

## CM515 Day 1: Plotting with ggplot()

**Goal:** Utilize tidy data to generate complex graphs with few lines of code

**Start by loading the data**

```
mpg <- mpg
```

**Check out the data**

```
# What are 5 functions we could use to explore the mpg dataset?
```

```
str(mpg)
```

```
## tibble [234 x 11] (S3: tbl_df/tbl/data.frame)
## $ manufacturer: chr [1:234] "audi" "audi" "audi" "audi" ...
## $ model       : chr [1:234] "a4" "a4" "a4" "a4" ...
## $ displ       : num [1:234] 1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...
## $ year        : int [1:234] 1999 1999 2008 2008 1999 1999 2008 1999 1999 2008 ...
## $ cyl         : int [1:234] 4 4 4 4 6 6 6 4 4 4 ...
## $ trans       : chr [1:234] "auto(l5)" "manual(m5)" "manual(m6)" "auto(av)" ...
## $ drv         : chr [1:234] "f" "f" "f" "f" ...
## $ cty         : int [1:234] 18 21 20 21 16 18 18 18 16 20 ...
## $ hwy         : int [1:234] 29 29 31 30 26 26 27 26 25 28 ...
## $ fl         : chr [1:234] "p" "p" "p" "p" ...
## $ class       : chr [1:234] "compact" "compact" "compact" "compact" ...
```

```
summary(mpg)
```

```
## manufacturer      model      displ      year
## Length:234      Length:234      Min.   :1.600      Min.   :1999
## Class :character  Class :character  1st Qu.:2.400      1st Qu.:1999
## Mode  :character  Mode  :character  Median :3.300      Median :2004
##                                     Mean   :3.472      Mean   :2004
##                                     3rd Qu.:4.600      3rd Qu.:2008
##                                     Max.    :7.000      Max.    :2008
##      cyl      trans      drv      cty
## Min.   :4.000      Length:234      Length:234      Min.   : 9.00
## 1st Qu.:4.000      Class :character  Class :character  1st Qu.:14.00
## Median :6.000      Mode  :character  Mode  :character  Median :17.00
## Mean   :5.889                                     Mean   :16.86
## 3rd Qu.:8.000                                     3rd Qu.:19.00
## Max.    :8.000                                     Max.    :35.00
##      hwy      fl      class
## Min.   :12.00      Length:234      Length:234
## 1st Qu.:18.00      Class :character  Class :character
## Median :24.00      Mode  :character  Mode  :character
## Mean   :23.44
## 3rd Qu.:27.00
## Max.    :44.00
```

```
colnames(mpg)
```

```
## [1] "manufacturer" "model"      "displ"      "year"      "cyl"
## [6] "trans"        "drv"        "cty"        "hwy"        "fl"
## [11] "class"
```

```
?mpg
```

```
head(mpg)
```

```
## # A tibble: 6 x 11
##   manufacturer model displ year   cyl trans      drv   cty   hwy fl  class
##   <chr>         <chr> <dbl> <int> <int> <chr>   <chr> <int> <int> <chr> <chr>
## 1 audi         a4      1.8  1999     4 auto(l5) f       18    29 p  compa~
## 2 audi         a4      1.8  1999     4 manual(m5) f       21    29 p  compa~
## 3 audi         a4      2    2008     4 manual(m6) f       20    31 p  compa~
## 4 audi         a4      2    2008     4 auto(av) f       21    30 p  compa~
## 5 audi         a4      2.8  1999     6 auto(l5) f       16    26 p  compa~
## 6 audi         a4      2.8  1999     6 manual(m5) f       18    26 p  compa~
```

```
# Which manufacturer has the most models in this dataset?
```

```
mpg %>%
  count(model) %>%
  arrange(n)
```

```
## # A tibble: 38 x 2
##   model          n
##   <chr>        <int>
## 1 land cruiser wagon 4wd     2
## 2 a6 quattro         3
## 3 expedition 2wd         3
## 4 maxima             3
## 5 navigator 2wd         3
## 6 k1500 tahoe 4wd        4
## 7 mountaineer 4wd        4
## 8 pathfinder 4wd        4
## 9 range rover         4
## 10 c1500 suburban 2wd      5
## # i 28 more rows
```

```
mpg %>%
  count(model) %>%
  arrange(desc(n))
```

```
## # A tibble: 38 x 2
##   model          n
##   <chr>        <int>
## 1 caravan 2wd         11
## 2 ram 1500 pickup 4wd   10
## 3 civic              9
## 4 dakota pickup 4wd     9
## 5 jetta              9
## 6 mustang            9
## 7 a4 quattro          8
## 8 grand cherokee 4wd    8
## 9 impreza awd         8
```

```
## 10 a4 7
## # i 28 more rows
```

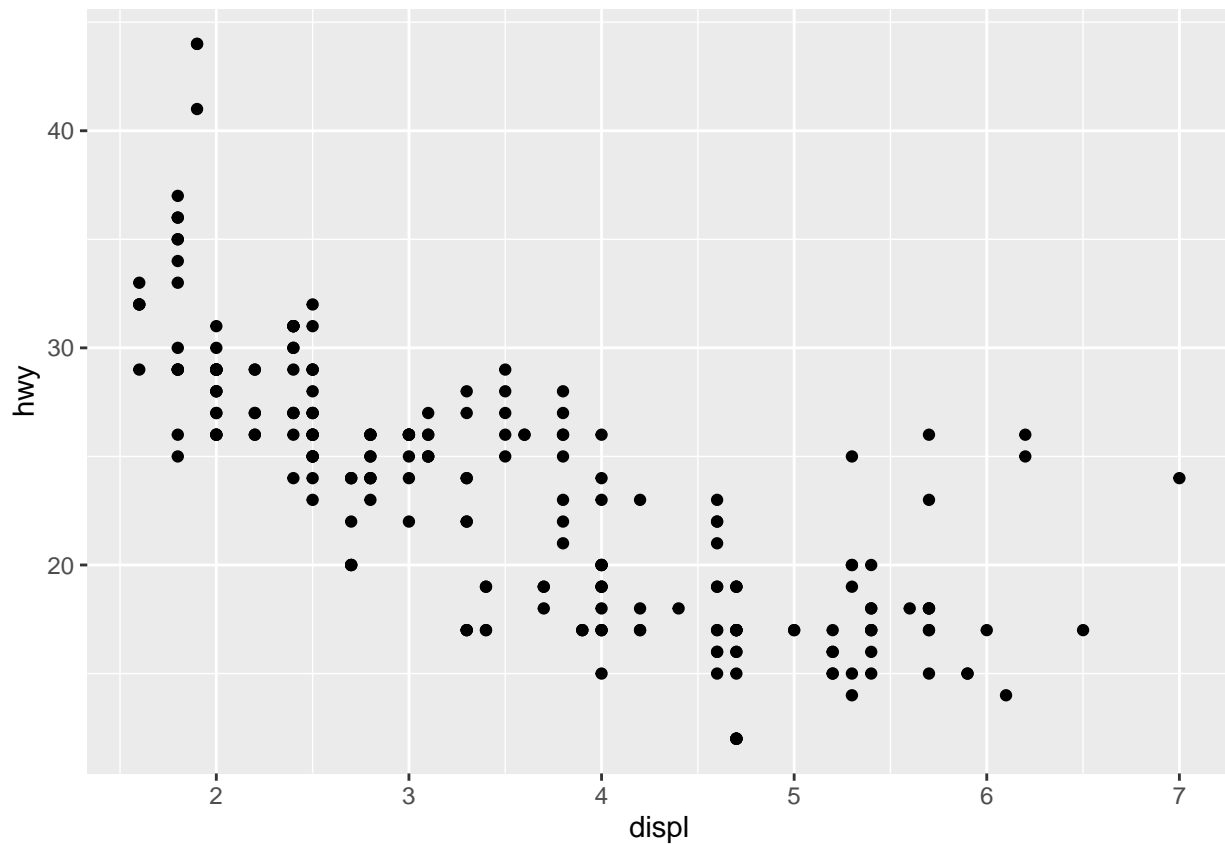
This dataset suggests many interesting questions. How are engine size and fuel economy related? Do certain manufacturers care more about fuel economy than others? Has fuel economy improved in the last ten years? We will try to answer some of these questions, and in the process learn how to create some basic plots with ggplot2.

**Every ggplot has three key components:**

- Data
- Aesthetic mappings between variables in the data
- A layer to render the information (geom)

**A simple example**

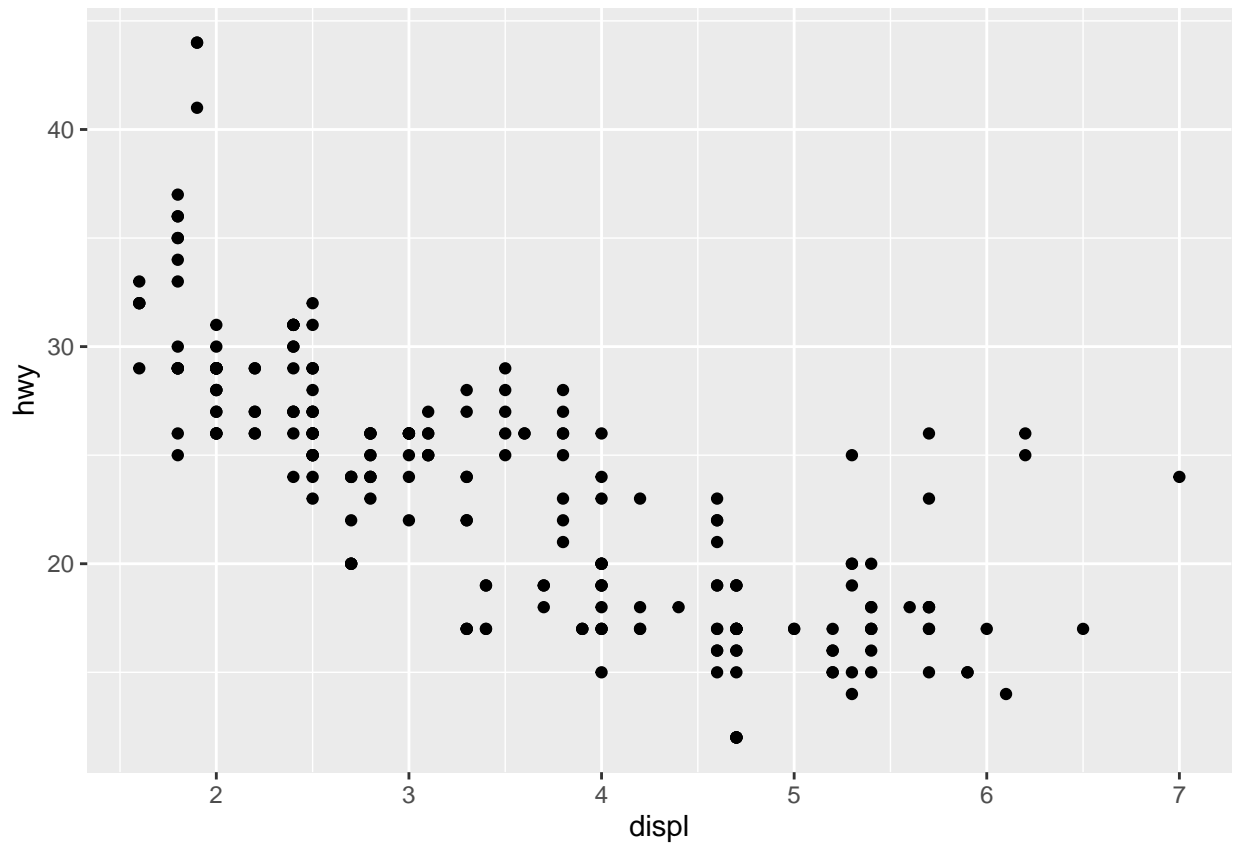
```
ggplot(mpg, aes(x = displ, y = hwy)) +  
  geom_point()
```



```
# Fill in the following:  
# Data: mpg  
# Aesthetic: engine oil displacement to fuel economy  
# Layers: point  
# Plus sign location: end of first line
```

x and y aesthetic can be implied

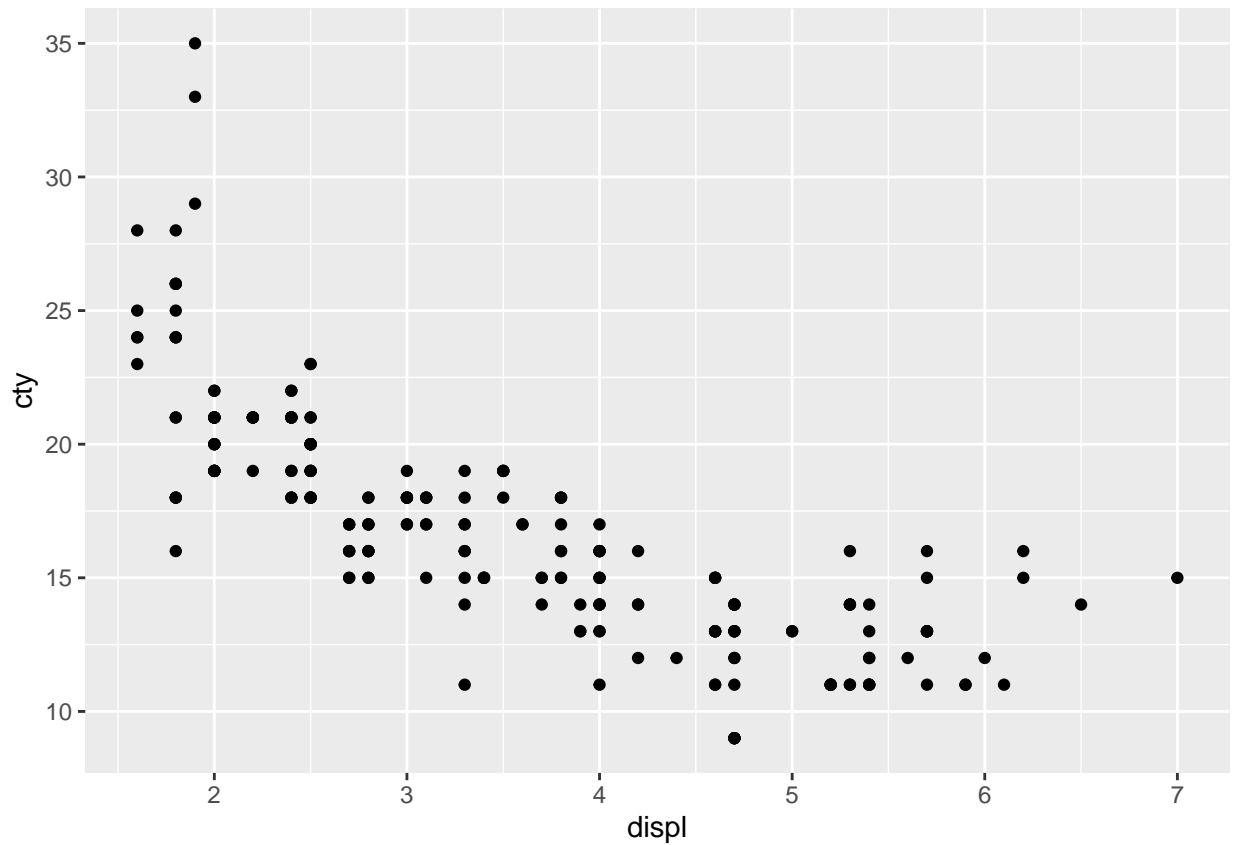
```
ggplot(mpg, aes(displ, hwy)) +  
  geom_point()
```



*# What conclusions can you make from this graph?*

*# Do these data look similar when comparing engine displacement to city miles per gallon?*

```
ggplot(mpg, aes(displ, cty)) +  
  geom_point()
```

