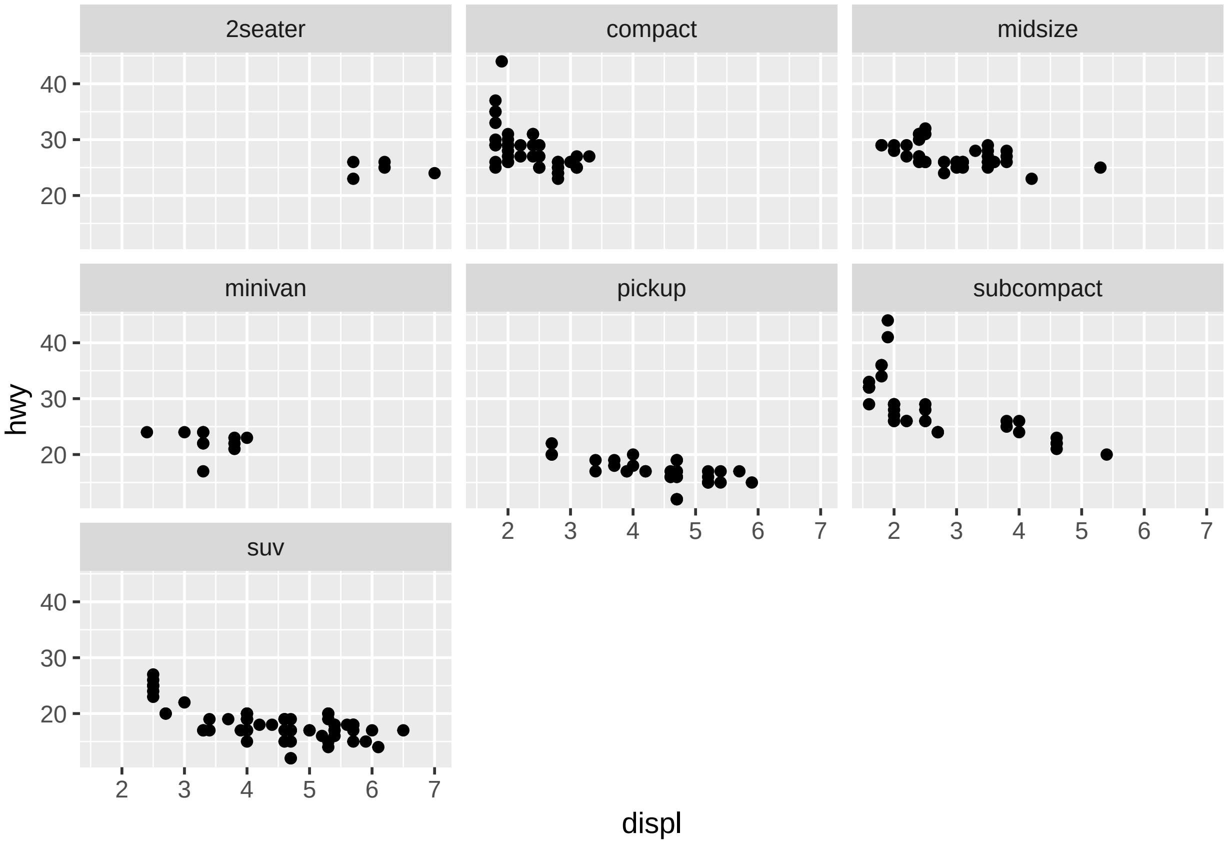
```
#What happened to the SUVs? ggplot2 will only use six shapes at a time.
#By default, additional groups will go unplotted when you use this aesthetic.

########################Unique color #####################################
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy), color = "blue")
```



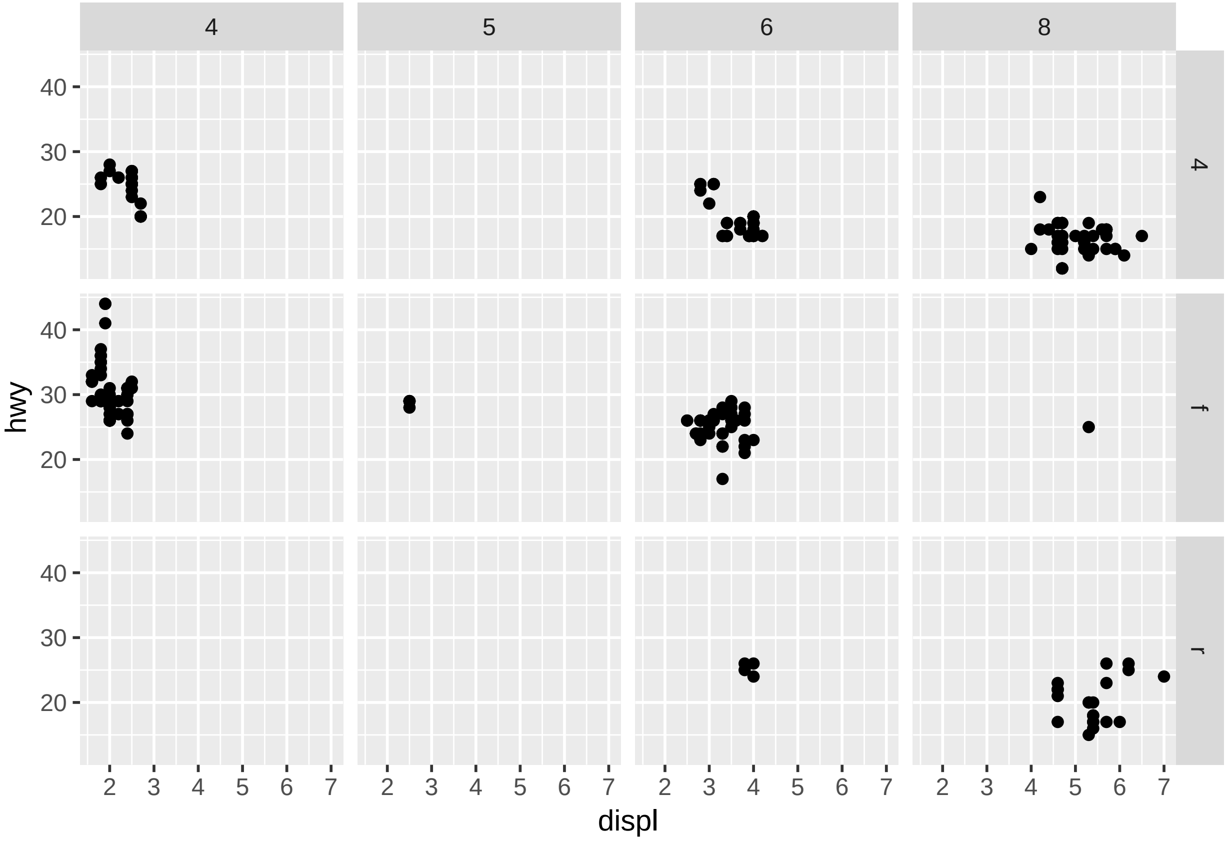
```
###########################Shape 11 and color blue #####################################
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy), color = "blue",shape=11)
```



```
########################### FACETS ######################################
# On class
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  facet_wrap(~ class, nrow = 3)
```

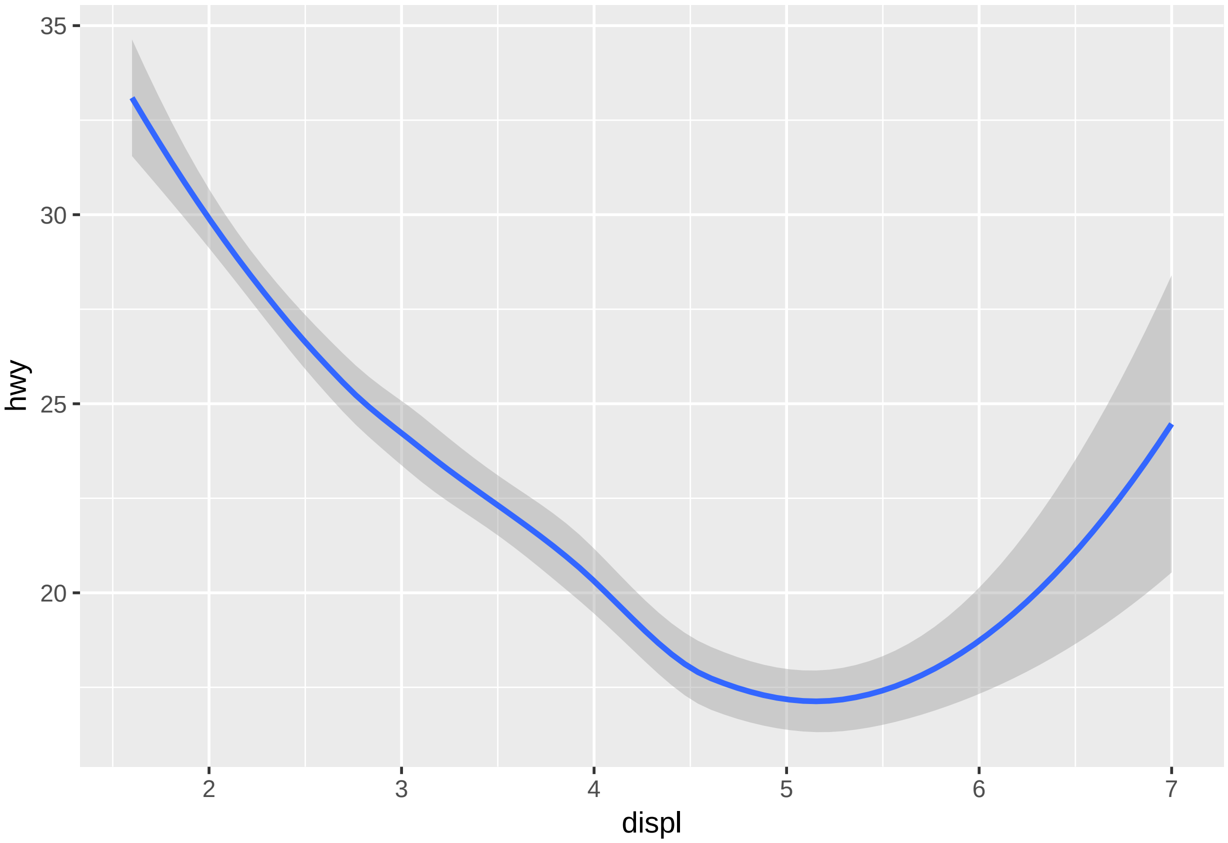


```
                    ####################

#On drv vs cyl
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  facet_grid(drv ~ cyl)
```