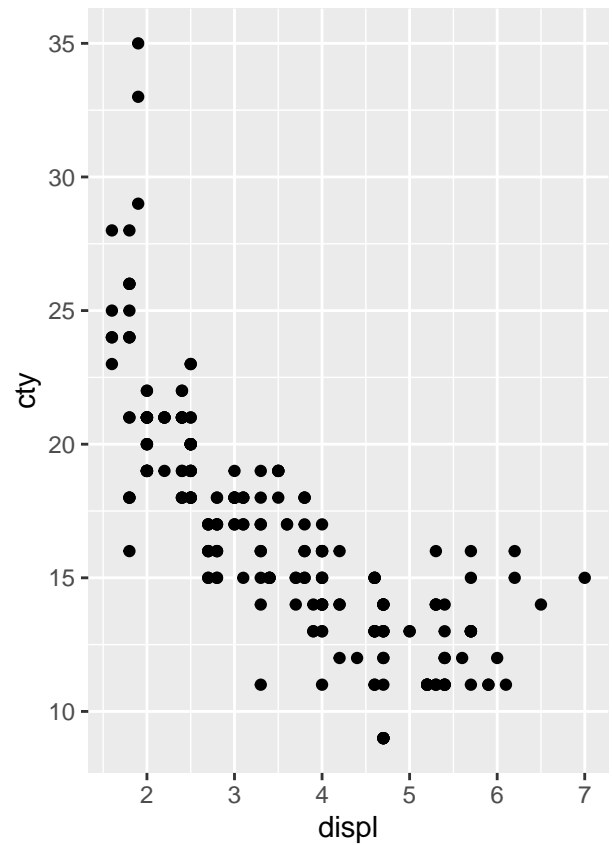
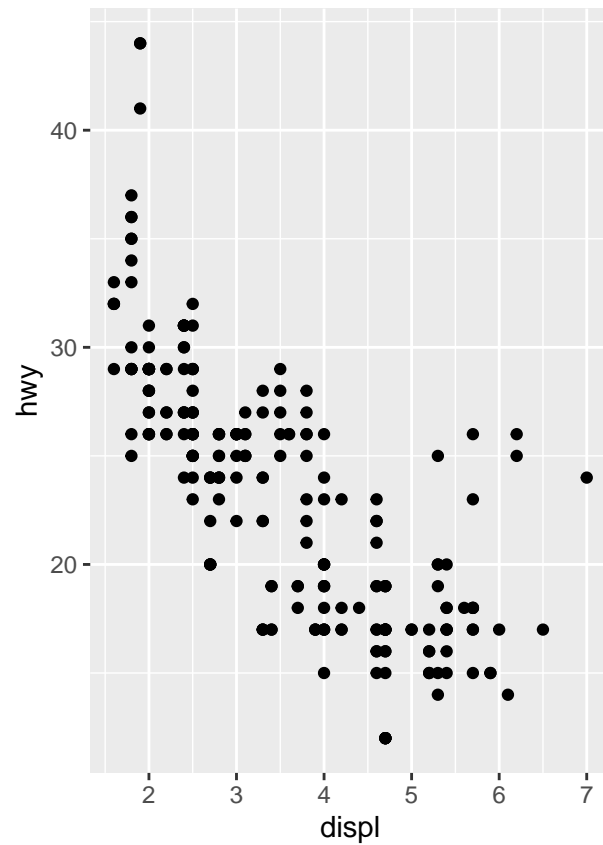


```
# We can more easily compare the highway and city miles per gallon using the gridExtra package.
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
## The following object is masked from 'package:dplyr':
##
##   combine
p1 <- ggplot(mpg, aes(displ, hwy)) +
  geom_point()

p2 <- ggplot(mpg, aes(displ, cty)) +
  geom_point()

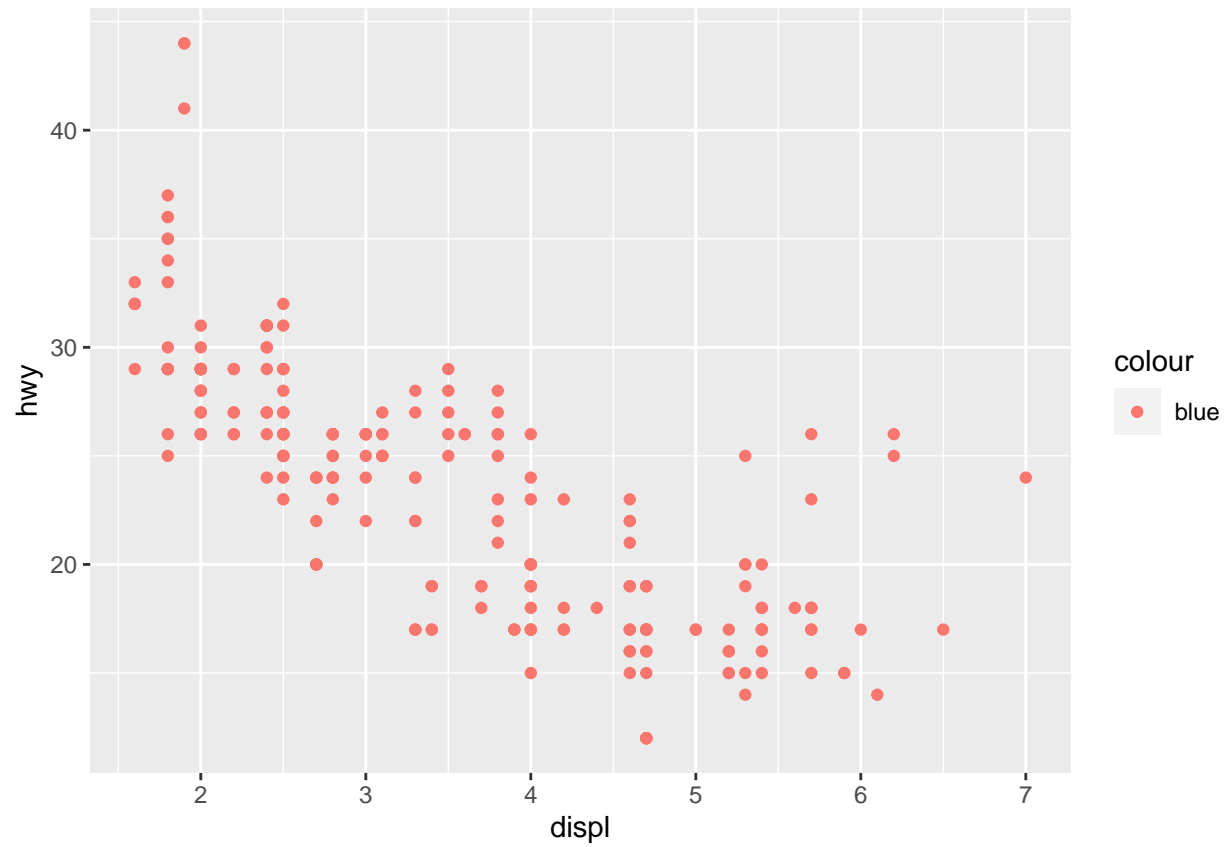
grid.arrange(p1, p2, ncol = 2)
```



## Aesthetic attributes

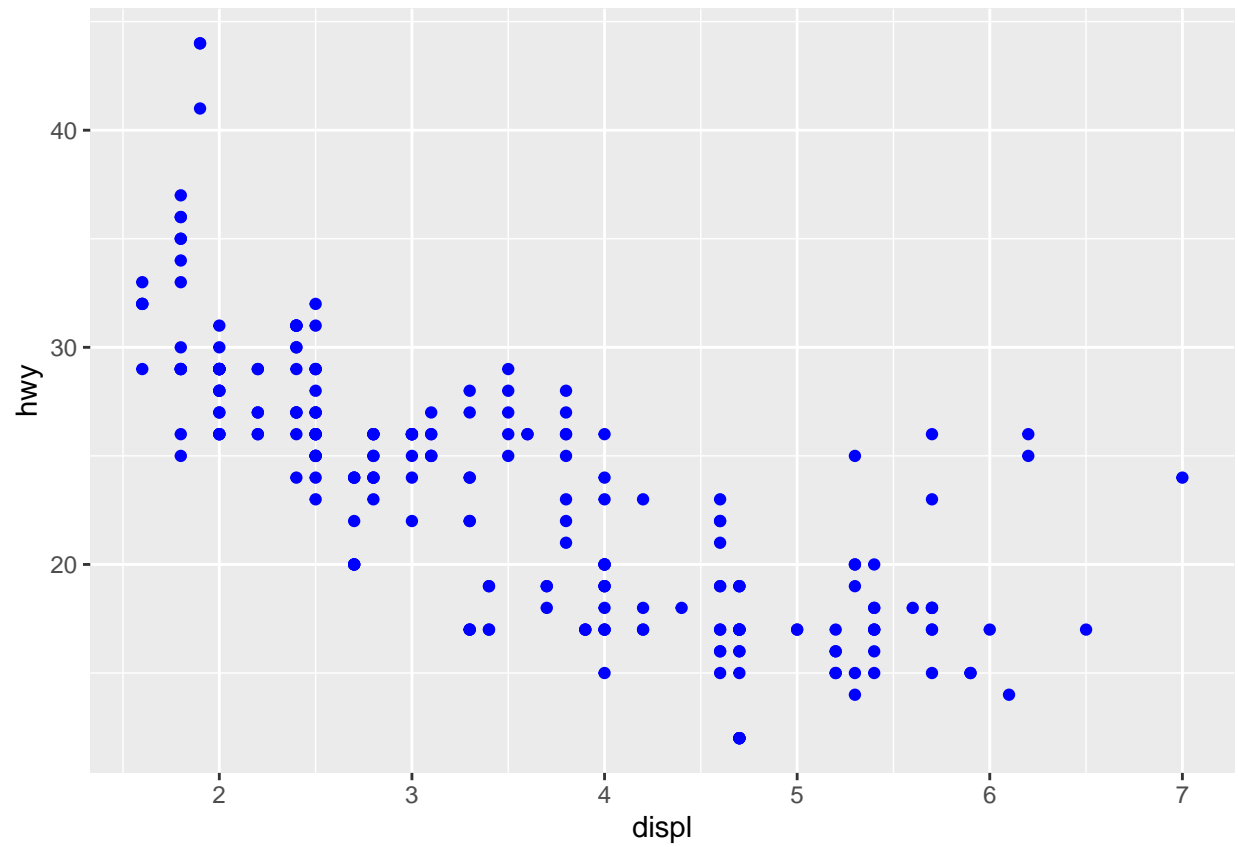
### Color

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



```
# Value "blue" is scaled to red color
```

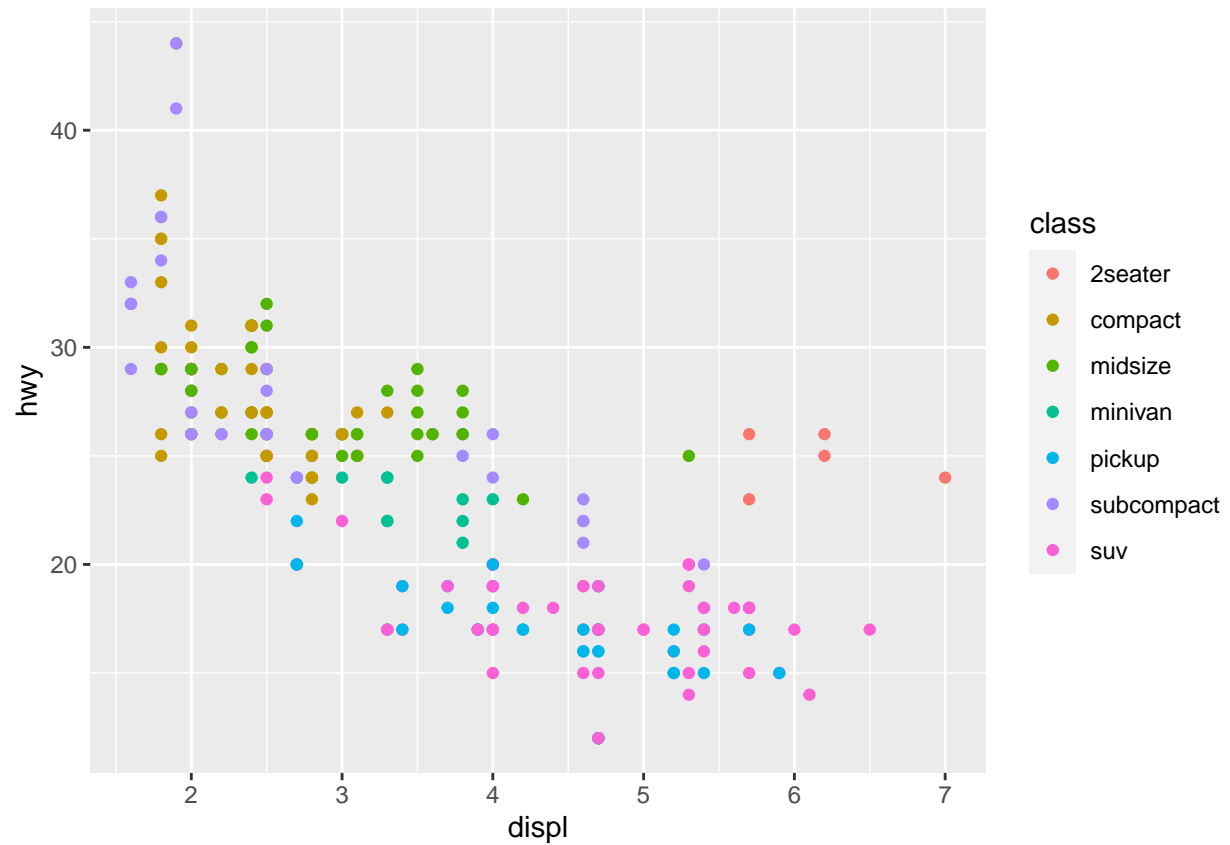
```
ggplot(mpg, aes(displ, hwy)) +  
  geom_point(color = "blue")
```



```
# Points are given color
```

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



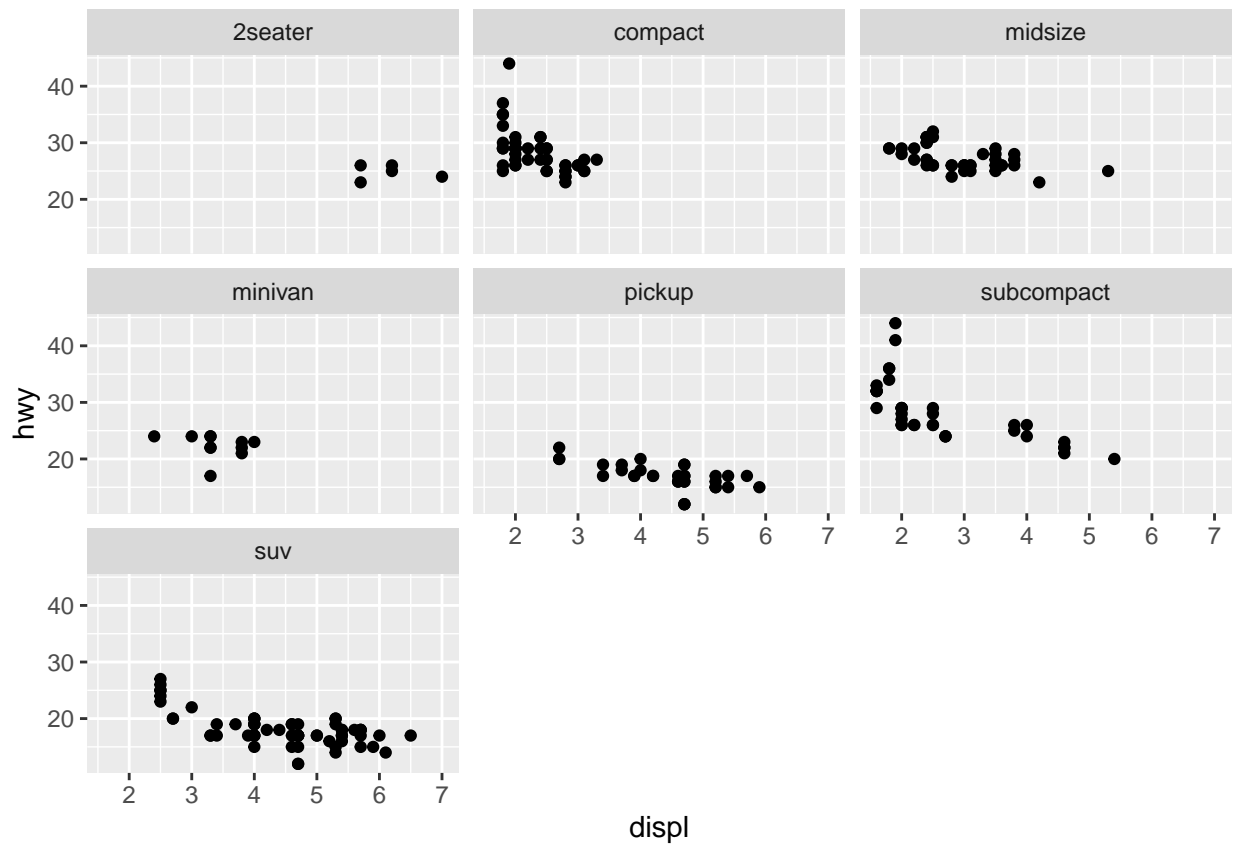


*# Unique color based on class of vehicle*

*# What additional information does the color add? What patterns do you see?*

### Facet wrap

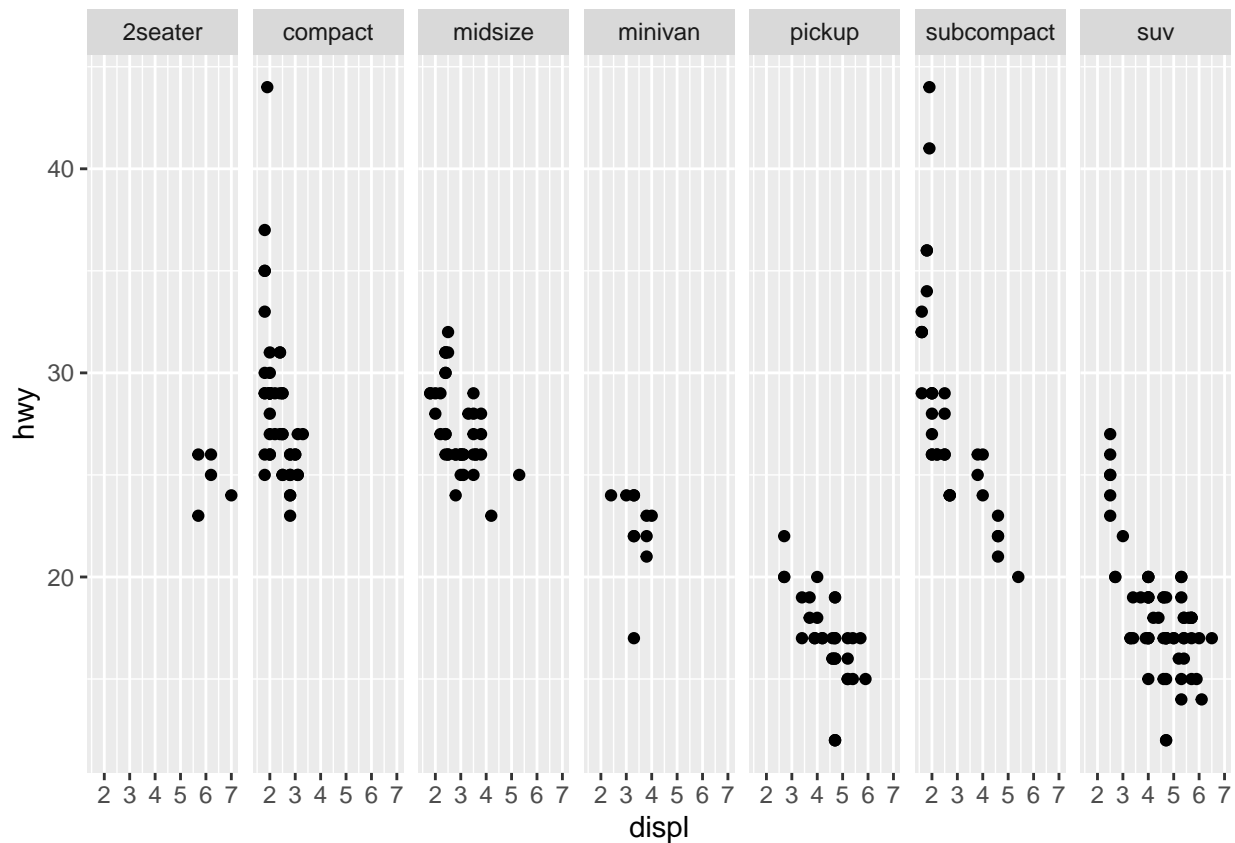
```
ggplot(mpg, aes(displ, hwy)) +
  geom_point() +
  facet_wrap(~ class)
```



*# Read the documentation for facet\_wrap. What arguments can you use to control how many rows and columns*

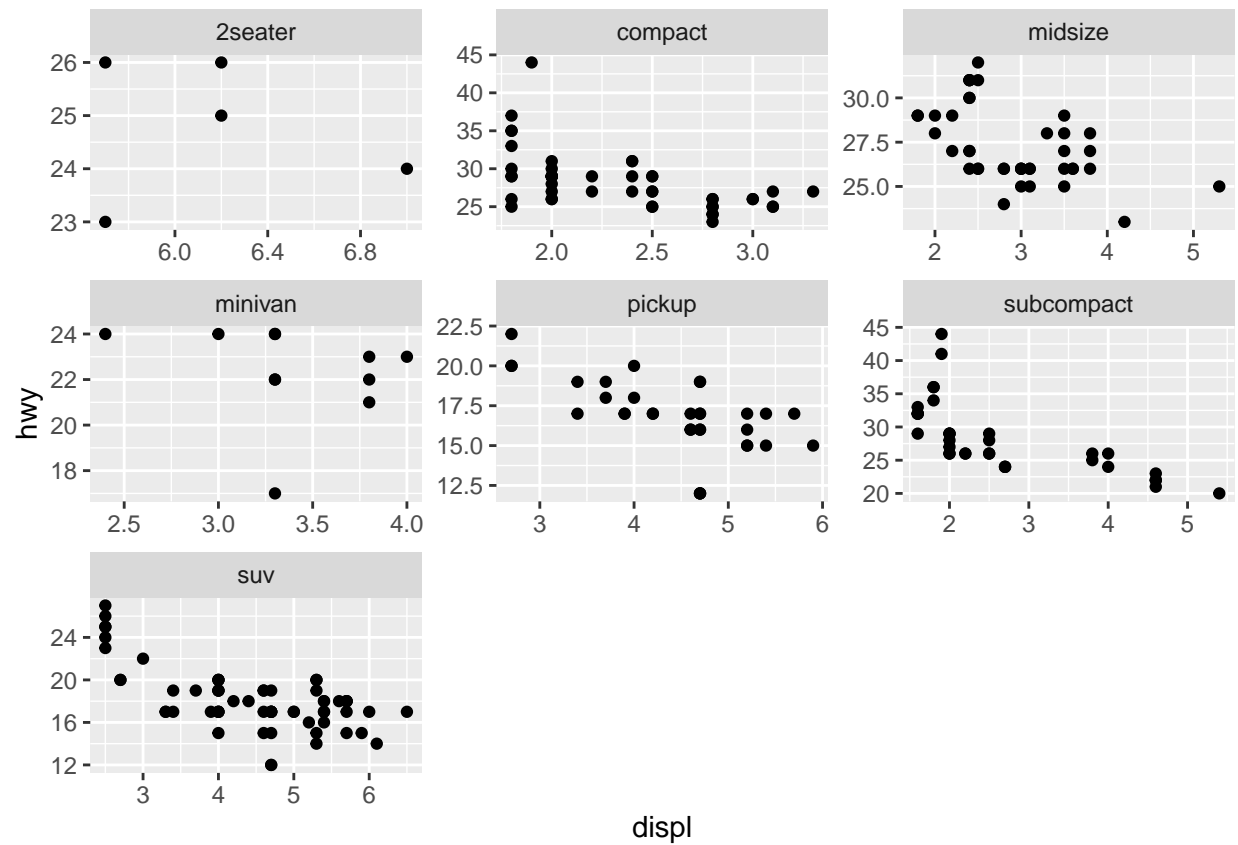
?facet\_wrap

```
ggplot(mpg, aes(displ, hwy)) +  
  geom_point() +  
  facet_wrap(~ class, nrow = 1)
```



*#What does the scales argument to facet\_wrap() do? When might you use it?*

```
ggplot(mpg, aes(displ, hwy)) +  
  geom_point() +  
  facet_wrap(~ class, scales = "free")
```

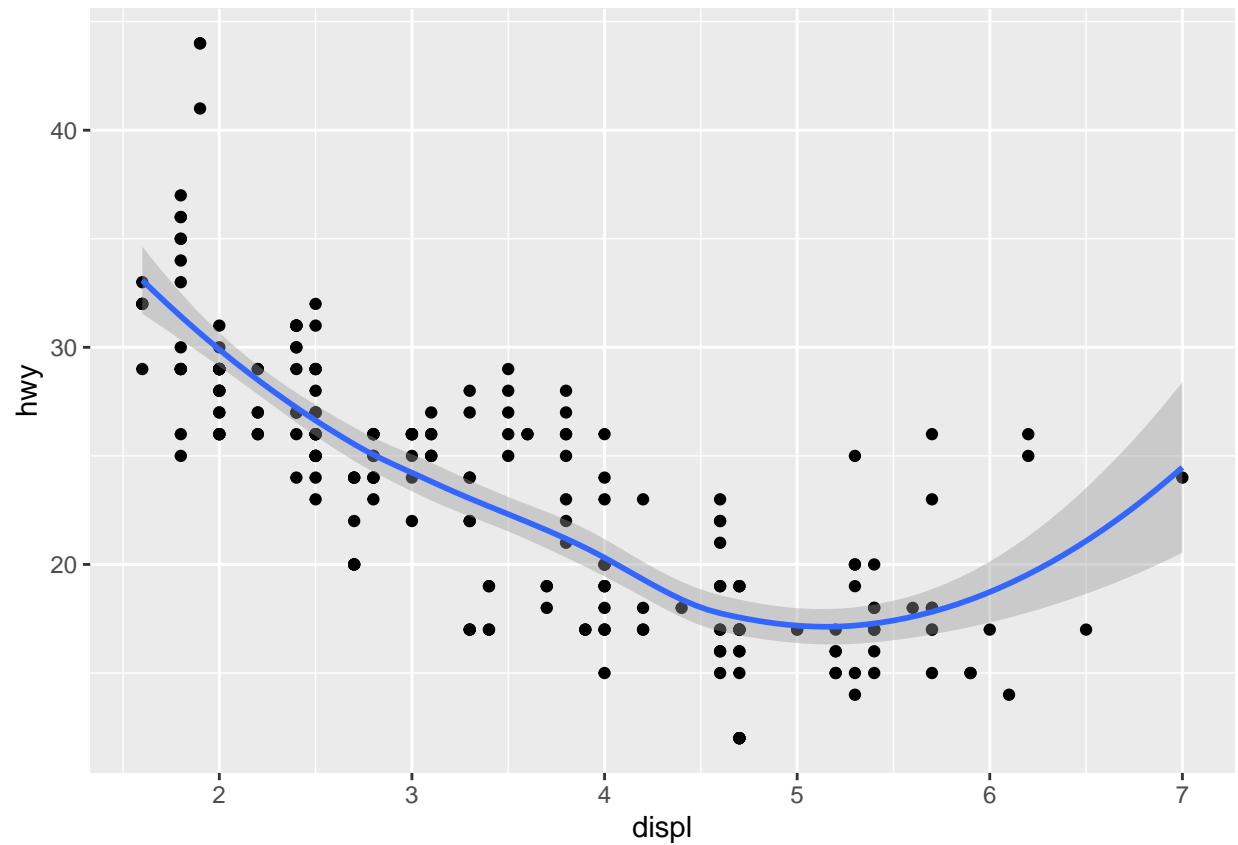


## Geoms

```
ggplot(mpg, aes(displ, hwy)) +  
  geom_point() +  
  geom_smooth()
```

## Geom\_smooth

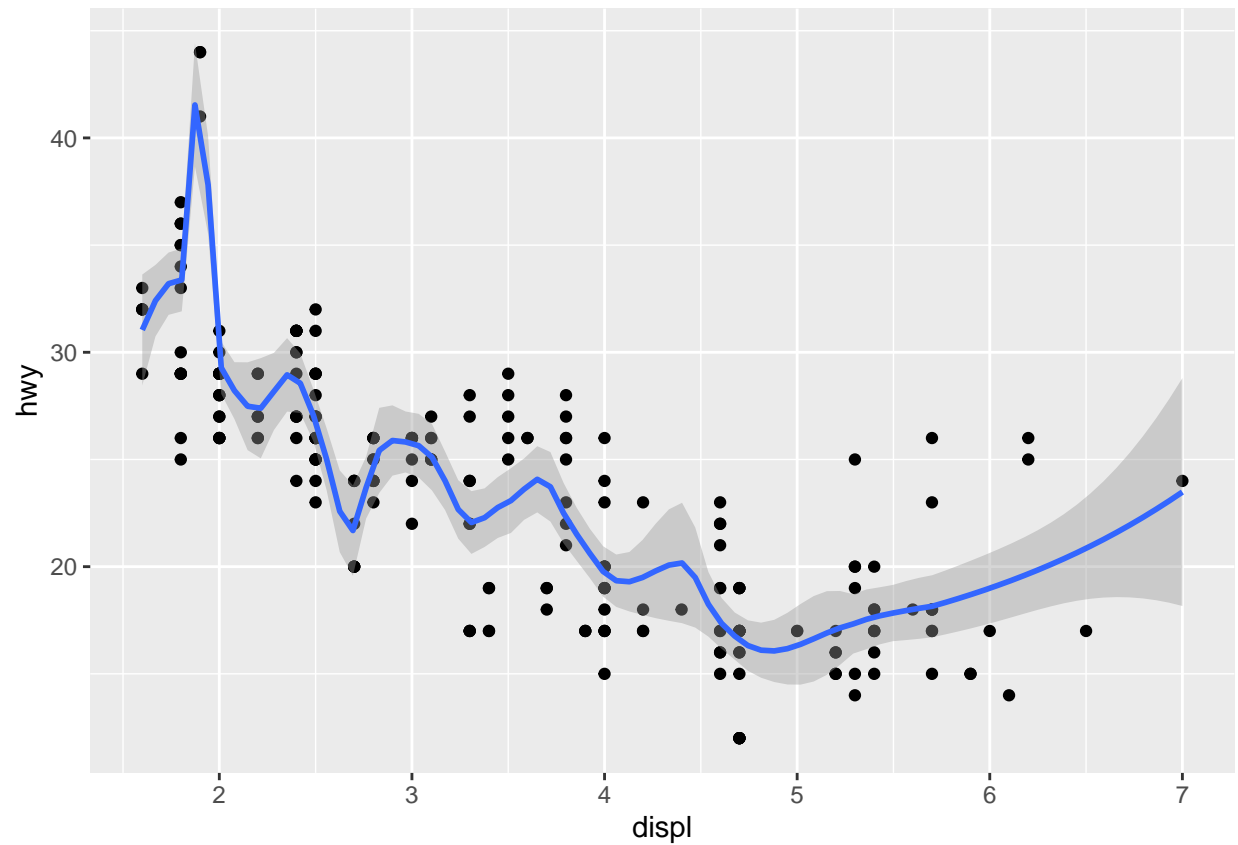
```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



```
# Smoothed line shows dominant pattern and level of uncertainty  
# Method = "loess" is default for small n (>1,000 points)
```

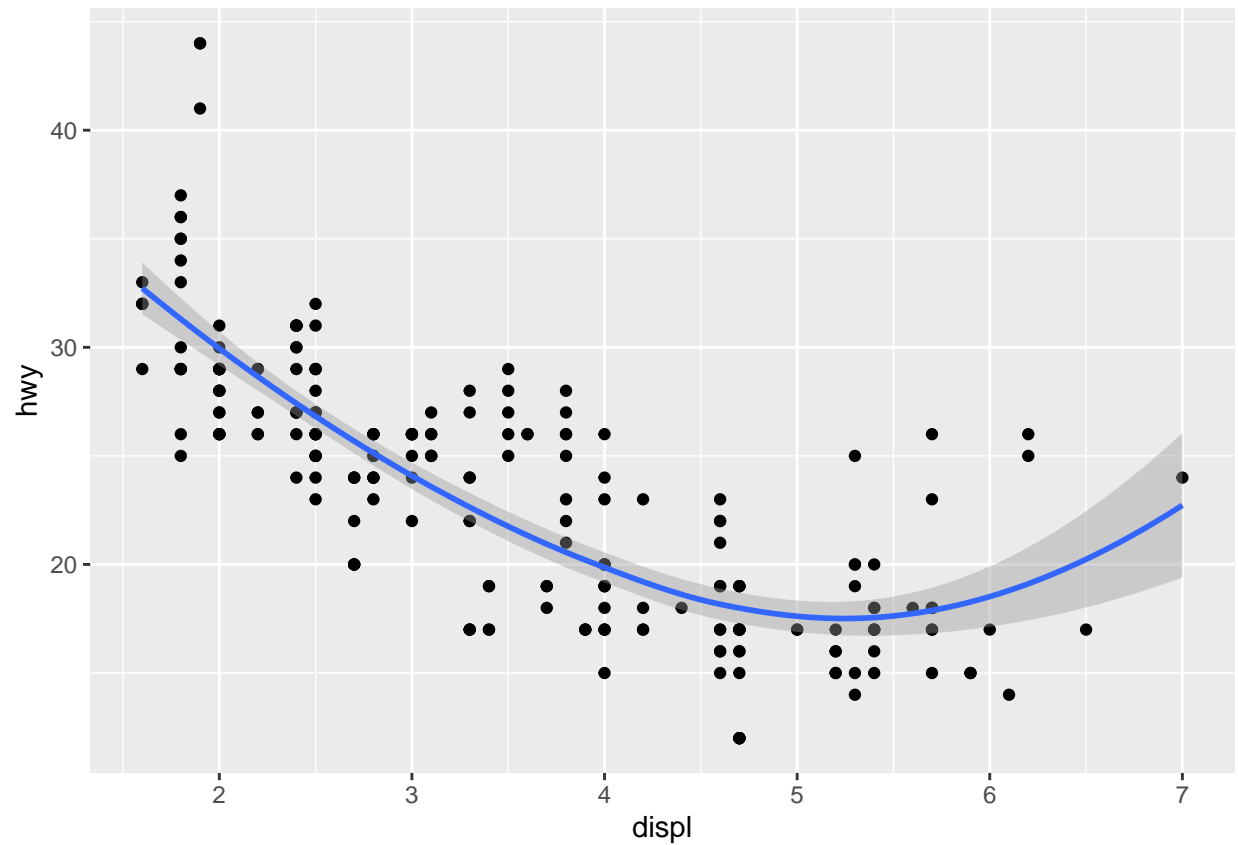
```
ggplot(mpg, aes(displ, hwy)) +  
  geom_point() +  
  geom_smooth(span = 0.2)
```

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



```
ggplot(mpg, aes(displ, hwy)) +  
  geom_point() +  
  geom_smooth(span = 1)
```

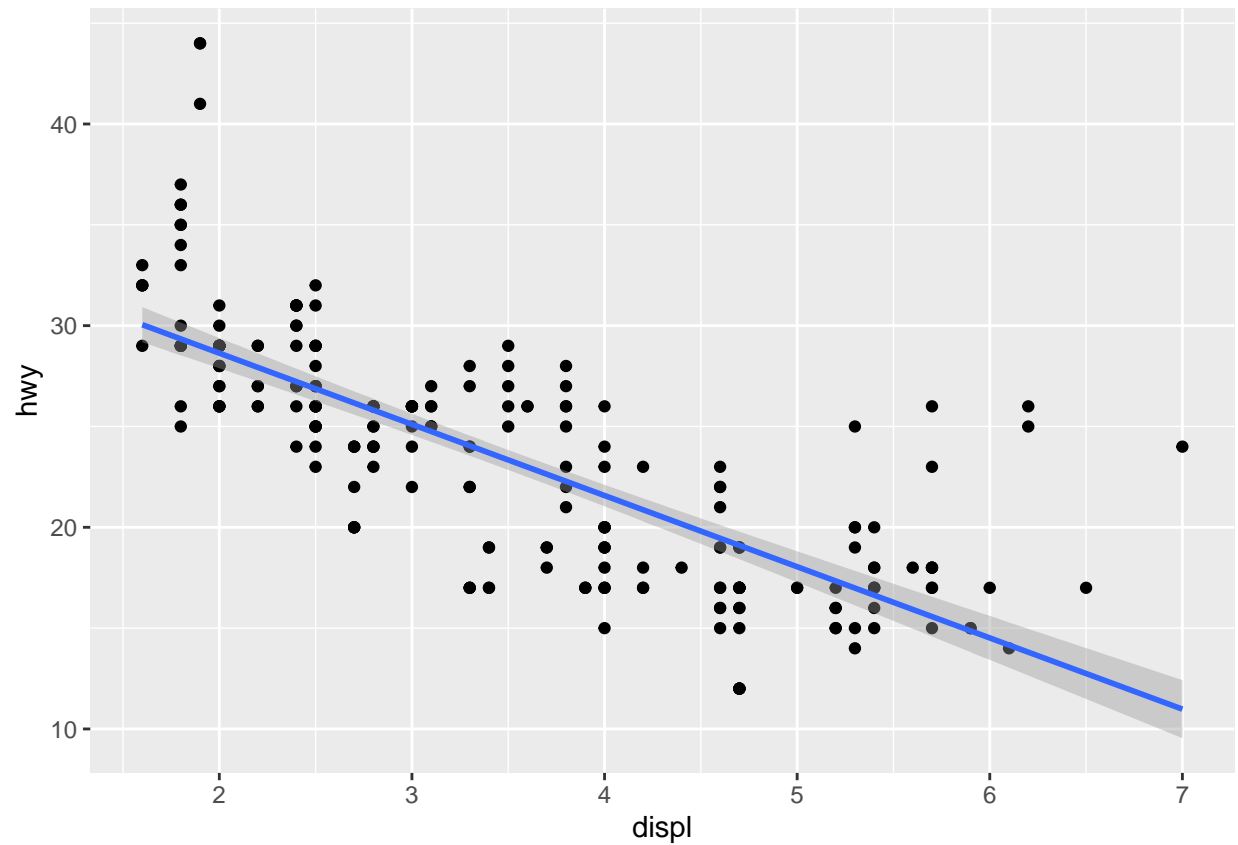
```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



```
# Span control "wiggleness of line"
```

```
ggplot(mpg, aes(displ, hwy)) +  
  geom_point() +  
  geom_smooth(method = "lm")
```