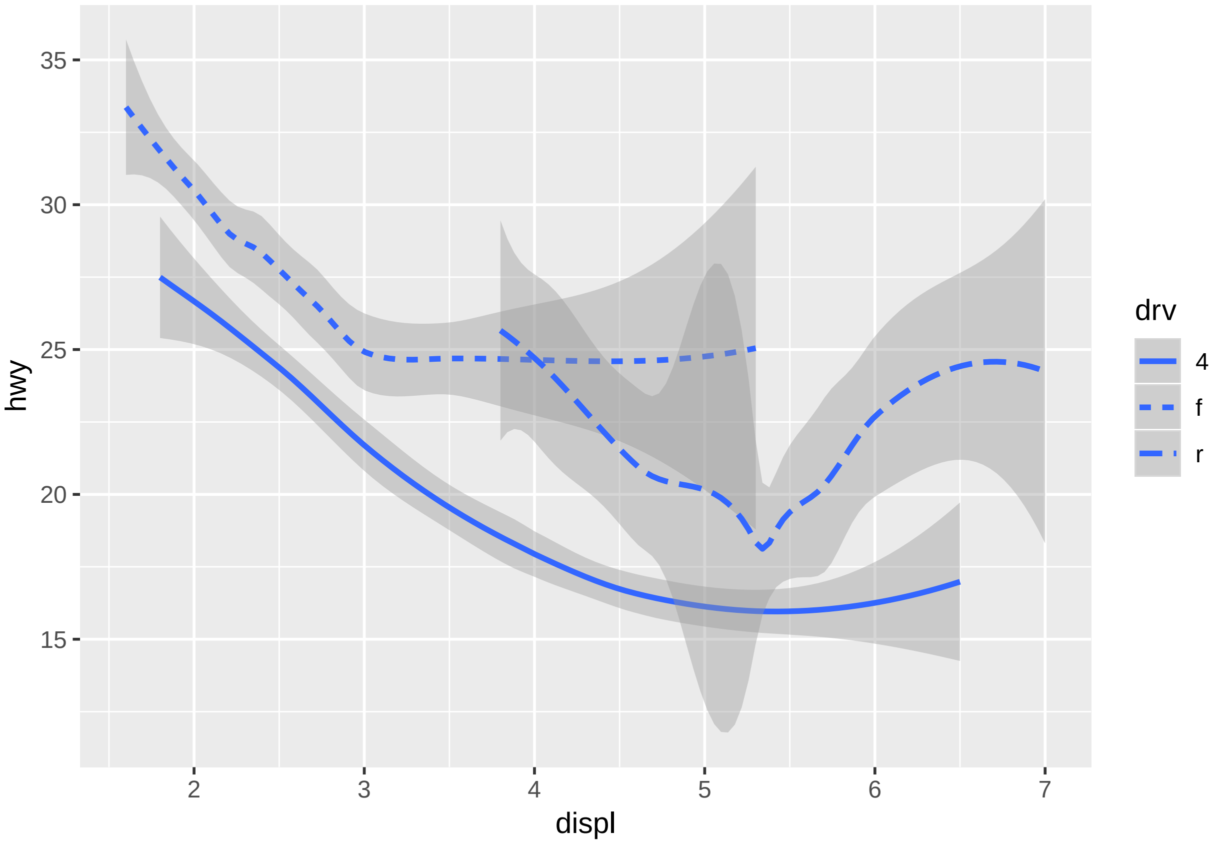
```
################################## lines ###################

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

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



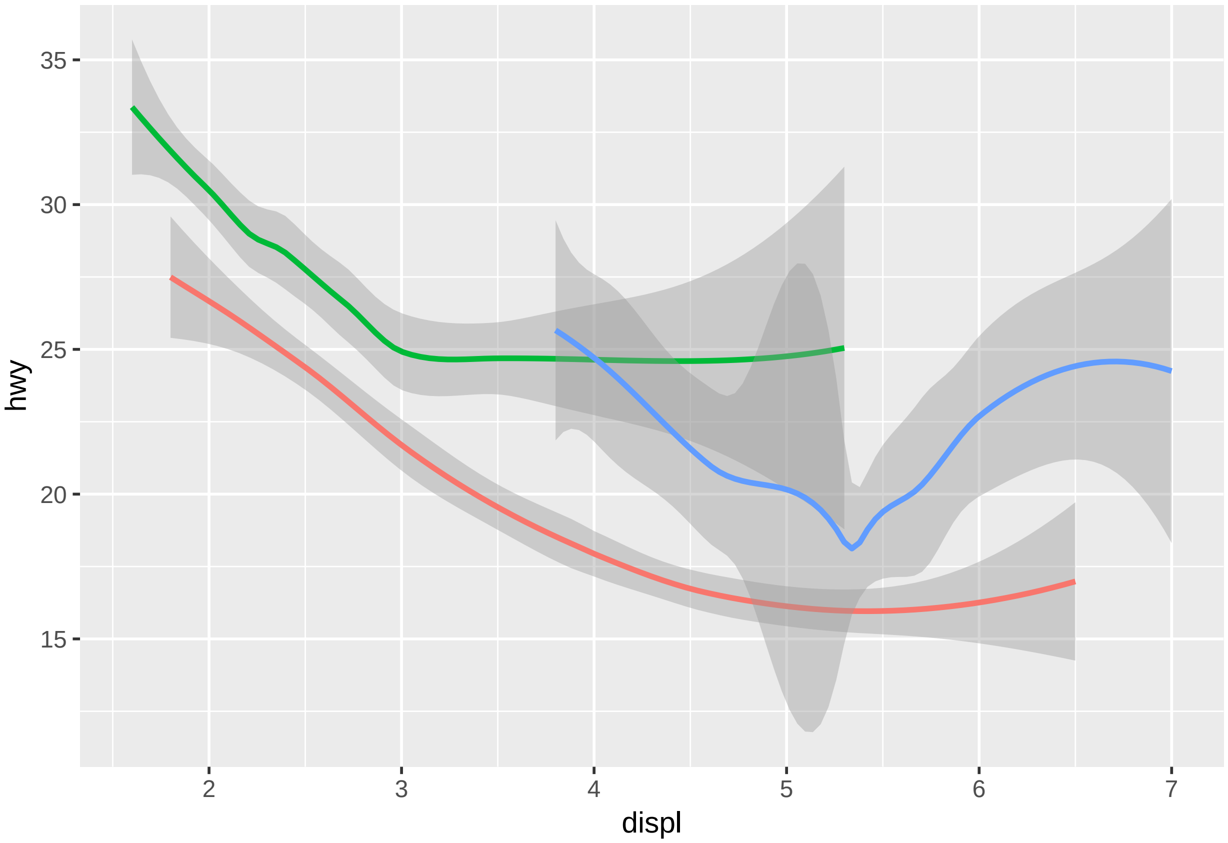
```r
ggplot(data = mpg) +
  geom_smooth(mapping = aes(x = displ, y = hwy, linetype = drv))
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



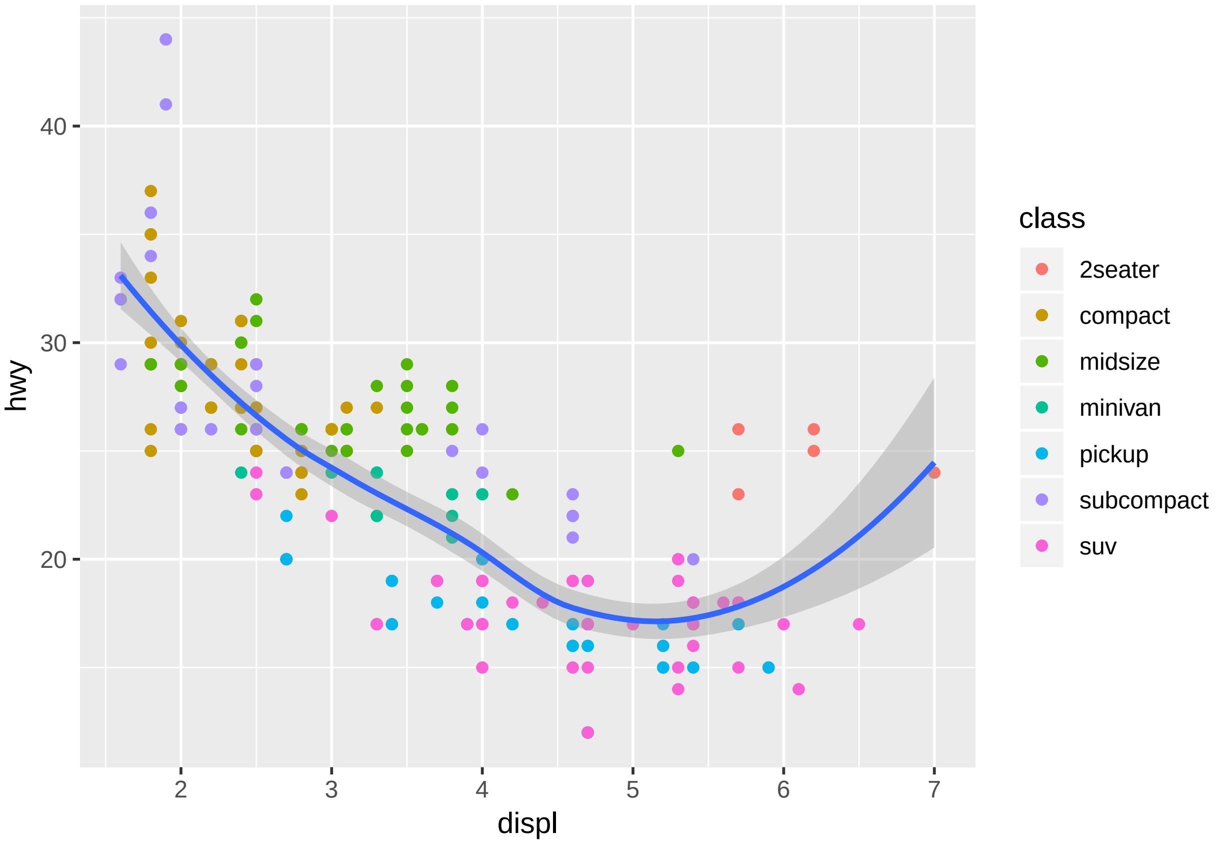
```
##########################################################################

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

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



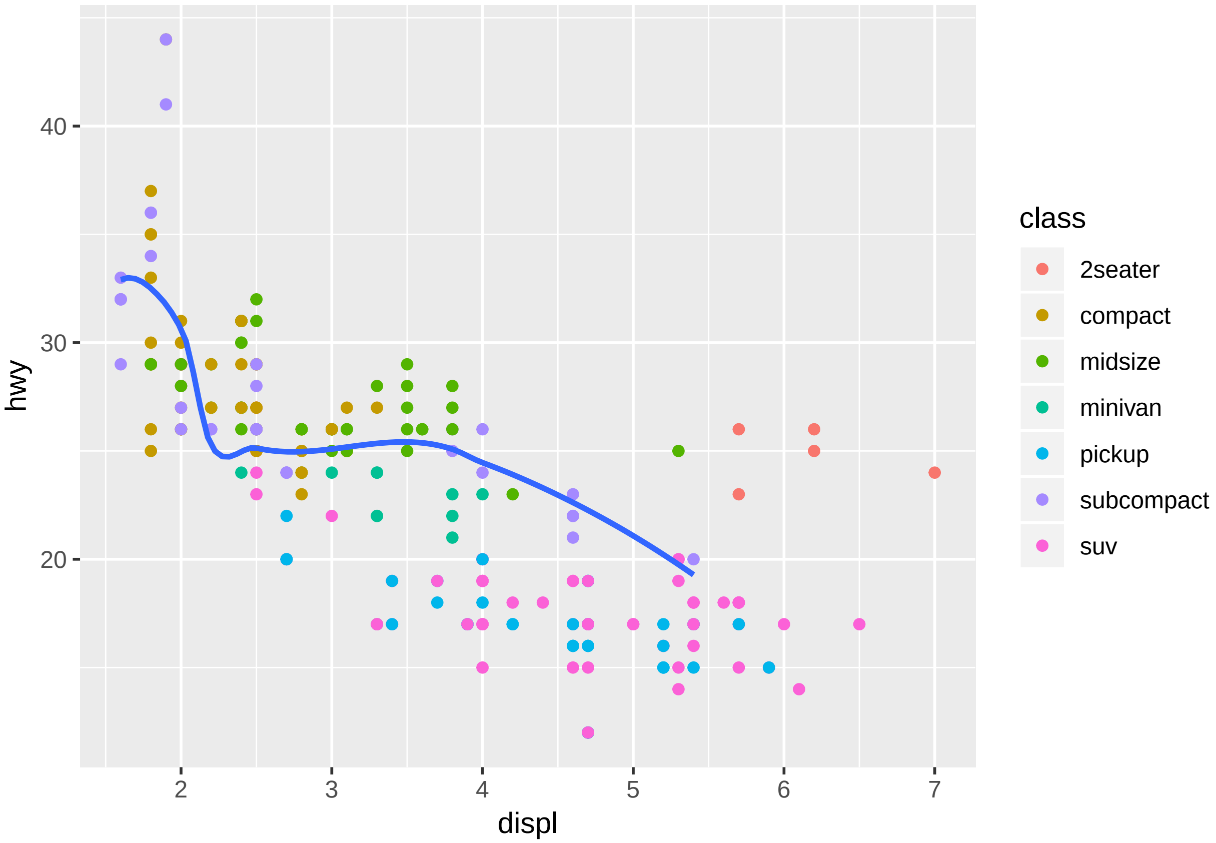
```
############################TWO OR MORE GEOM ON A GRAPH ###################
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point(mapping = aes(color = class)) +
  geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