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



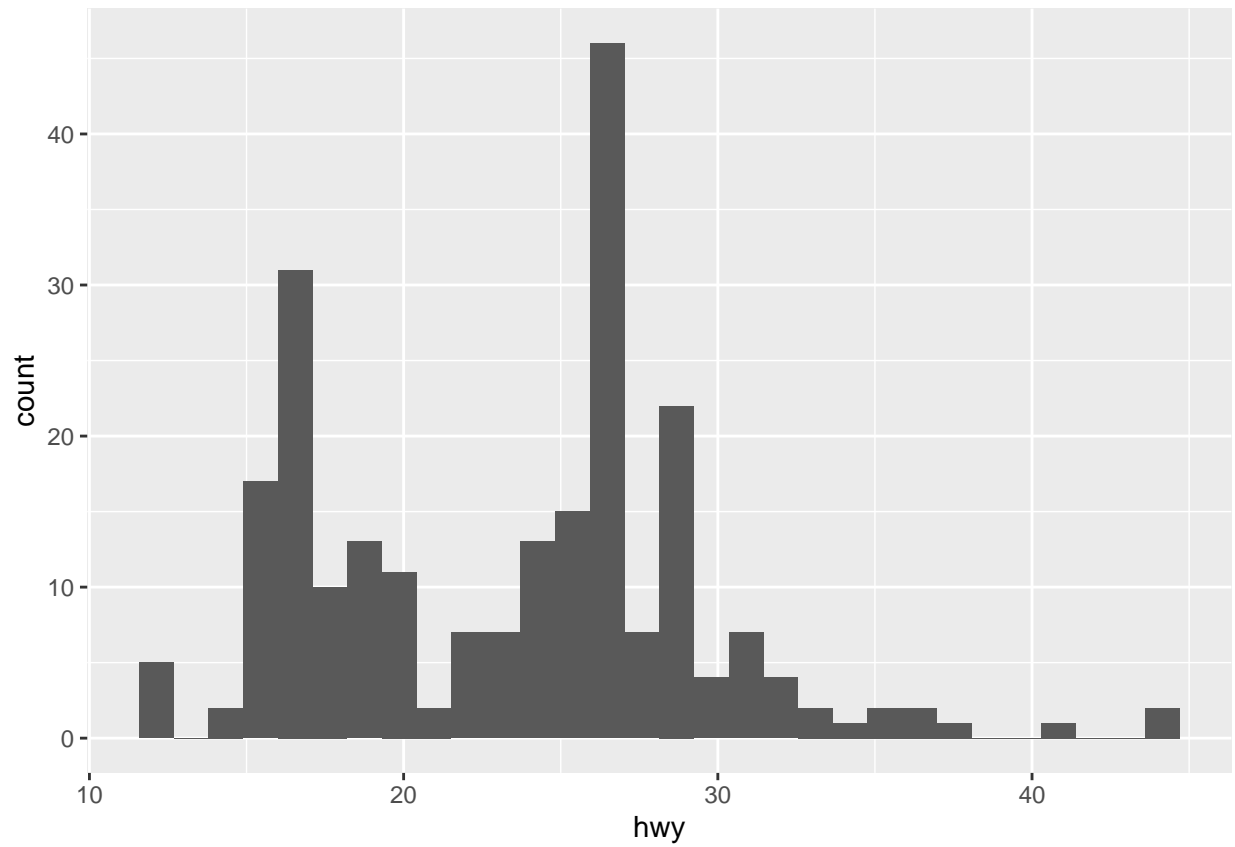
*# Linear model gives the line of best fit*

```
ggplot(mpg, aes(hwy)) +  
  geom_histogram()
```

**Histograms and frequency polygons**

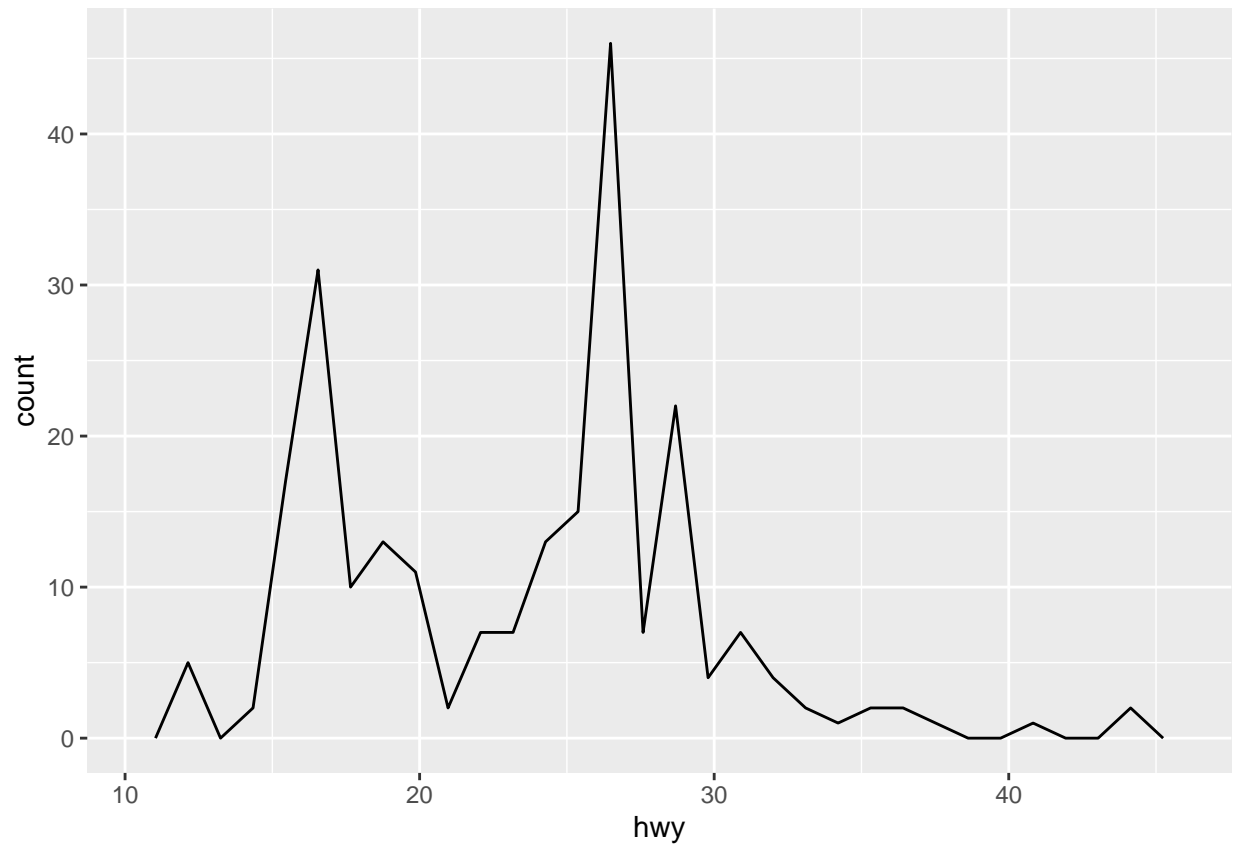
## ``stat_bin()`` using ``bins = 30``. Pick better value with ``binwidth``.



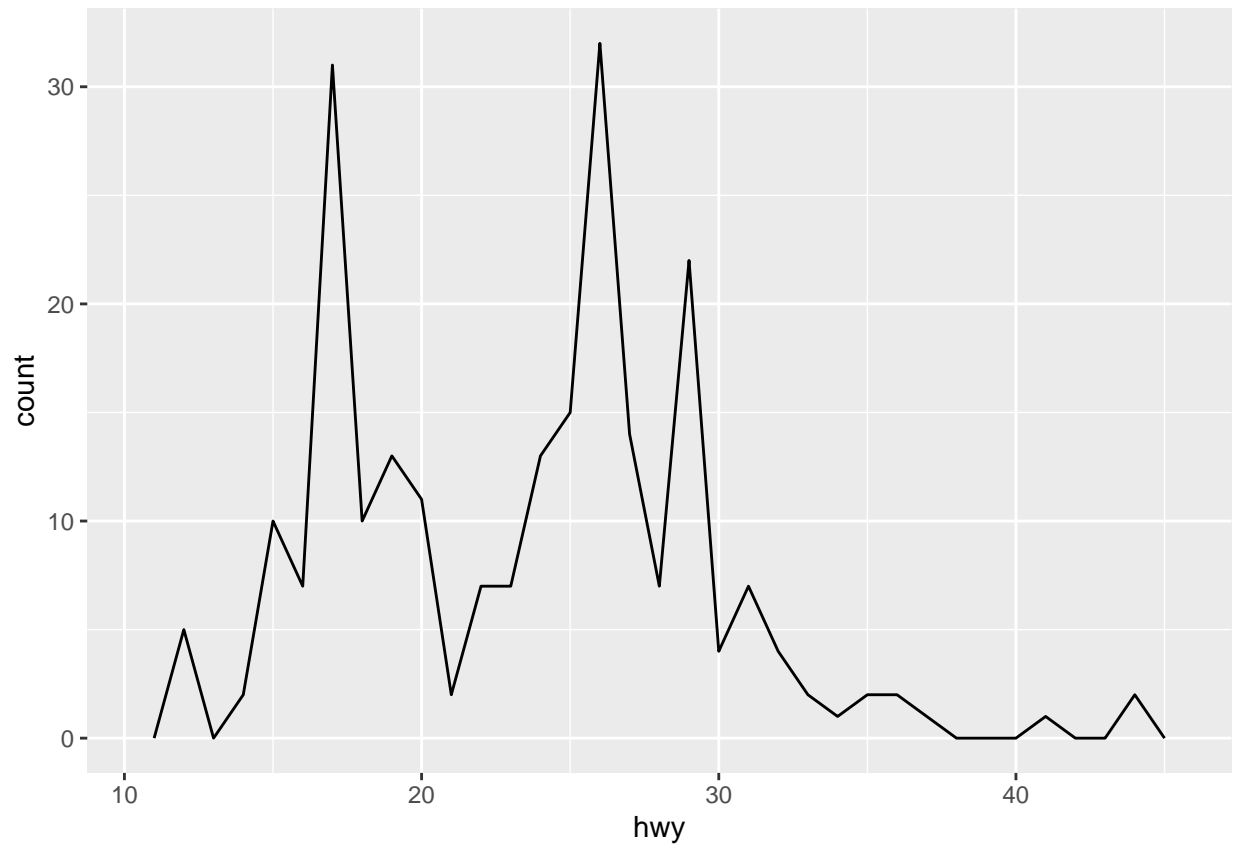


```
ggplot(mpg, aes(hwy)) +  
  geom_freqpoly()
```

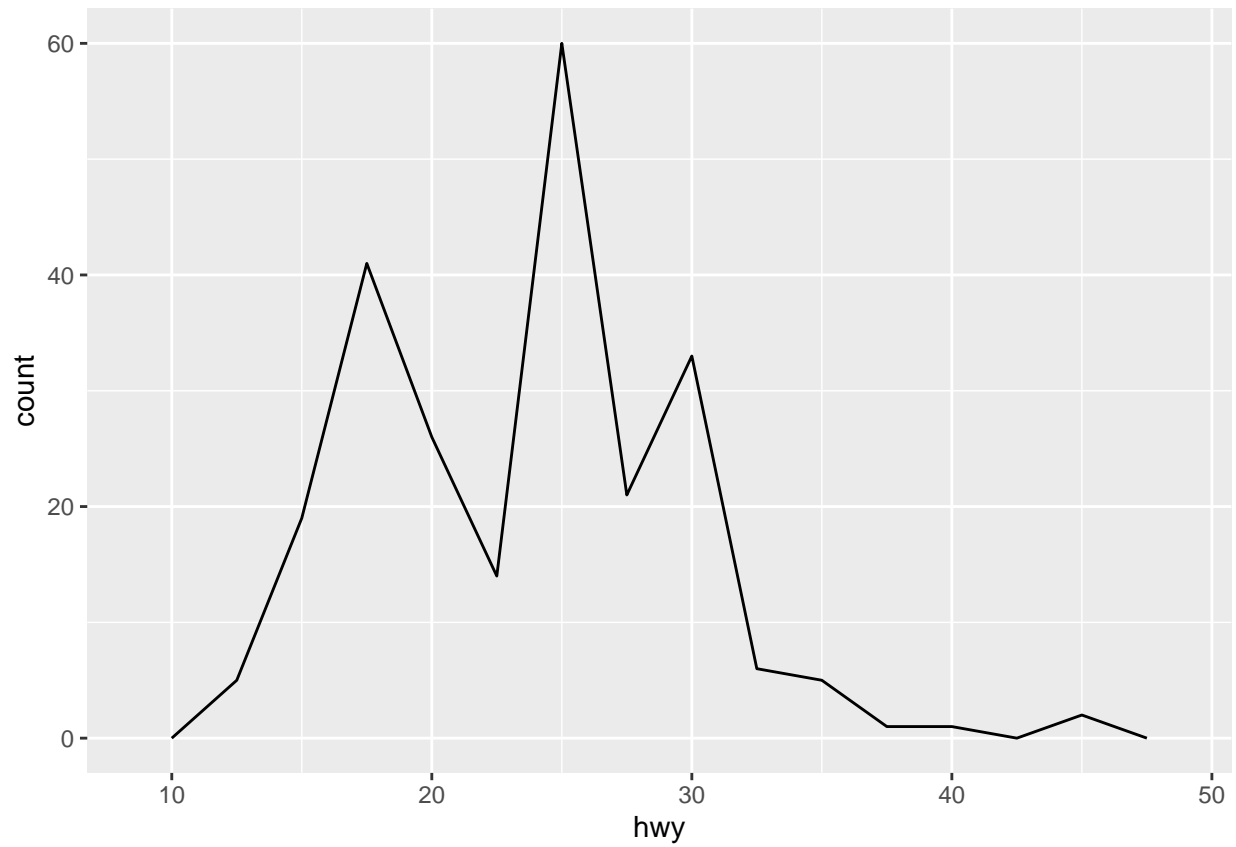
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



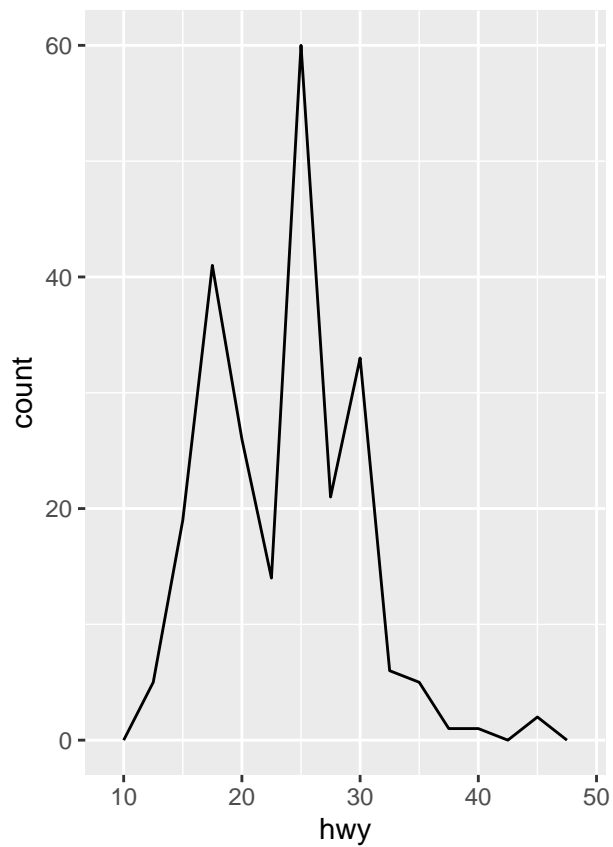
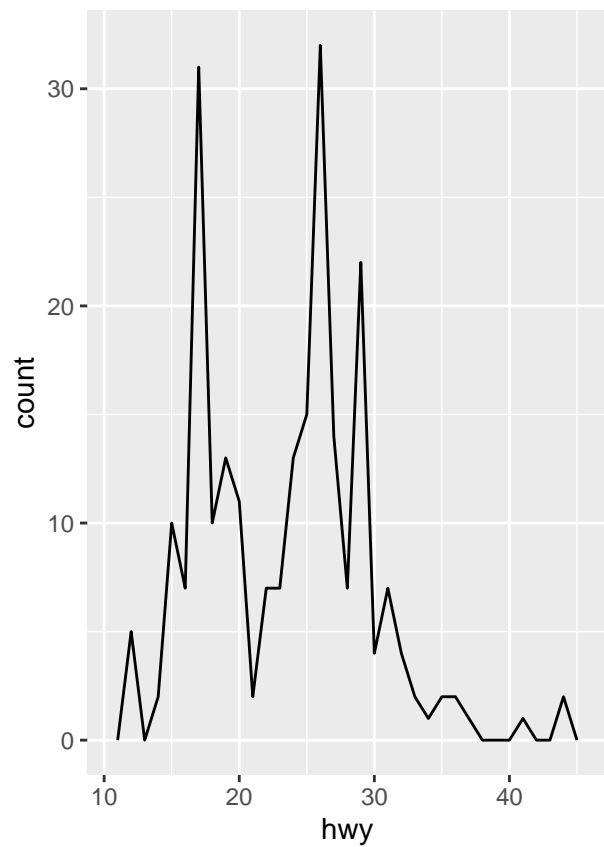
```
# Binwidth controls the width of bins (probably NOT ideal for your data, try experimenting)  
ggplot(mpg, aes(hwy)) +  
  geom_freqpoly(binwidth = 1)
```