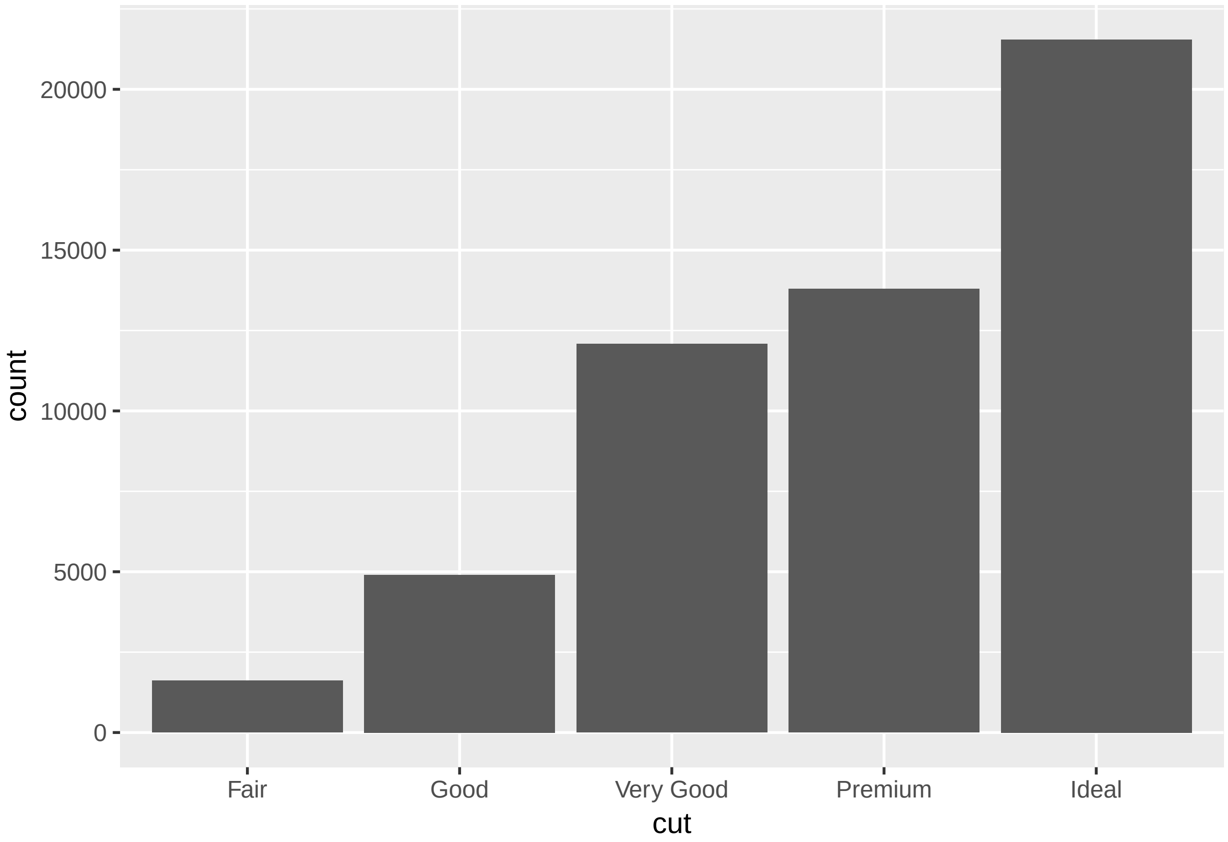


```
                        #######################

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

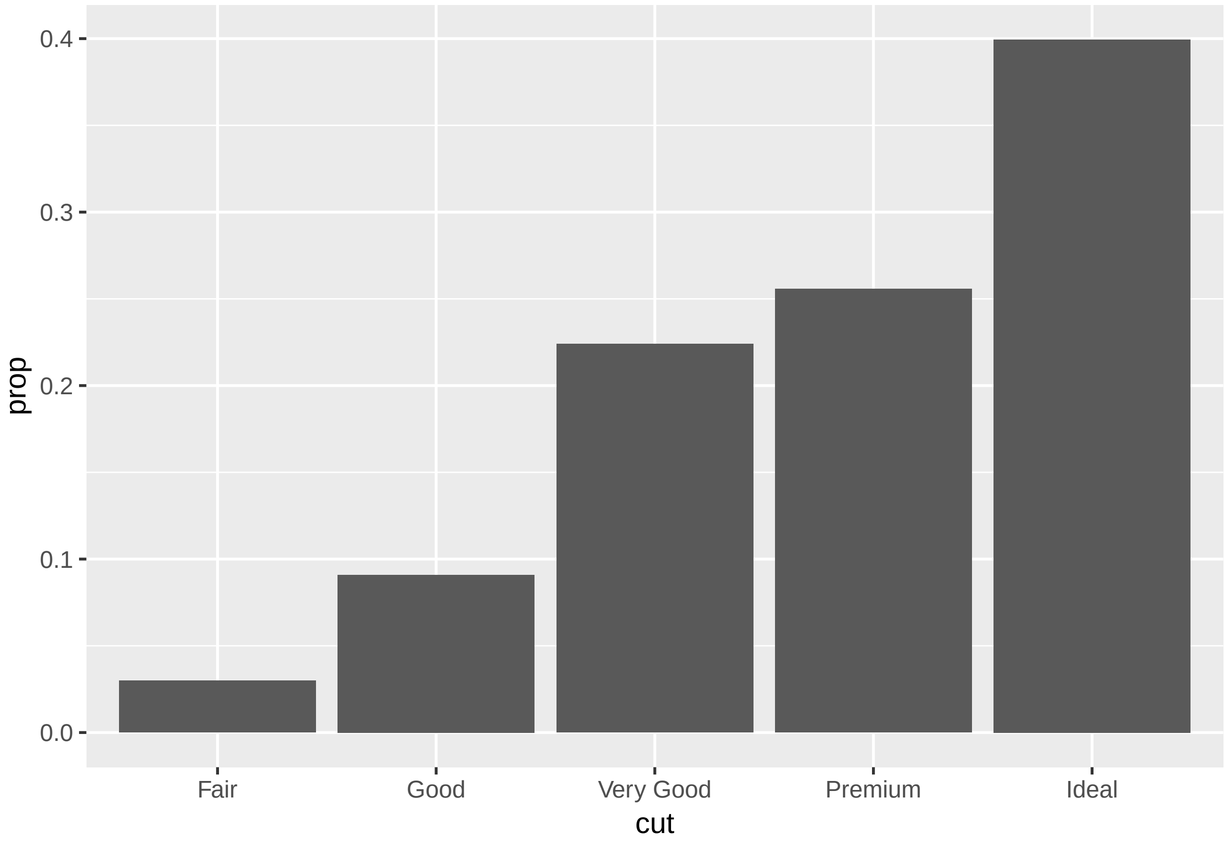
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



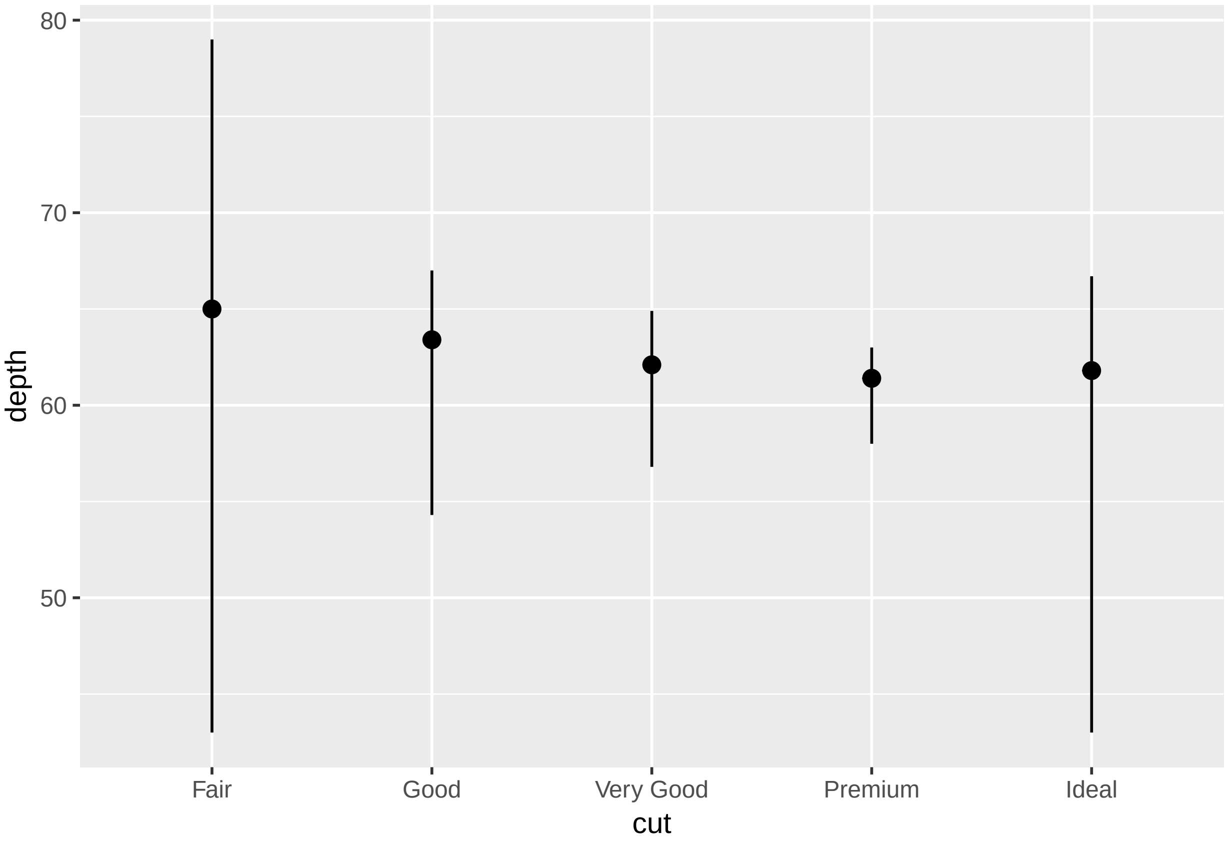
```
###################################STATISTICAL TRANSFORMATION###############

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



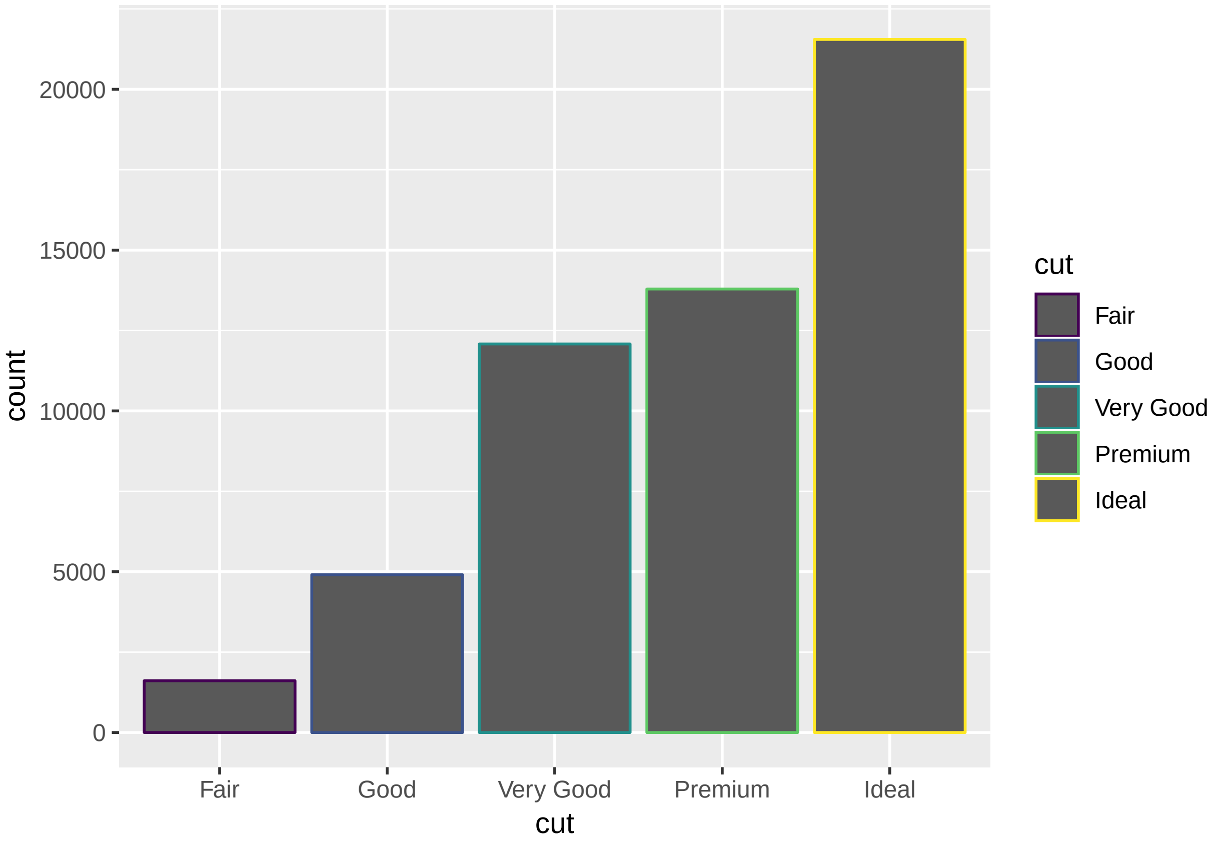
```
    #######################By proportion ############################

ggplot(data = diamonds) +
  geom_bar(
    mapping = aes(x = cut, y = ..prop.., group = 1)
  )
```



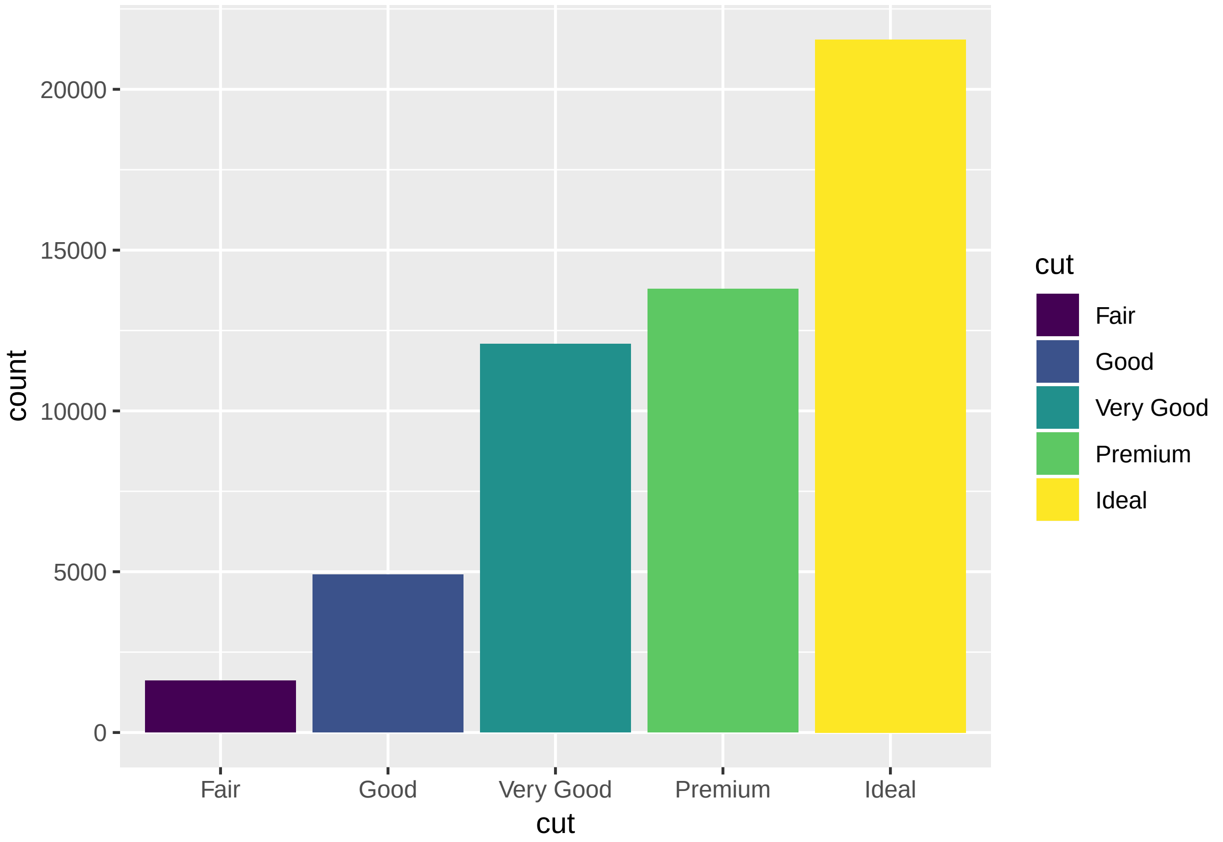
```
########################By Summary ##########################

ggplot(data = diamonds) +
  stat_summary(
    mapping = aes(x = cut, y = depth),
    fun.ymin = min,
    fun.ymax = max,
    fun.y = median
  )
```



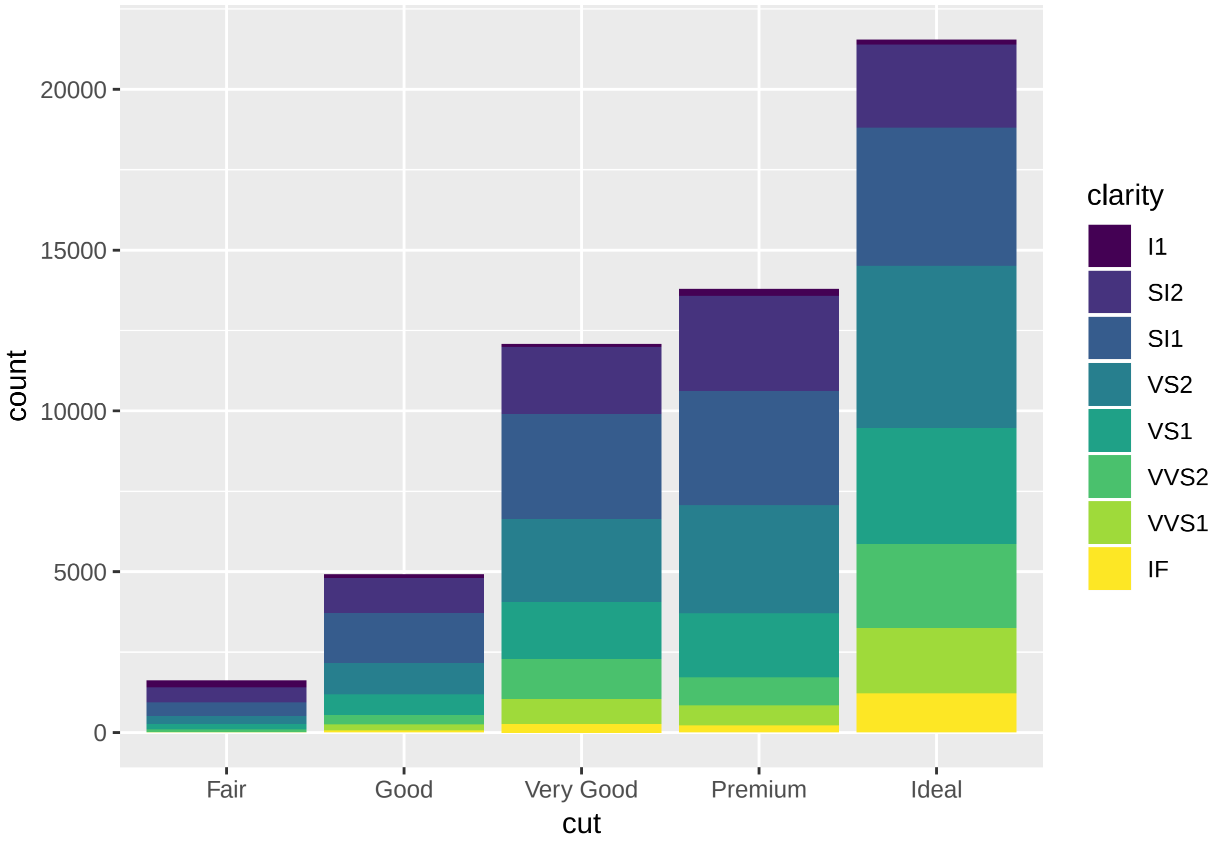
```
##########################POSITION ADJUSTMENT#######################
ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut, color = cut))
```



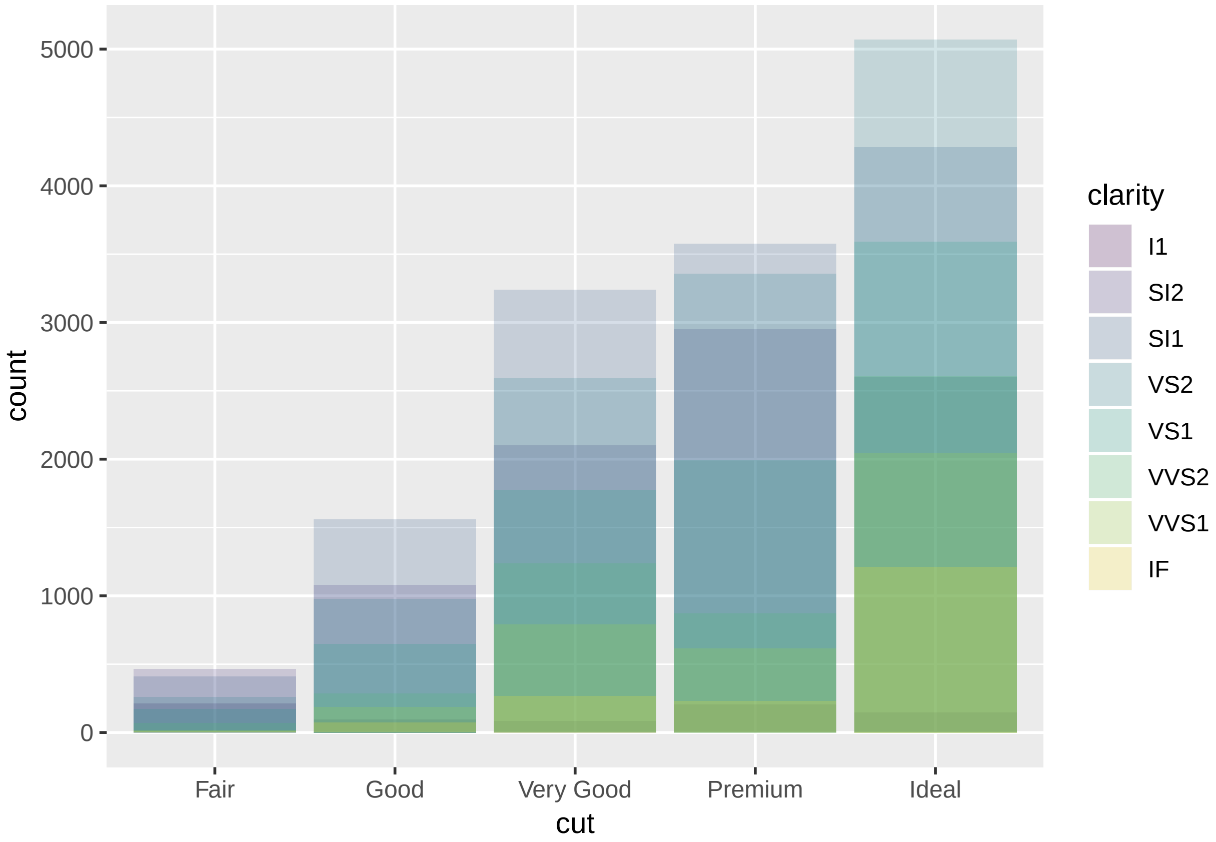
```
ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut, fill = cut))
```