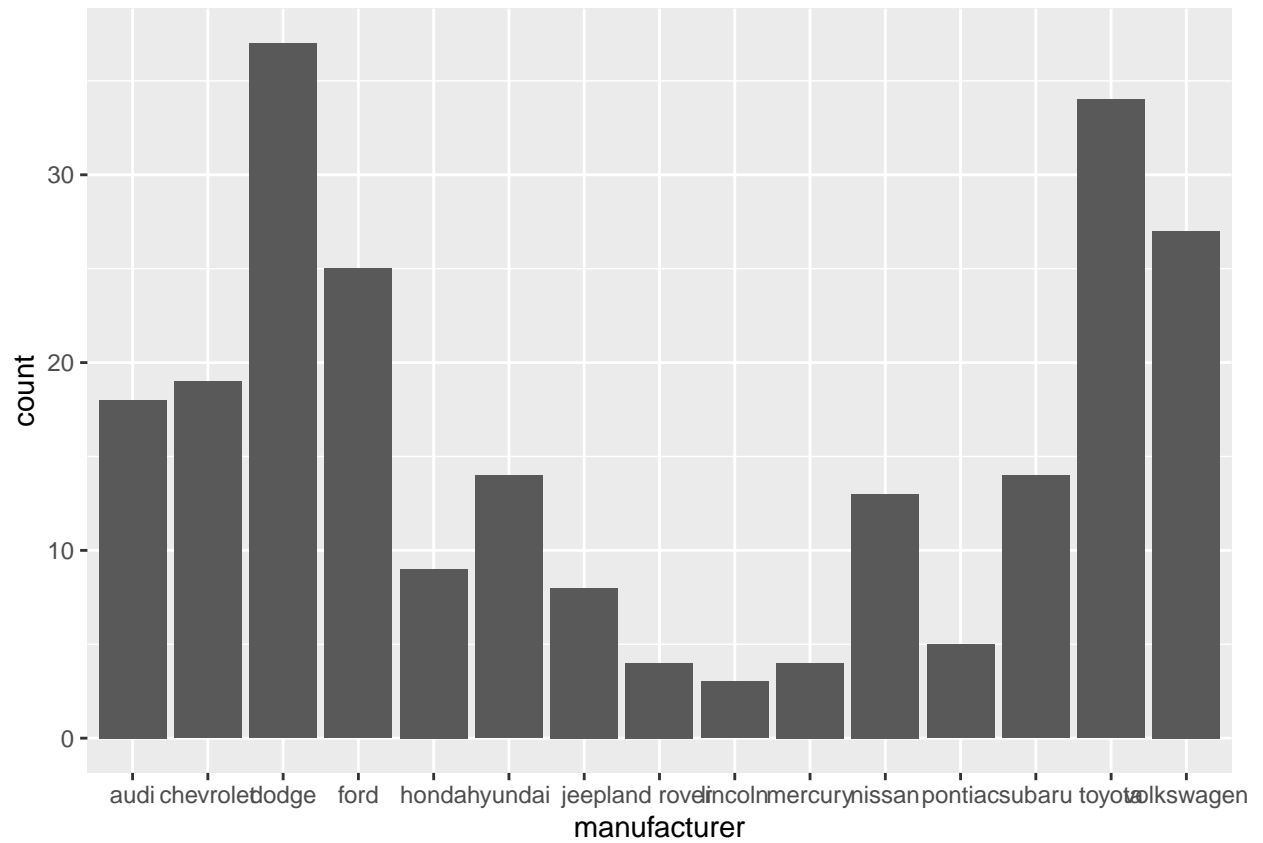
```
ggplot(mpg, aes(hwy)) +  
  geom_freqpoly(binwidth = 2.5)
```



```
p3 <- ggplot(mpg, aes(hwy)) +  
  geom_freqpoly(binwidth = 1)  
  
p4 <- ggplot(mpg, aes(hwy)) +  
  geom_freqpoly(binwidth = 2.5)  
  
grid.arrange(p3, p4, ncol = 2)
```

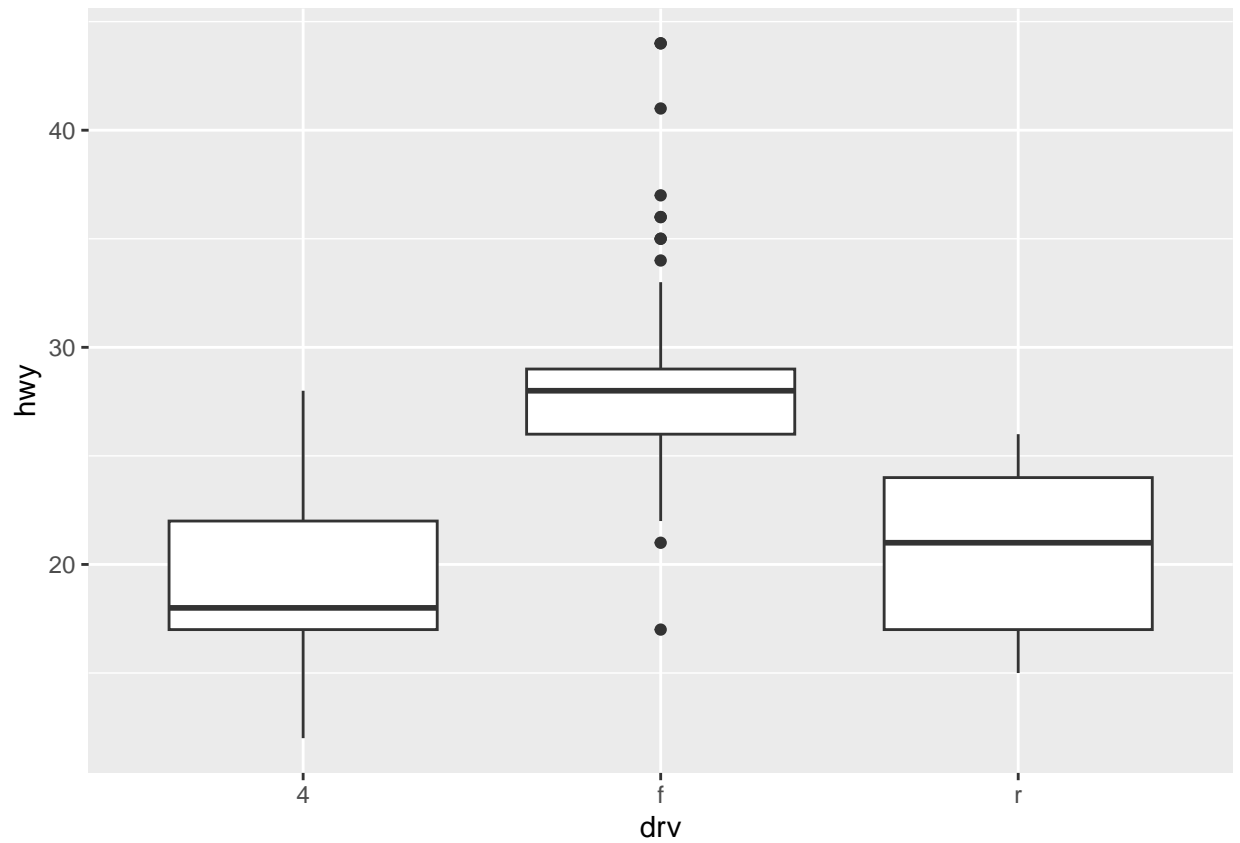


```
ggplot(mpg, aes(manufacturer)) +  
  geom_bar()
```



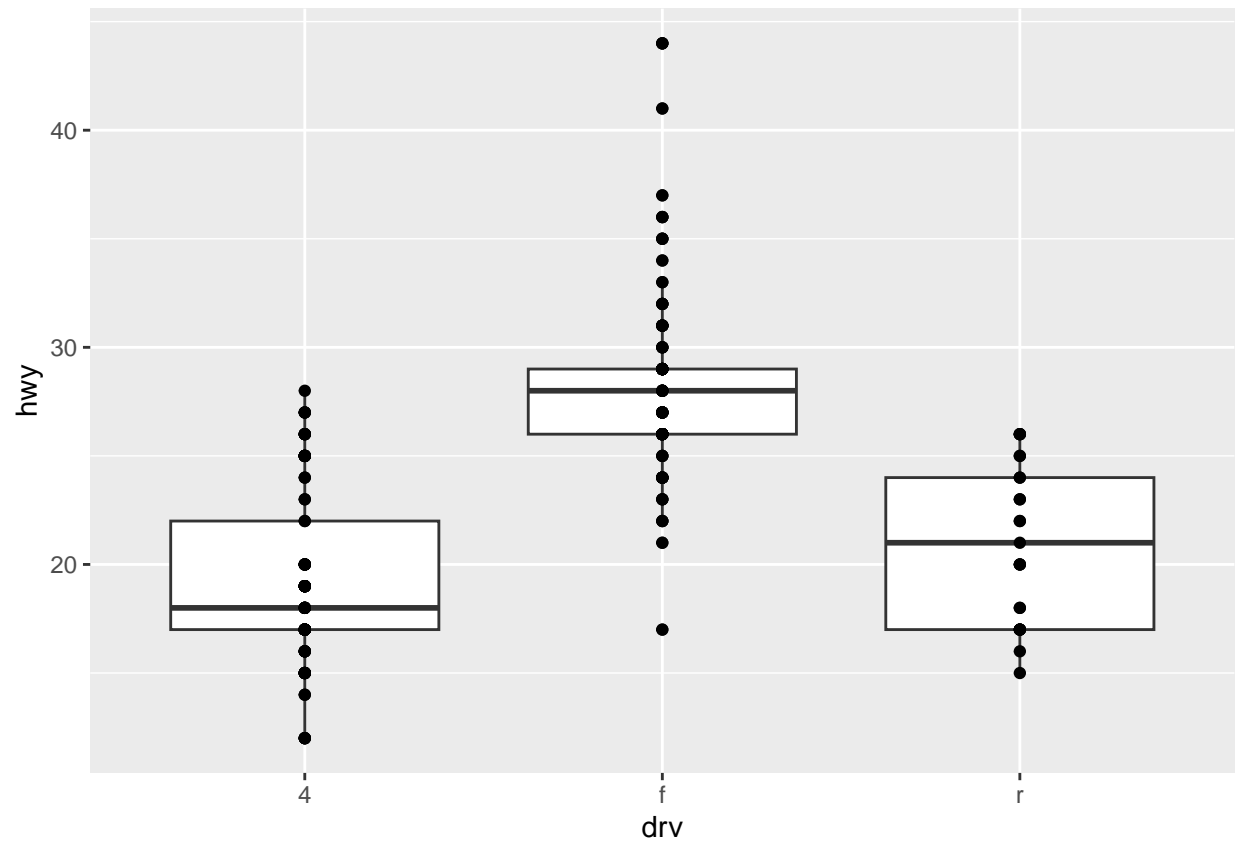
Bar chart

```
ggplot(mpg, aes(drv, hwy)) +  
  geom_boxplot()
```



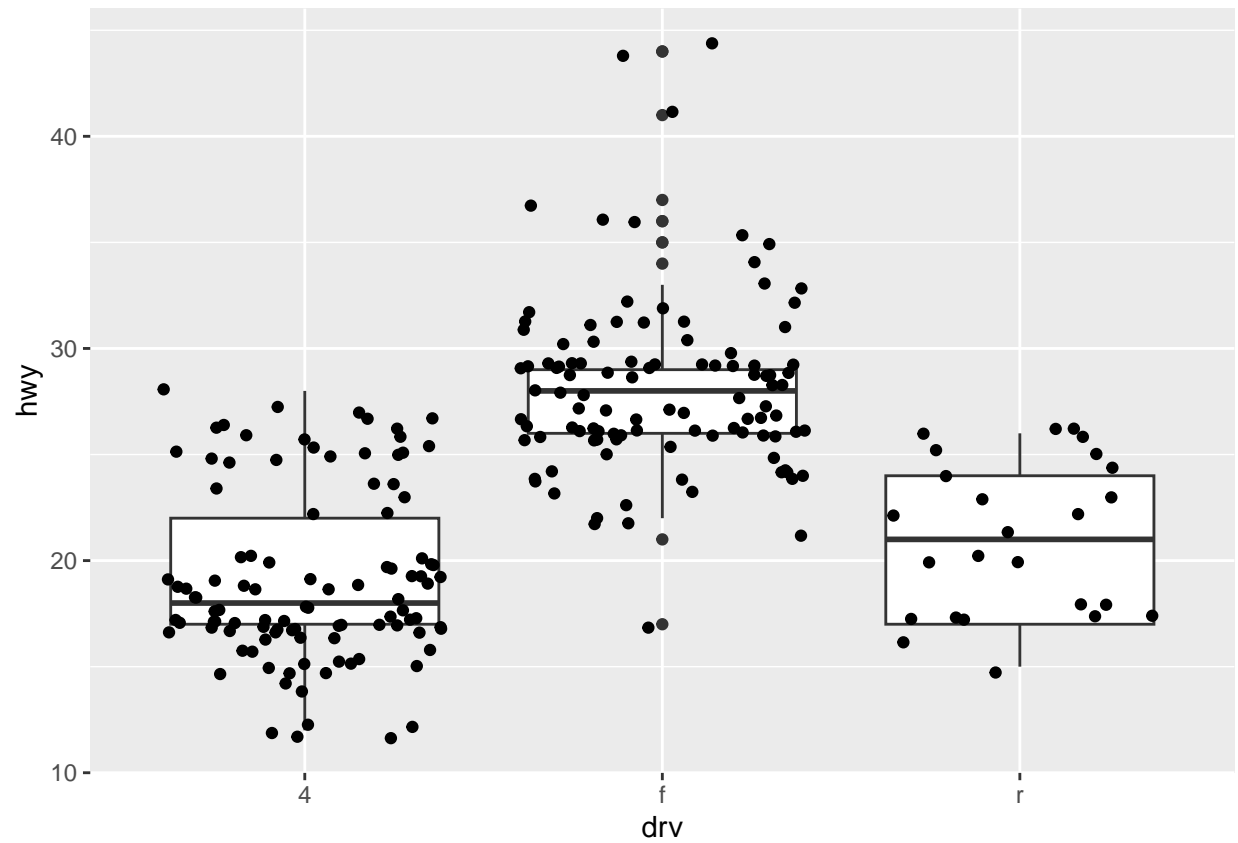
Boxplot

```
ggplot(mpg, aes(drv, hwy)) +  
  geom_boxplot() +  
  geom_point()
```

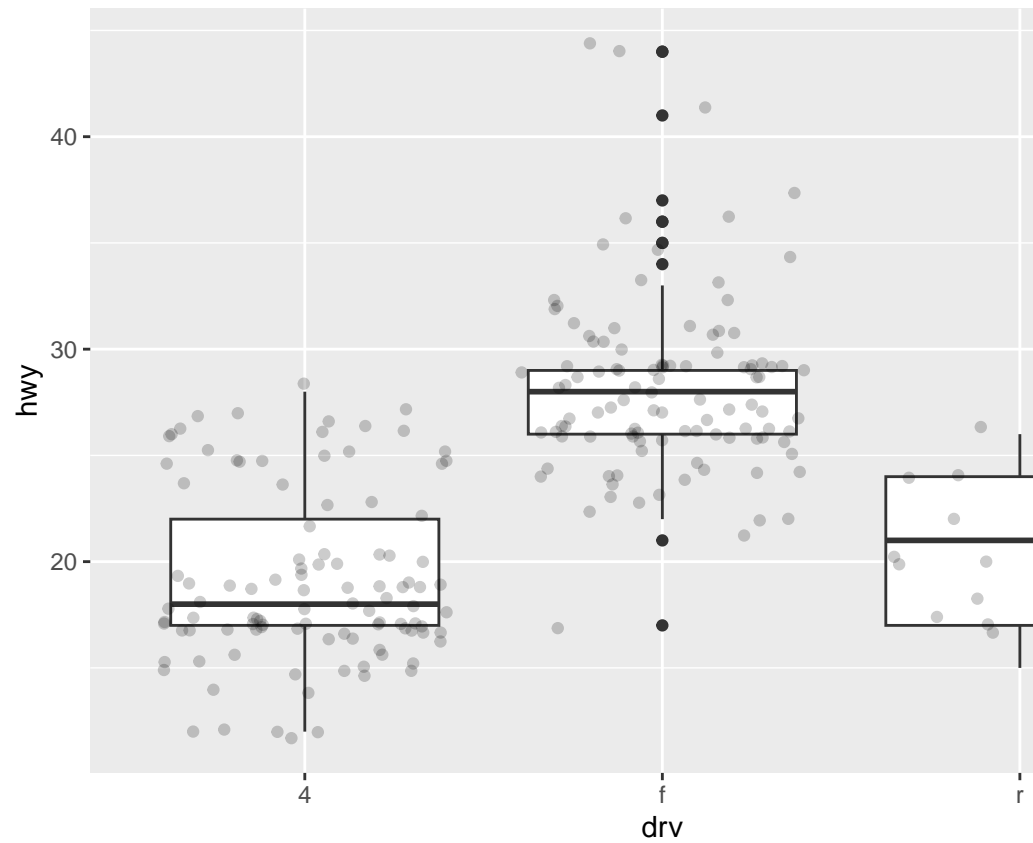


```
ggplot(mpg, aes(drv, hwy)) +  
  geom_boxplot() +  
  geom_jitter()
```



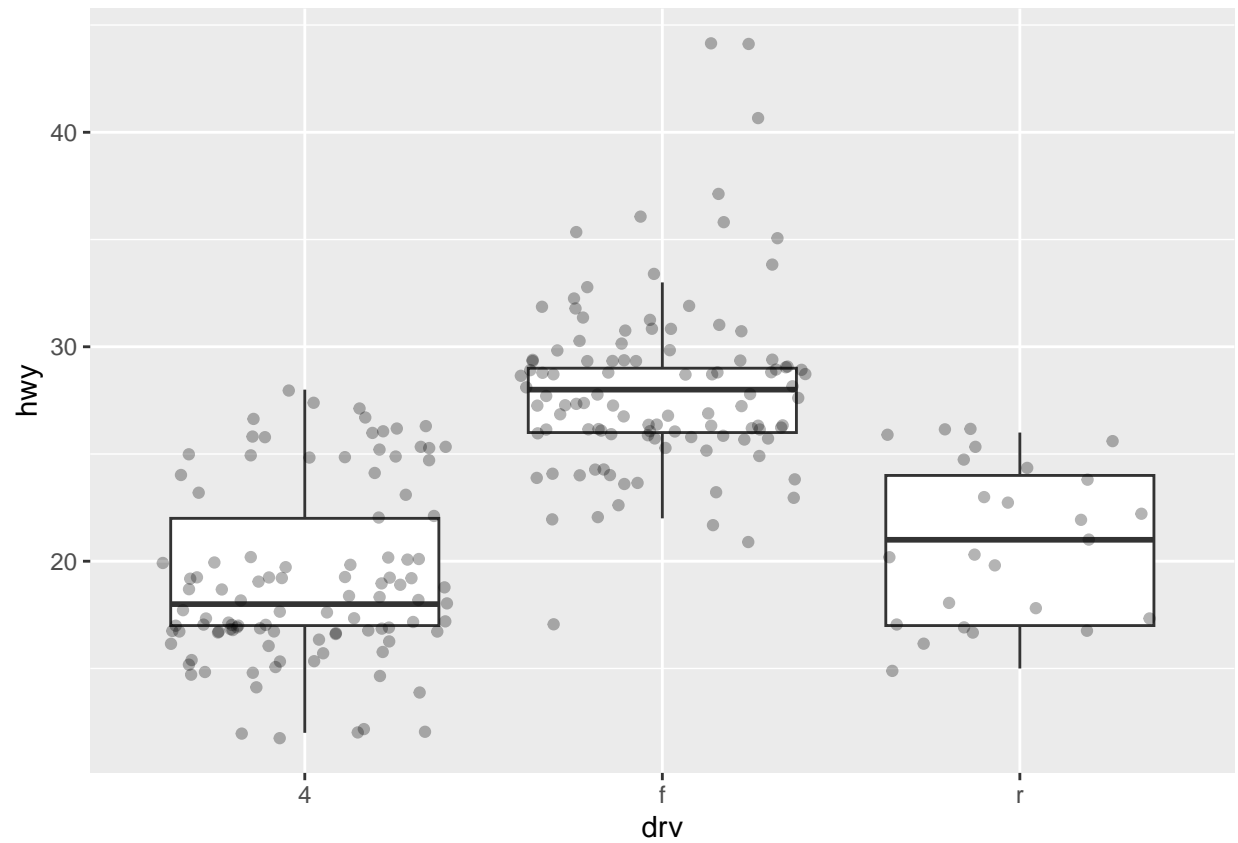


```
ggplot(mpg, aes(drv, hwy)) +  
  geom_boxplot() +  
  geom_jitter(alpha = 0.2)
```

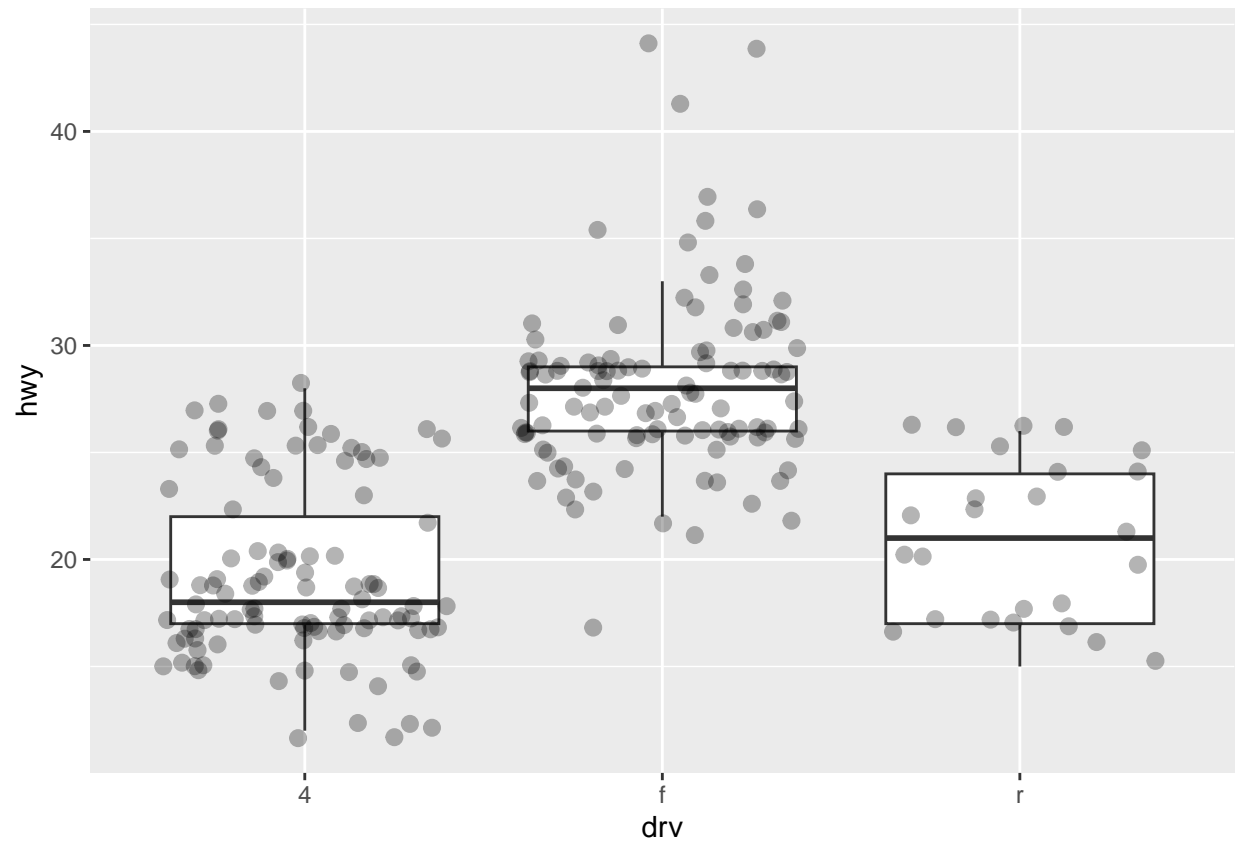


Size and transparency control

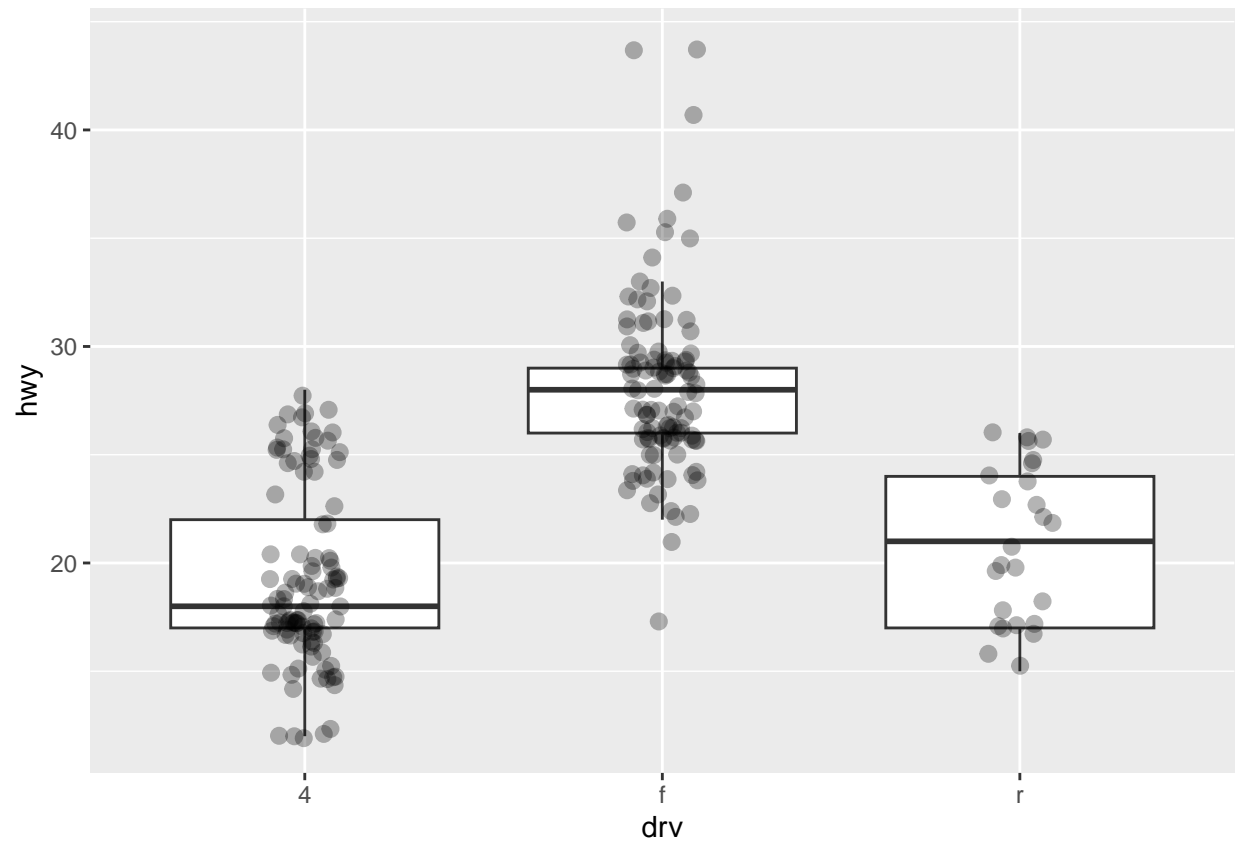
```
ggplot(mpg, aes(drv, hwy)) +  
  geom_boxplot(outlier.shape = NA) +  
  geom_jitter(alpha = 0.3)
```



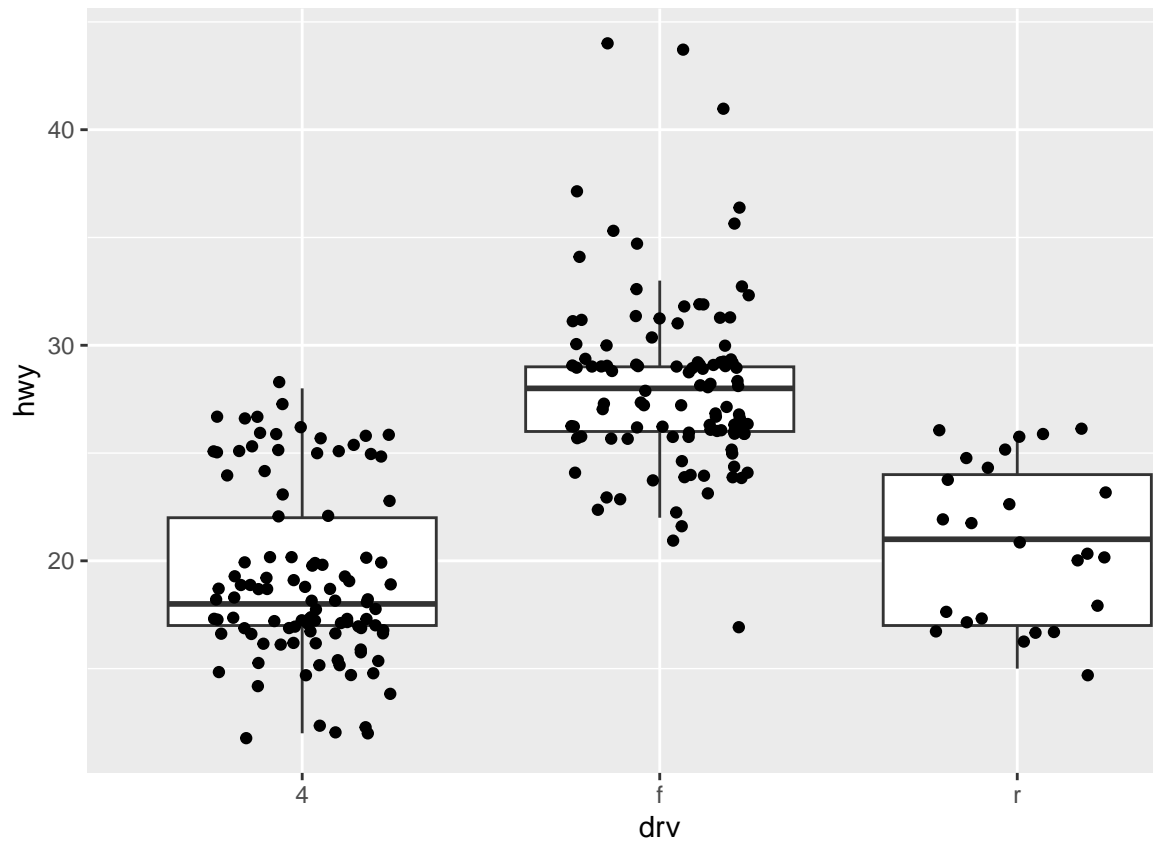
```
ggplot(mpg, aes(drv, hwy)) +  
  geom_boxplot(outlier.shape = NA) +  
  geom_jitter(alpha = 0.3, size = 2.5)
```



```
ggplot(mpg, aes(drv, hwy)) +  
  geom_boxplot(outlier.shape = NA) +  
  geom_jitter(alpha = 0.3, size = 2.5, width = 0.1)
```



```
ggplot(mpg, aes(drv, hwy)) +  
  geom_boxplot(outlier.shape = NA) +  
  geom_jitter(width = 0.25)
```

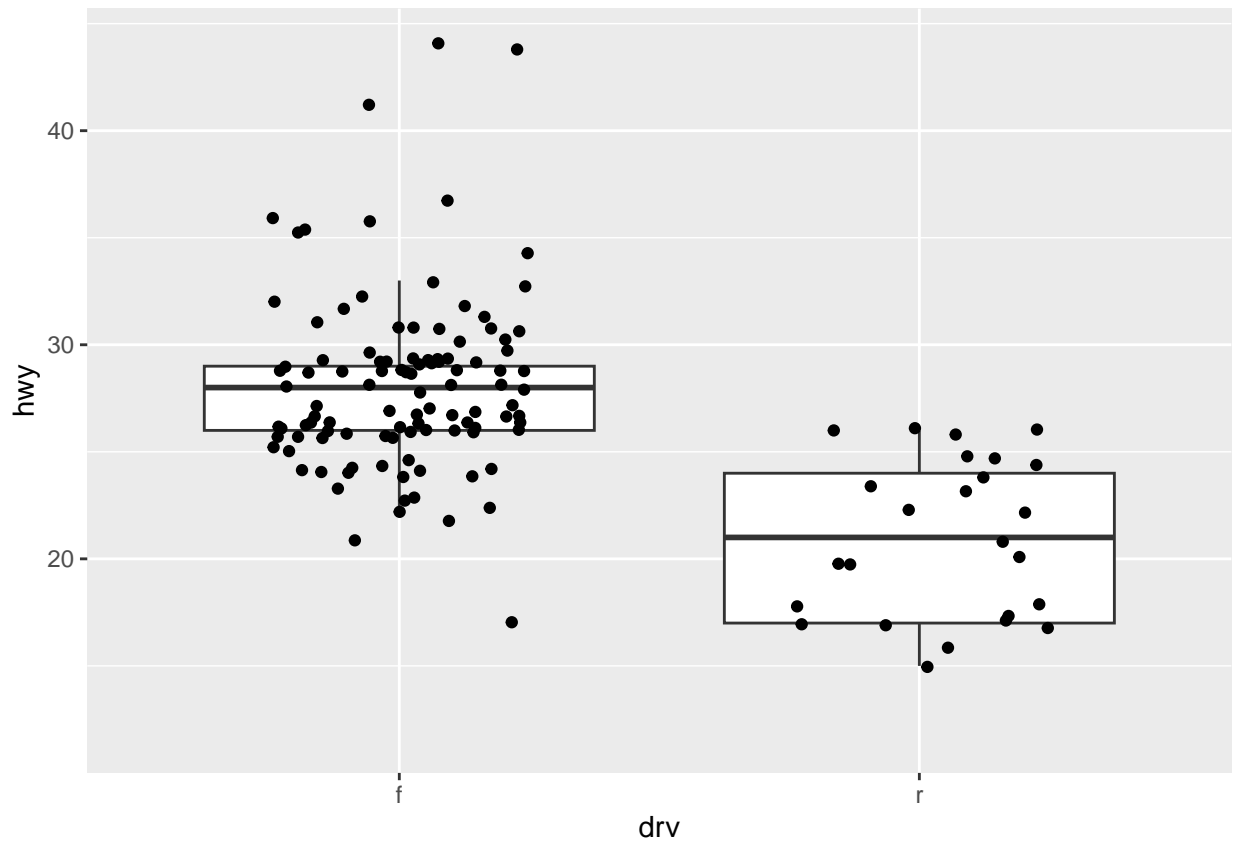


Modification of axes

```
ggplot(mpg, aes(drv, hwy)) +  
  geom_boxplot(outlier.shape = NA) +  
  geom_jitter(width = 0.25) +  
  xlim("f", "r")
```