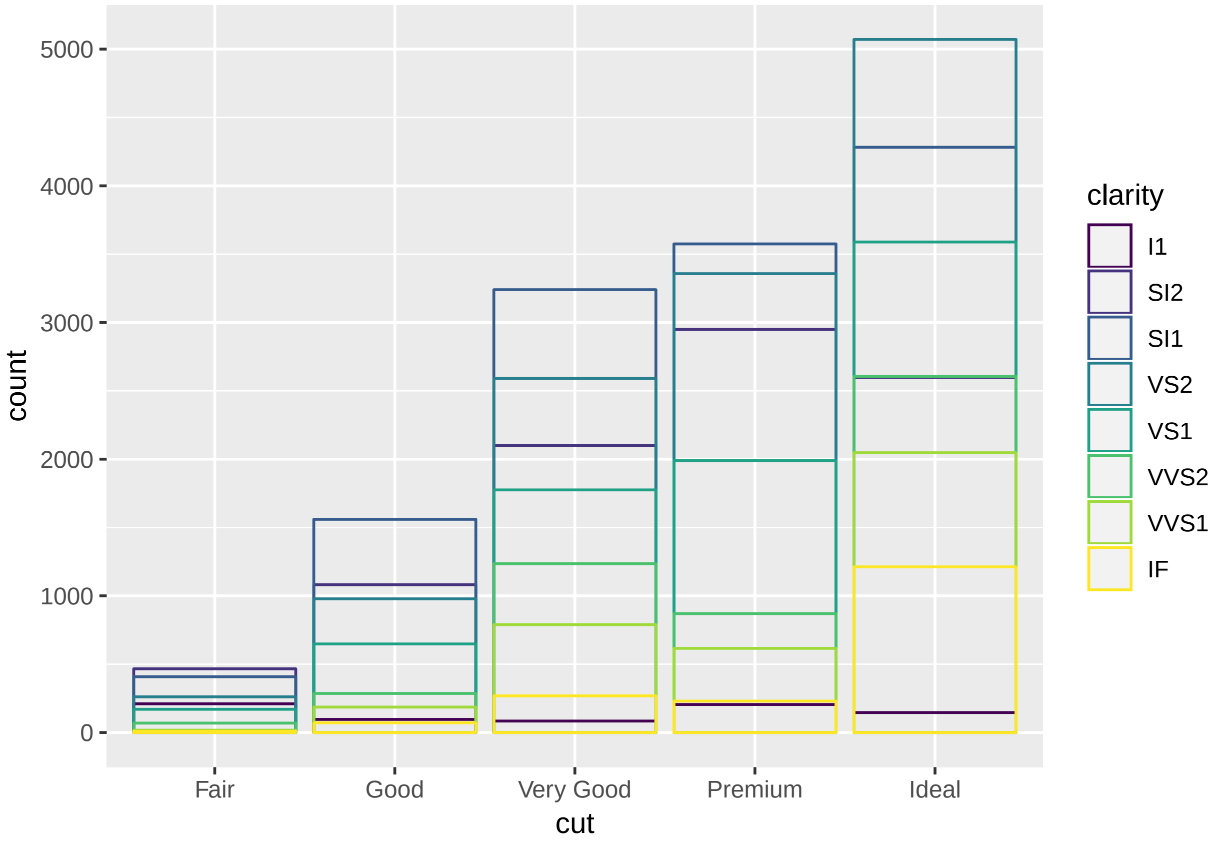
```
              ##########################

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



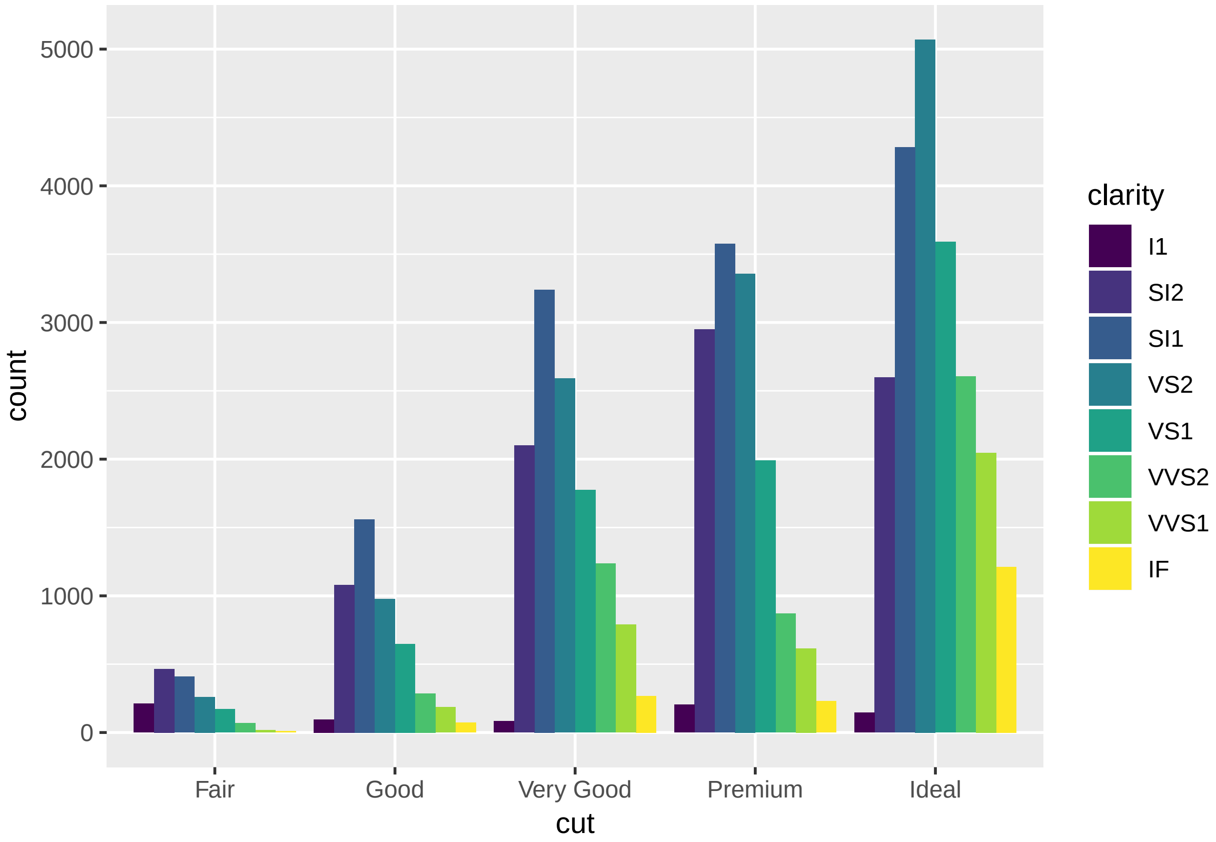
```
            #######################