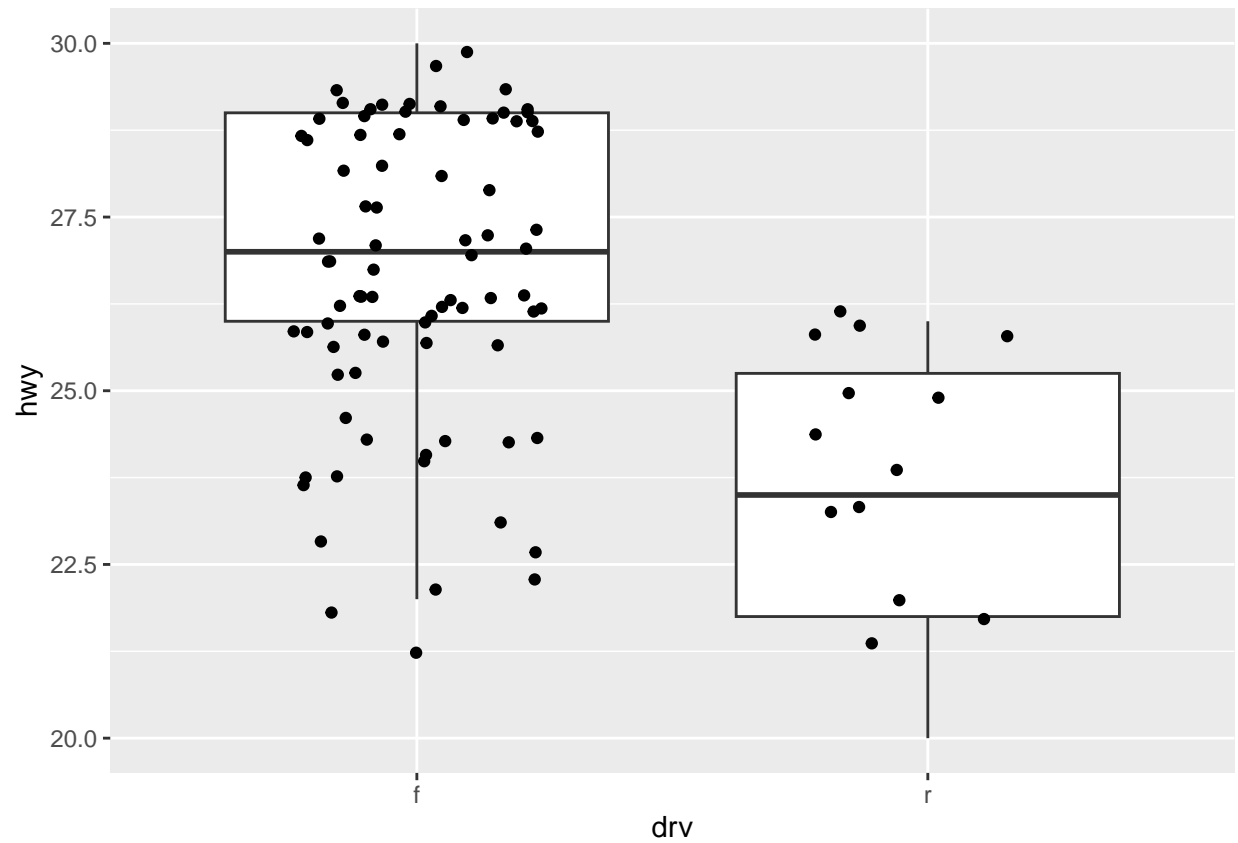
```
## Warning: Removed 103 rows containing missing values (`stat_boxplot()`).
```

```
## Warning: Removed 103 rows containing missing values (`geom_point()`).
```



```
ggplot(mpg, aes(drv, hwy)) +  
  geom_boxplot(outlier.shape = NA) +  
  geom_jitter(width = 0.25) +  
  xlim("f", "r") +  
  ylim(20, 30)
```

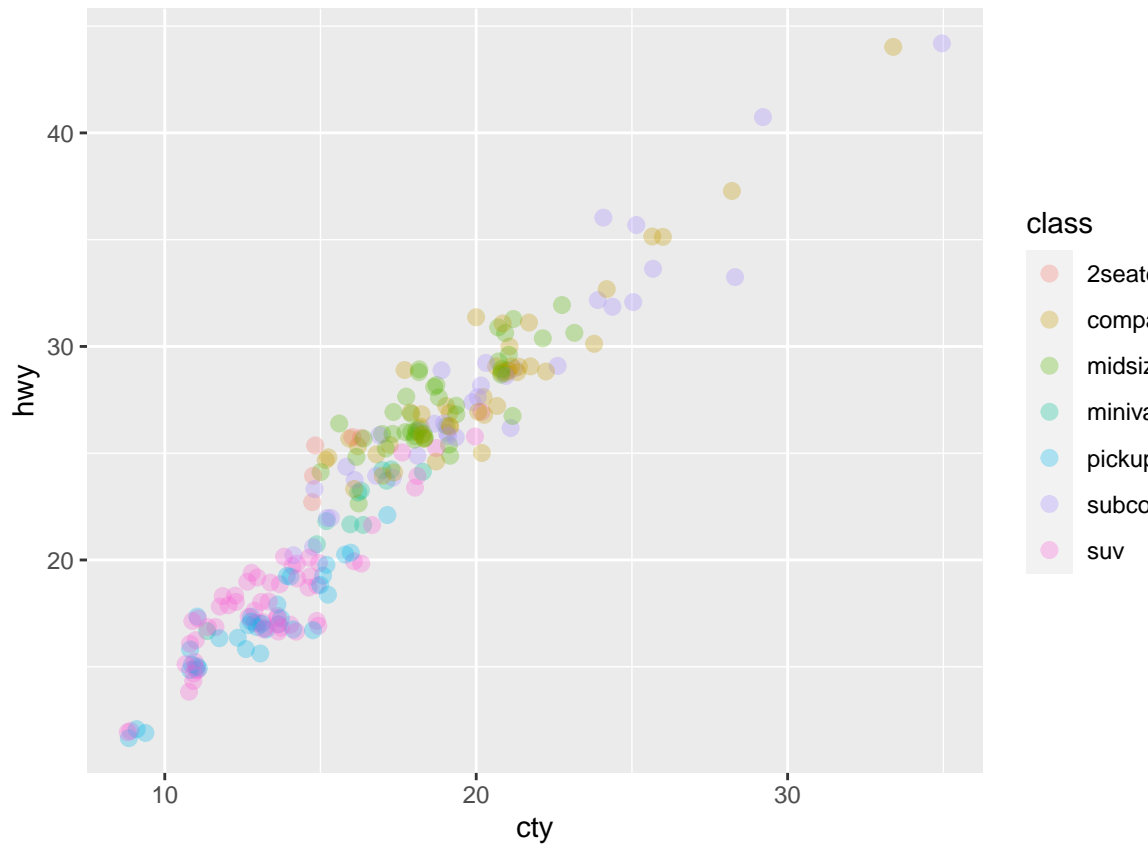
```
## Warning: Removed 103 rows containing missing values (`stat_boxplot()`).  
## Warning: Removed 32 rows containing non-finite values (`stat_boxplot()`).  
## Warning: Removed 140 rows containing missing values (`geom_point()`).
```



*#> Warning: Removed 138 rows containing missing values (`geom\_point()`).*

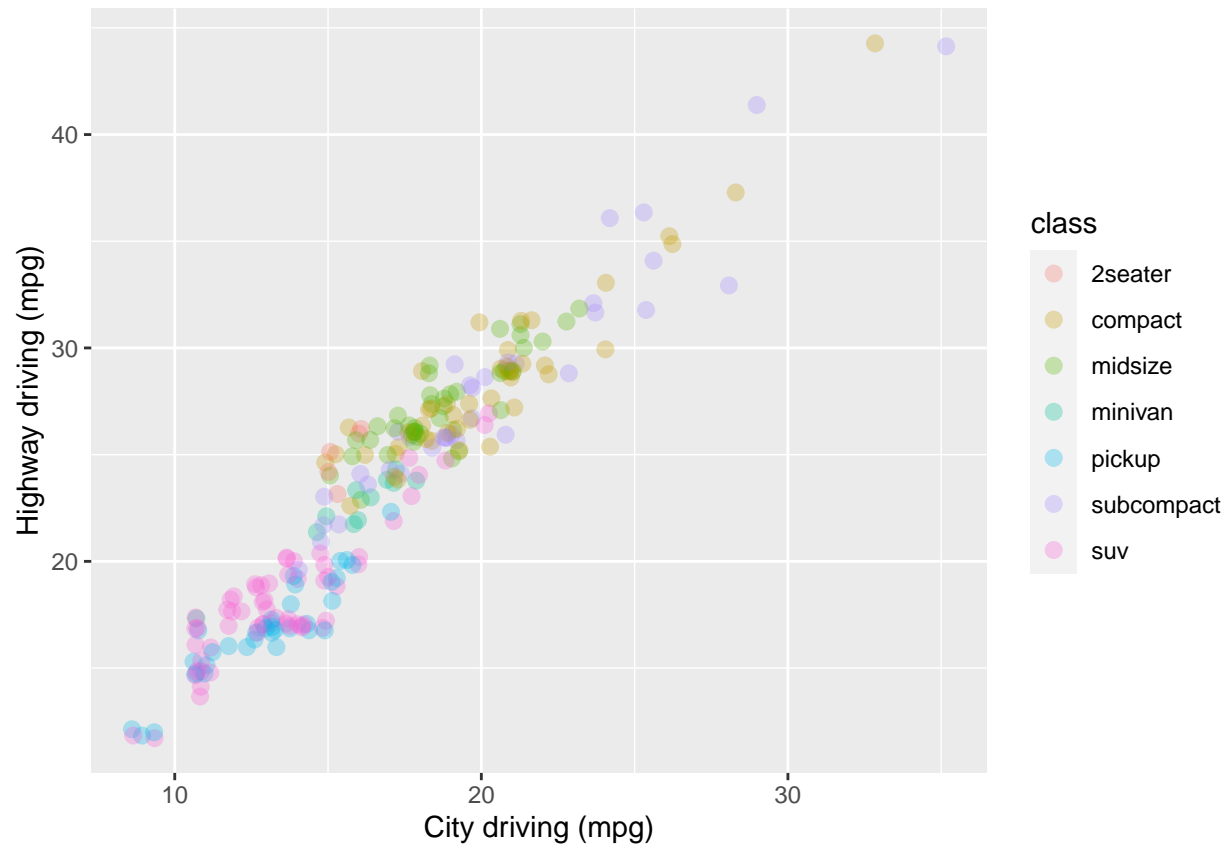
```
ggplot(mpg, aes(cty, hwy, color = class)) +  
  geom_jitter(alpha = 0.3, size = 2.5)
```



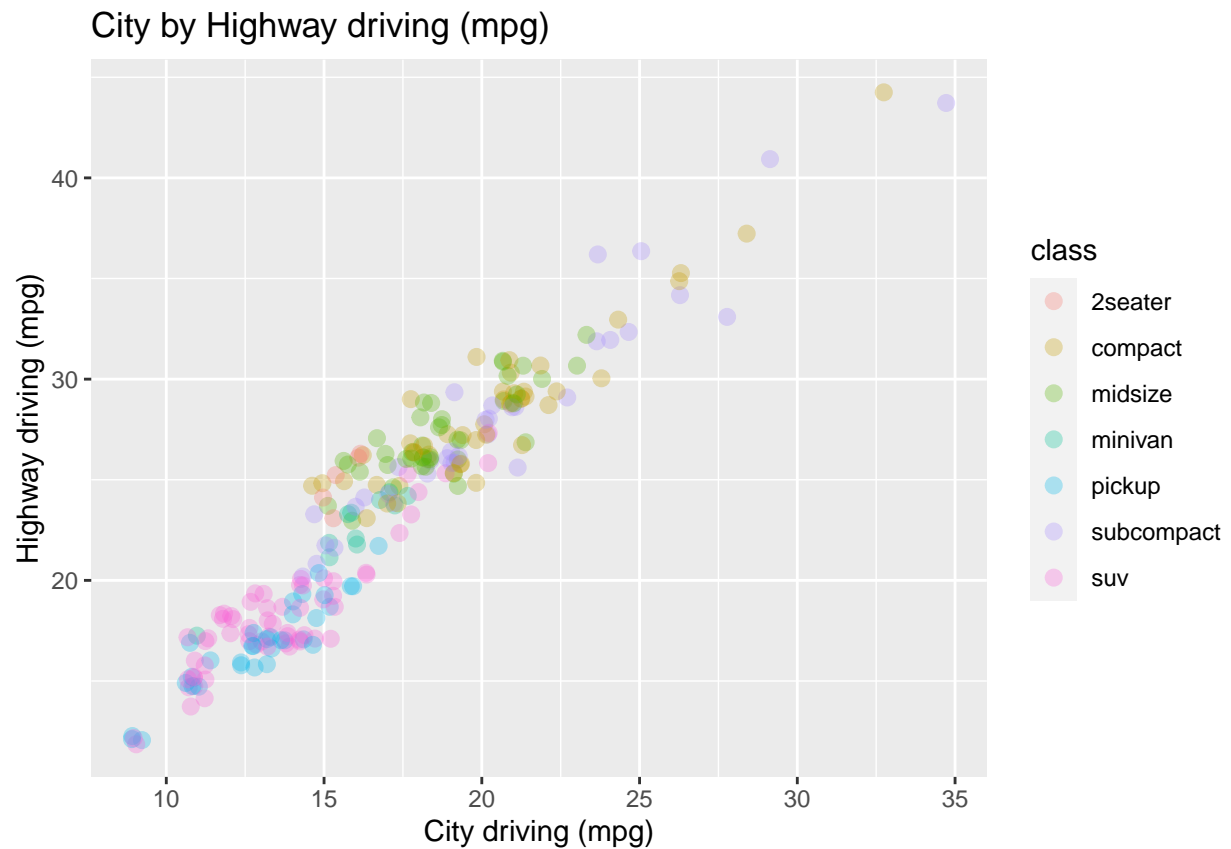


#### Title and axis names

```
# Change axis titles
ggplot(mpg, aes(cty, hwy, color = class)) +
  geom_jitter(alpha = 0.3, size = 2.5) +
  xlab("City driving (mpg)") +
  ylab("Highway driving (mpg)")
```

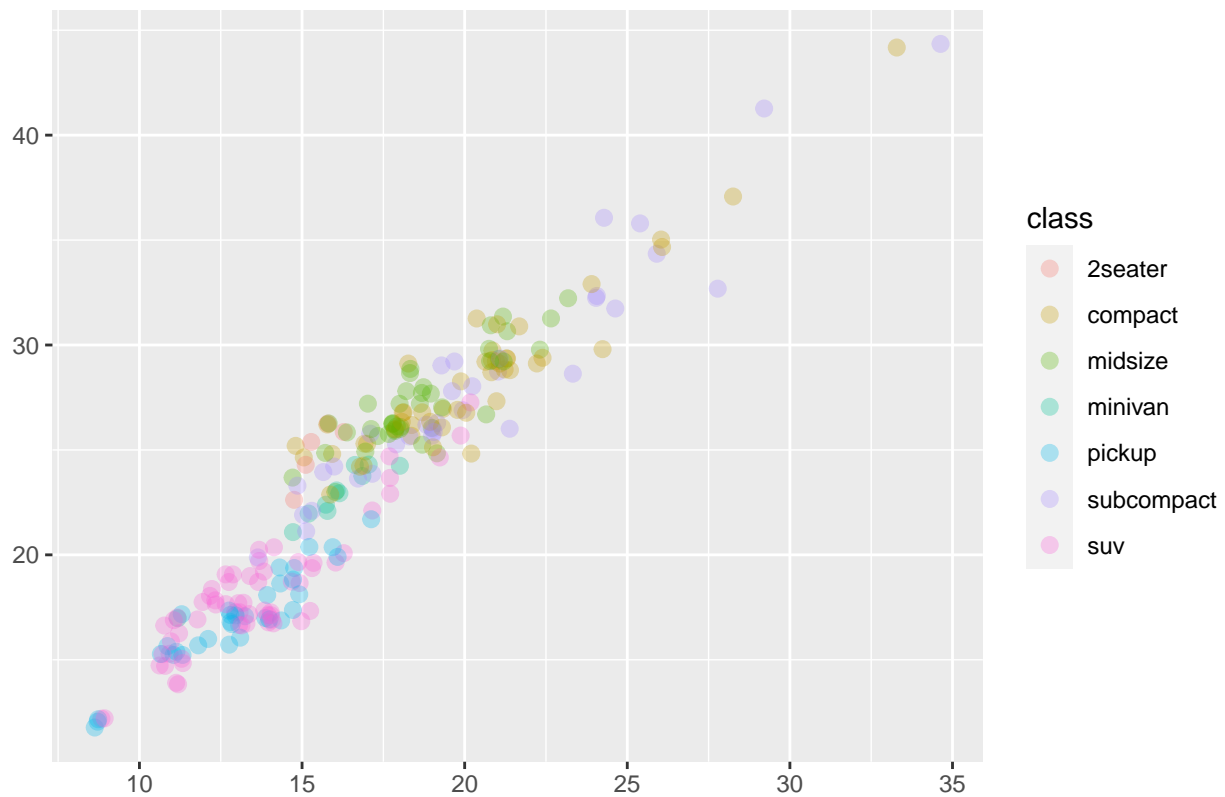


```
ggplot(mpg, aes(cty, hwy, color = class)) +  
  geom_jitter(alpha = 0.3, size = 2.5) +  
  xlab("City driving (mpg)") +  
  ylab("Highway driving (mpg)") +  
  ggtitle("City by Highway driving (mpg)")
```