ggplot(
  data = diamonds,
  mapping = aes(x = cut, fill = clarity)
)+
  geom_bar(alpha = 1/5, position = "identity")
```



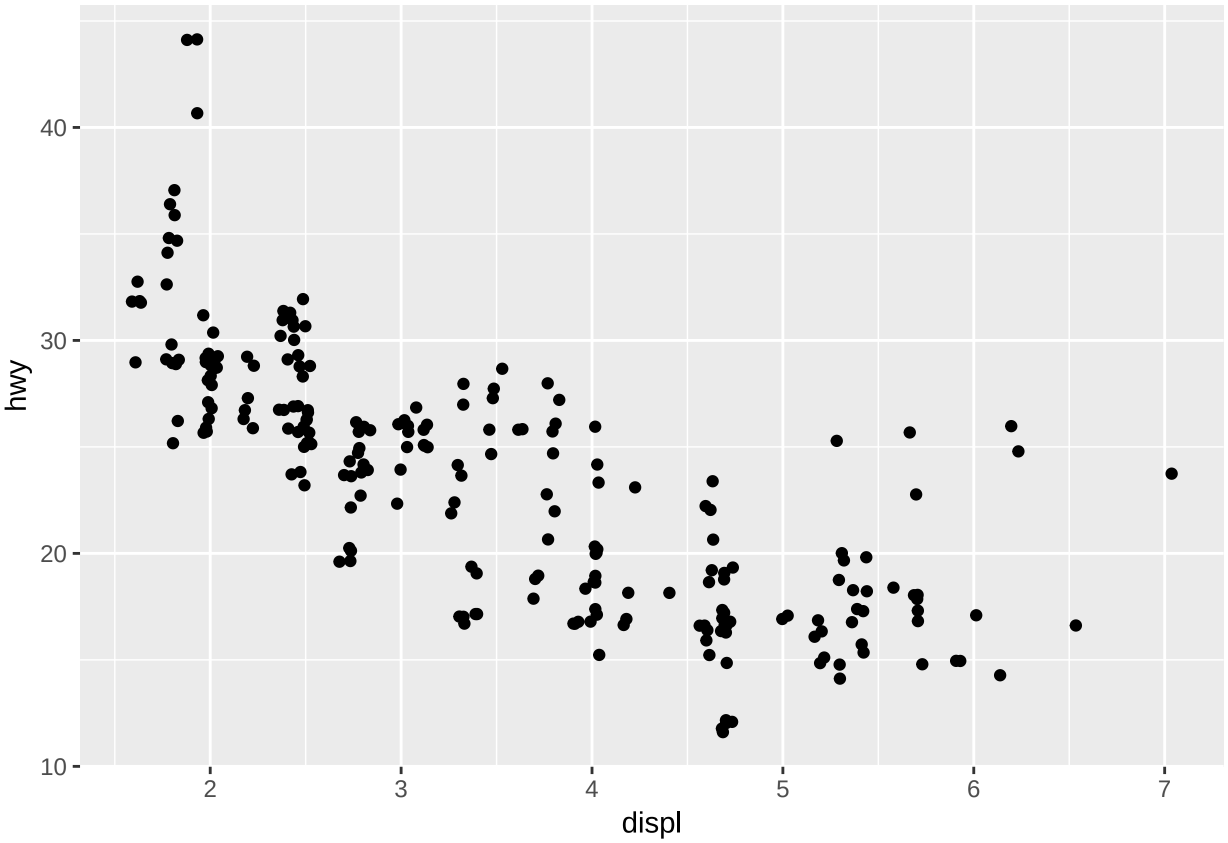
```r
ggplot(
  data = diamonds,
  mapping = aes(x = cut, color = clarity)
)+
  geom_bar(fill = NA, position = "identity")
```



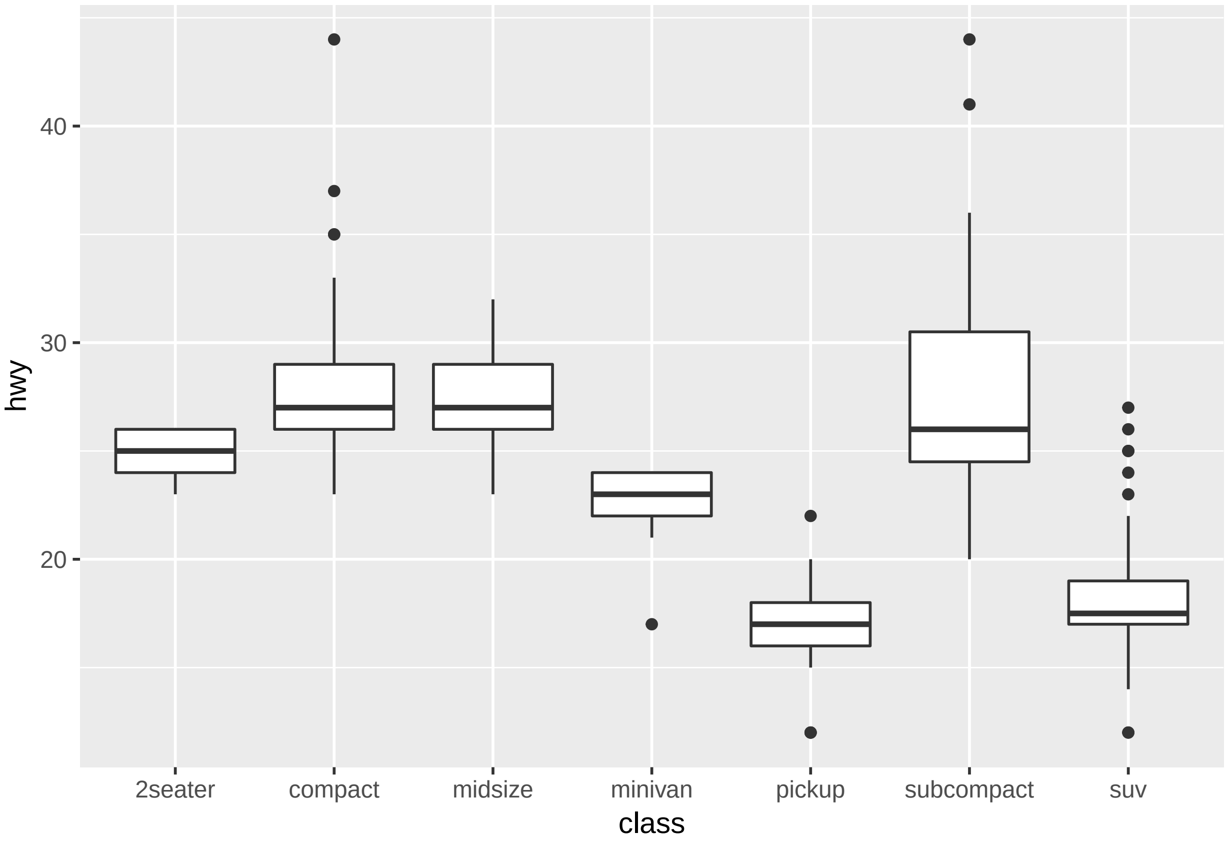
```
#position = "dodge" places overlapping objects directly beside one another.
#This makes it easier to compare individual values:

  ggplot(data = diamonds) +
 geom_bar(
   mapping = aes(x = cut, fill = clarity),
   position = "dodge"
 )
```



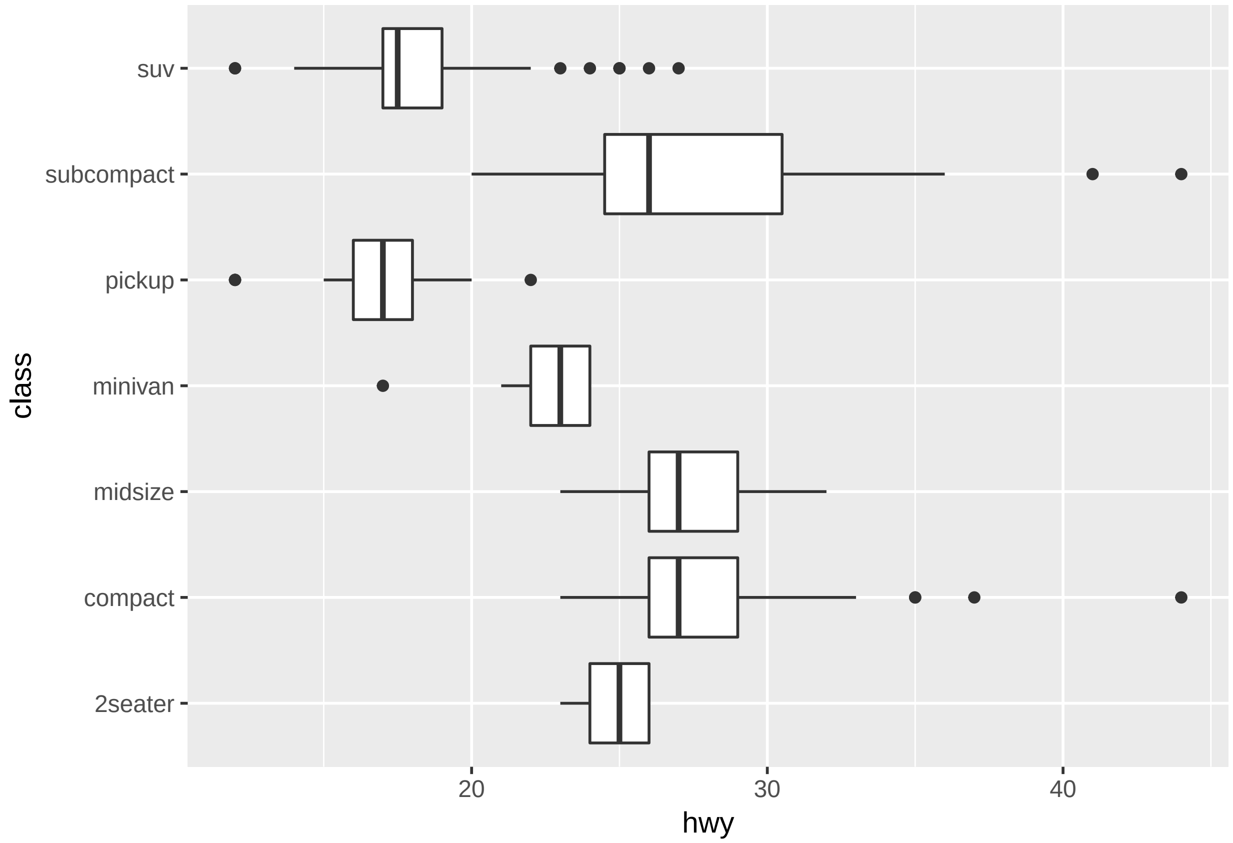
```
               #######  Jitters to differenciate point

#You can avoid this gridding by setting the position adjustment to "jitter." position = "jitter"
   #adds a small amount of random noise to each point. This spreads the points out because no two point
     ggplot(data = mpg) +
     geom_point(
       mapping = aes(x = displ, y = hwy),
       position = "jitter"
     )
```



```
###############################COORDINATES SYSTEM#######################################

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



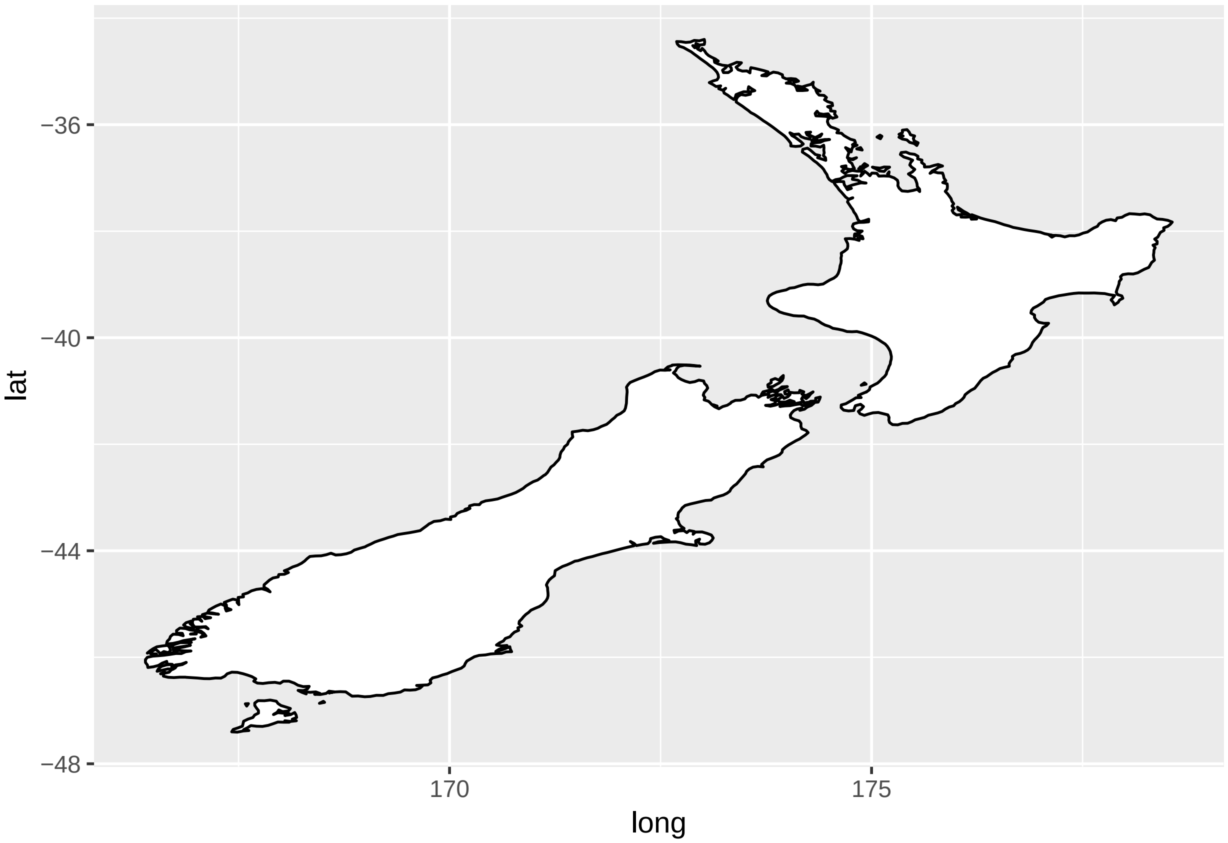
```
ggplot(data = mpg, mapping = aes(x = class, y = hwy)) +
  geom_boxplot() +
  coord_flip()
```



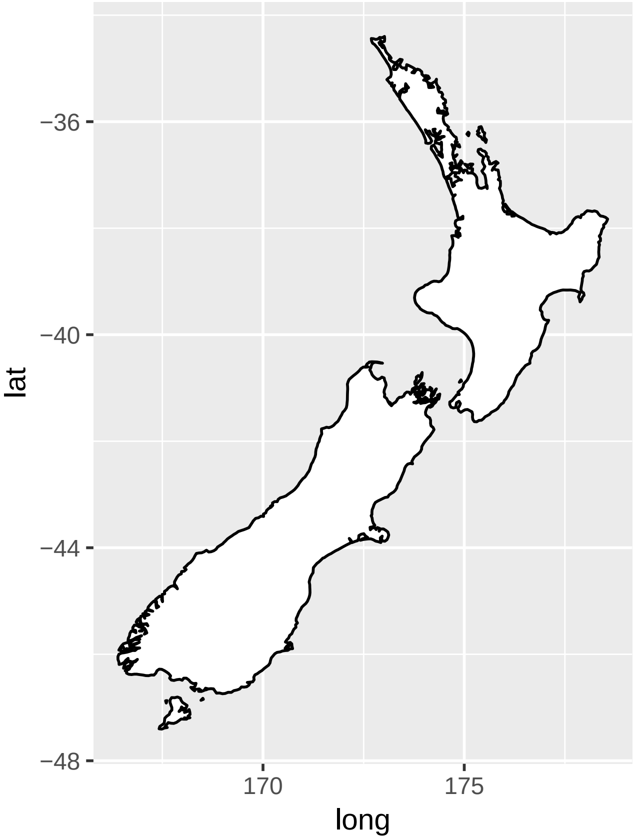
```
     ############################### COORDINATES

#coord_quickmap() sets the aspect ratio correctly for maps.
#This is very important if you're plotting spatial data with ggplot2
    #(which unfortunately we don't have the space to cover in this book):
       nz <- map_data("nz")
```

```
##
## Attaching package: 'maps'

## The following object is masked from 'package:purrr':
##
##     map
```

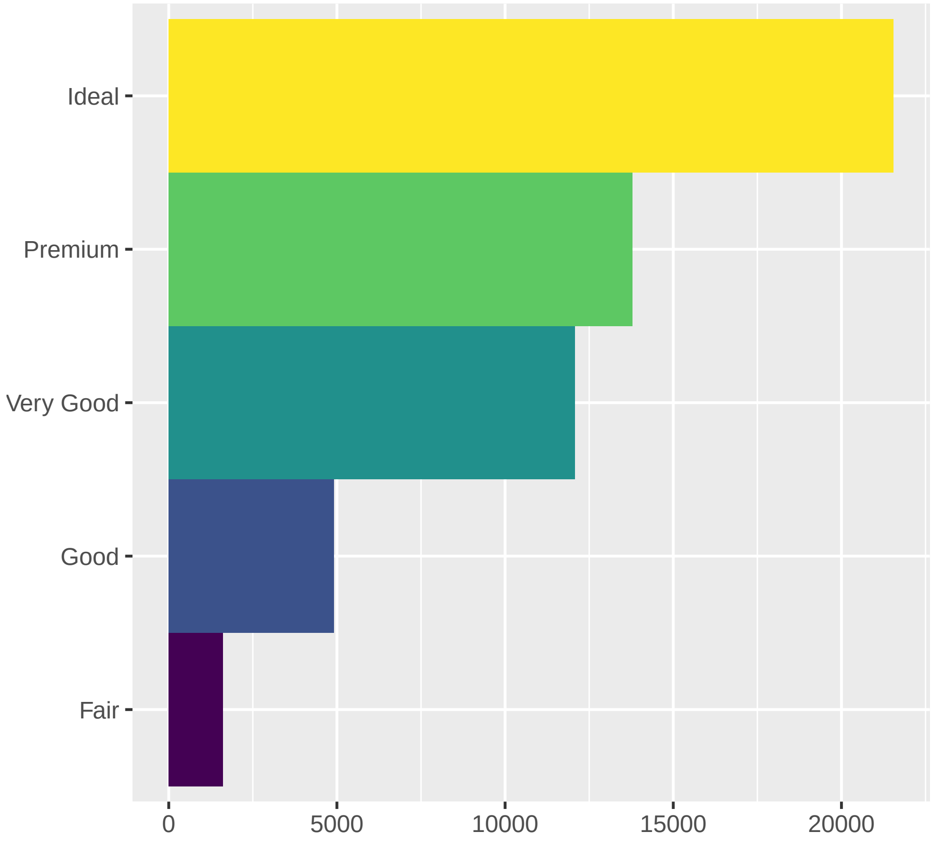
```
     ggplot(nz, aes(long, lat, group = group)) +
       geom_polygon(fill = "white", color = "black")
```



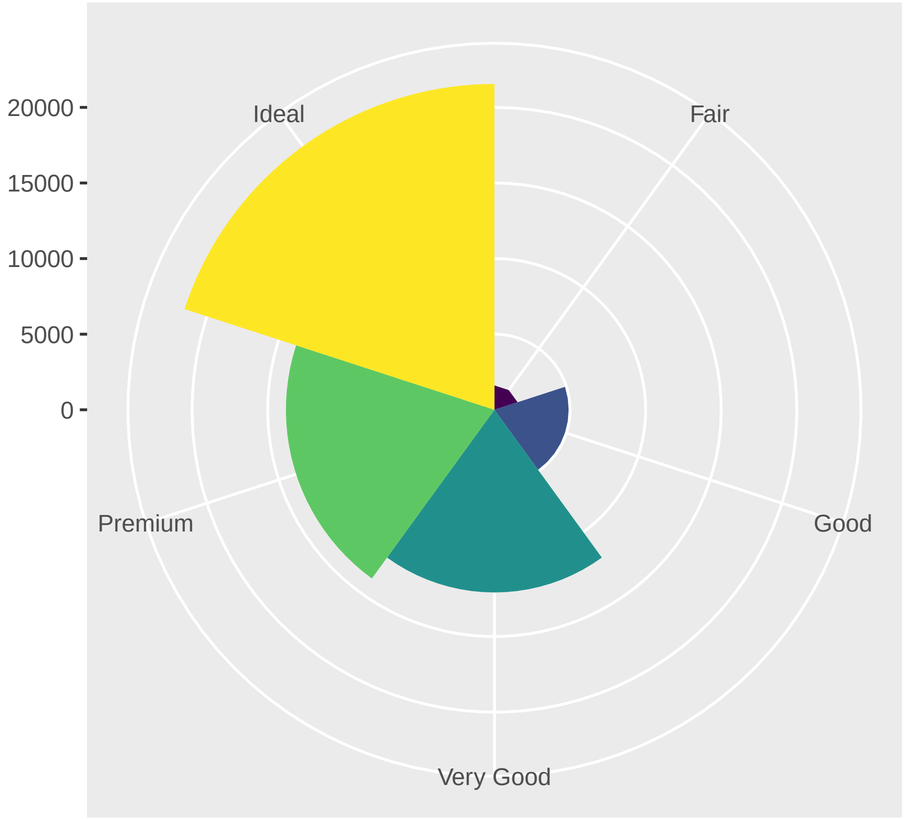
```
ggplot(nz, aes(long, lat, group = group)) +
  geom_polygon(fill = "white", color = "black") +
  coord_quickmap()
```



```
#coord_polar() uses polar coordinates. Polar coordinates reveal an interesting connection
  #between a bar chart and a Coxcomb chart:
       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()
```


Ideal
Fair
Premium
Good
Very Good