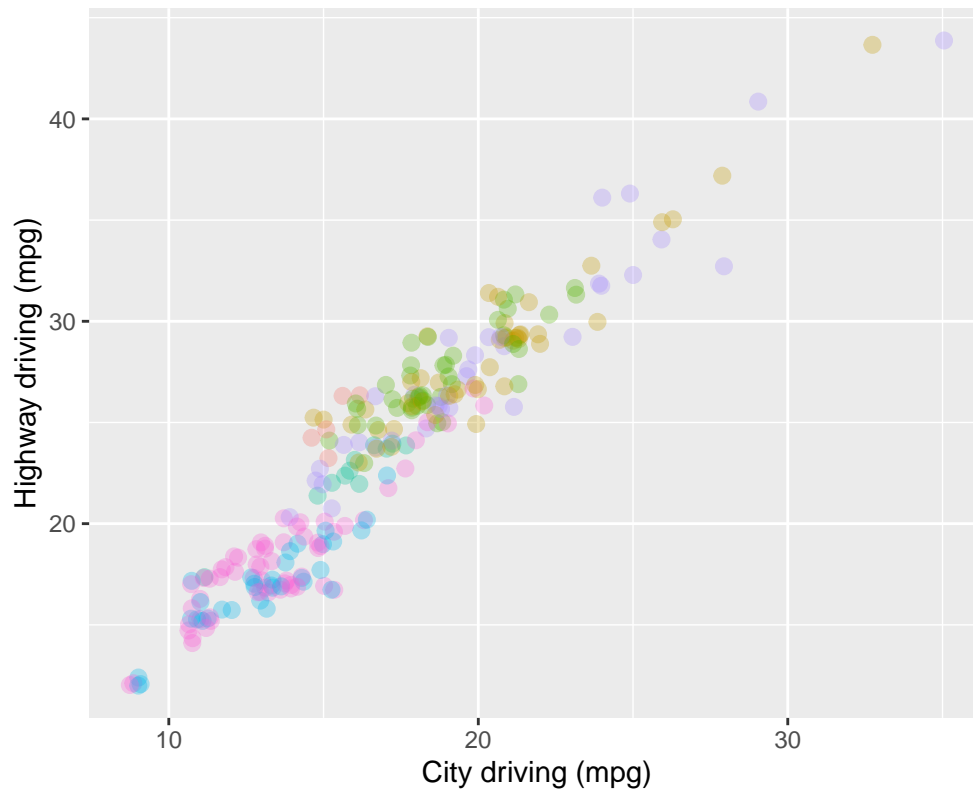
```
# Remove the axis labels with NULL
ggplot(mpg, aes(cty, hwy, color = class)) +
  geom_jitter(alpha = 0.3, size = 2.5) +
  xlab(NULL) +
  ylab(NULL) +
  ggtitle("City by Highway driving (mpg)")
```

City by Highway driving (mpg)



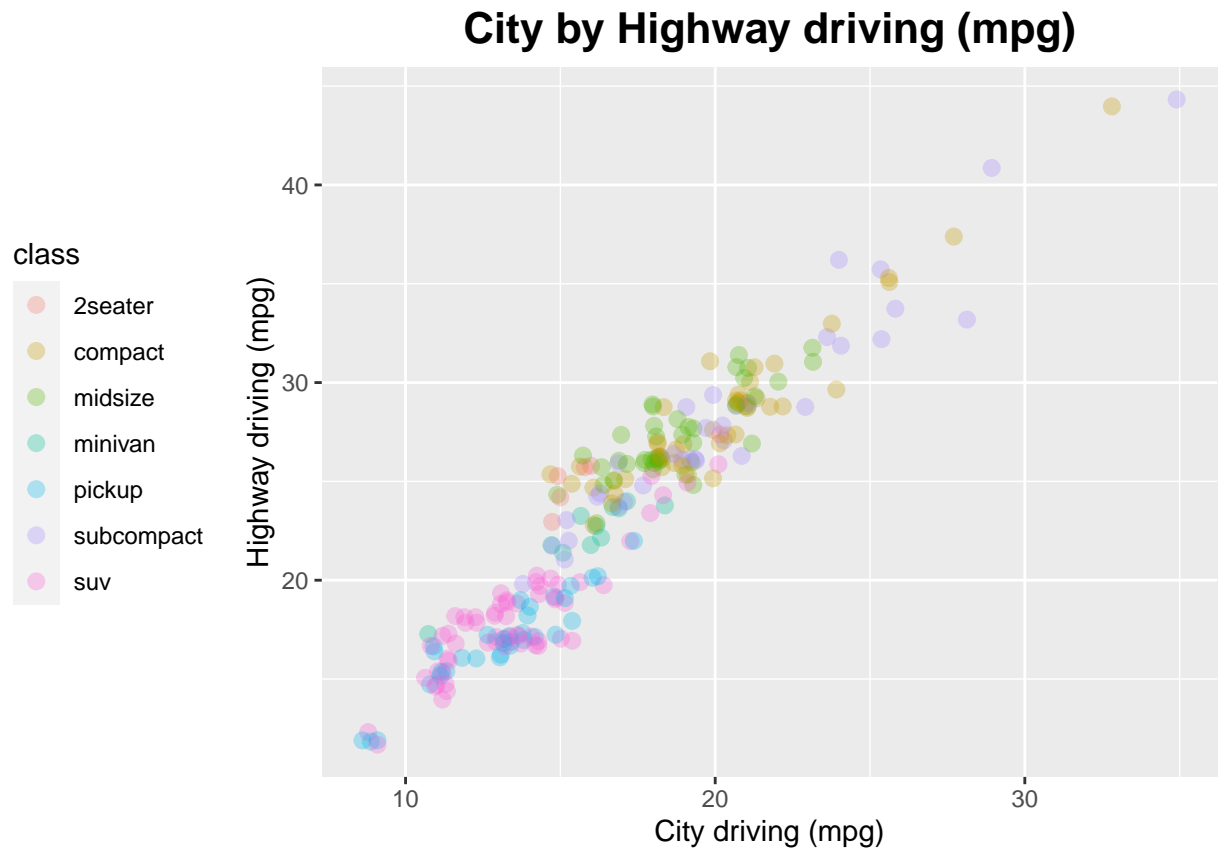
```
# Title
ggplot(mpg, aes(cty, hwy, color = class)) +
  geom_jitter(alpha = 0.3, size = 2.5) +
  xlab("City driving (mpg)") +
  ylab("Highway driving (mpg)") +
  ggtitle("City by Highway driving (mpg)") +
  theme(
    plot.title = element_text(size = 16, face = "bold", hjust = 0.5))
```

## City by Highway driving (mpg)

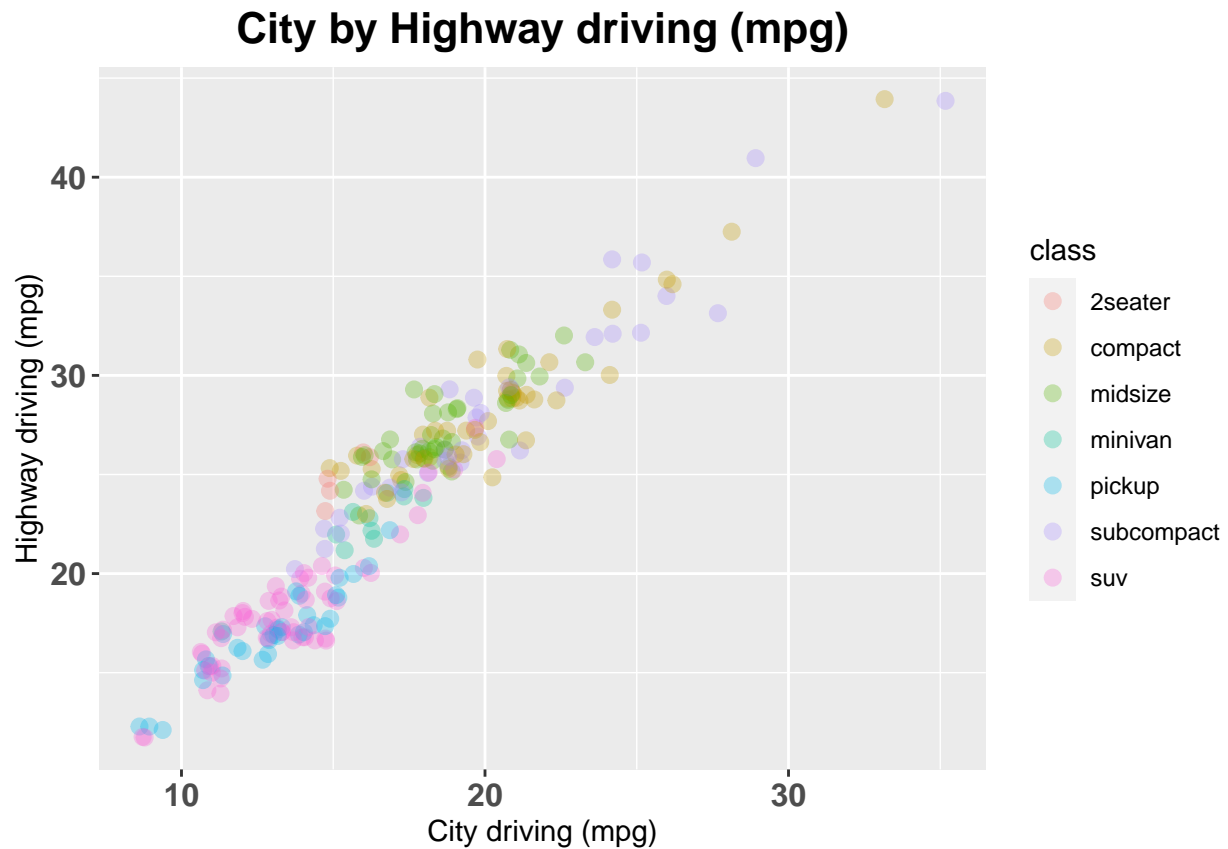


Editing title and axis using Theme

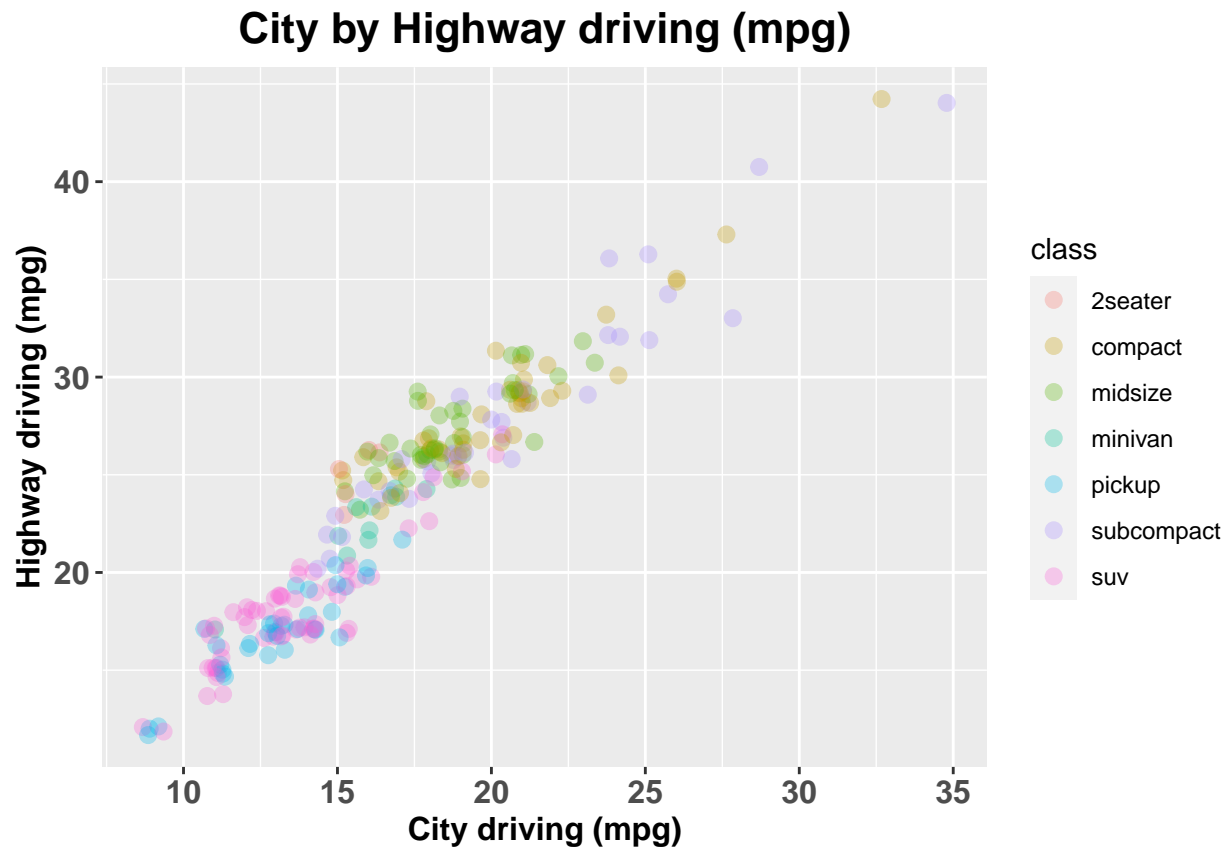
```
# Legend
ggplot(mpg, aes(cty, hwy, color = class)) +
  geom_jitter(alpha = 0.3, size = 2.5) +
  xlab("City driving (mpg)") +
  ylab("Highway driving (mpg)") +
  ggtitle("City by Highway driving (mpg)") +
  theme(
    plot.title = element_text(size = 16, face = "bold", hjust = 0.5),
    legend.position = "left")
```



```
# Axis
ggplot(mpg, aes(cty, hwy, color = class)) +
  geom_jitter(alpha = 0.3, size = 2.5) +
  xlab("City driving (mpg)") +
  ylab("Highway driving (mpg)") +
  ggtitle("City by Highway driving (mpg)") +
  theme(
    plot.title = element_text(size = 16, face = "bold", hjust = 0.5),
    legend.position = "right",
    axis.text.y = element_text(size = 12, face = "bold"),
    axis.text.x = element_text(size = 12, face = "bold"))
```



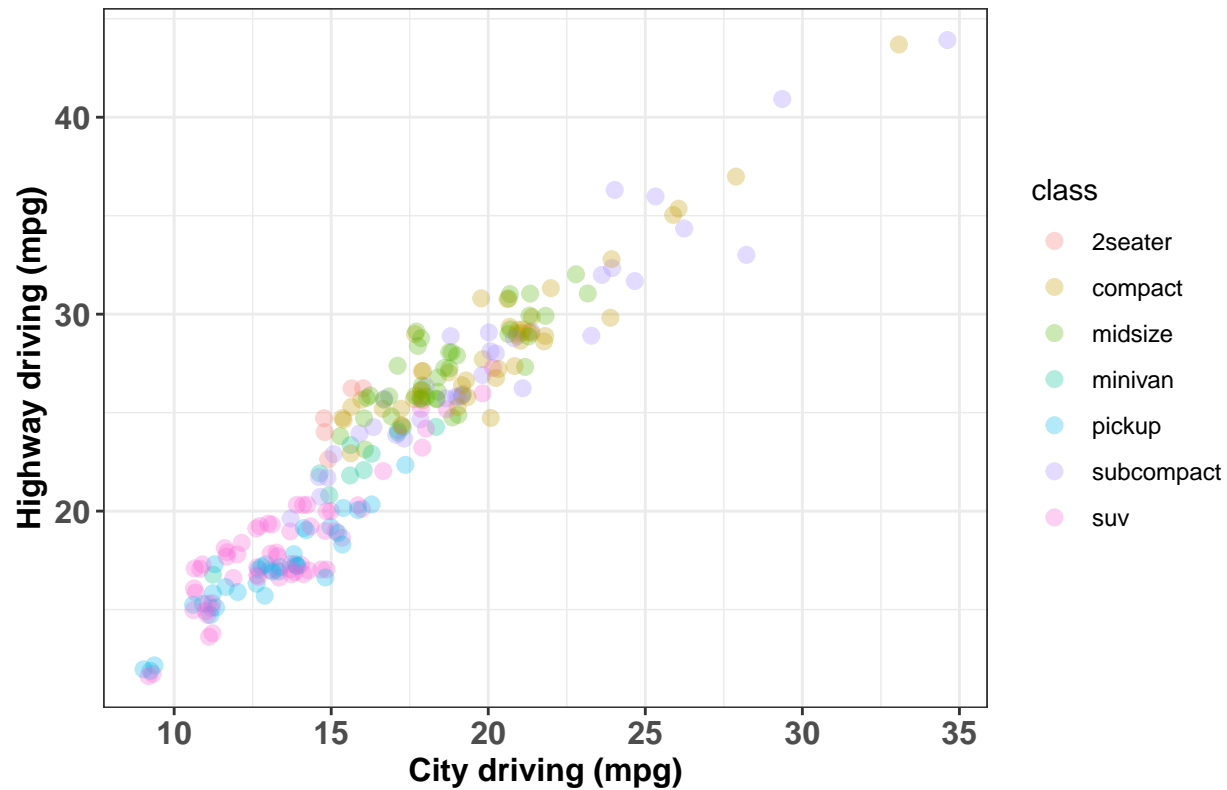
```
# Axis name
ggplot(mpg, aes(cty, hwy, color = class)) +
  geom_jitter(alpha = 0.3, size = 2.5) +
  xlab("City driving (mpg)") +
  ylab("Highway driving (mpg)") +
  ggtitle("City by Highway driving (mpg)") +
  theme(
    plot.title = element_text(size = 16, face = "bold", hjust = 0.5),
    legend.position = "right",
    axis.text.y = element_text(size = 12, face = "bold"),
    axis.text.x = element_text(size = 12, face = "bold"),
    axis.title.x = element_text(size = 12, face = "bold"),
    axis.title.y = element_text(size = 12, face = "bold"))
```



```
# Theme black and white
ggplot(mpg, aes(cty, hwy, color = class)) +
  geom_jitter(alpha = 0.3, size = 2.5) +
  xlab("City driving (mpg)") +
  ylab("Highway driving (mpg)") +
  ggtitle("City by Highway driving (mpg)") +
  theme_bw() +
  theme(
    plot.title = element_text(size = 16, face = "bold", hjust = 0.5),
    legend.position = "right",
    axis.text.y = element_text(size = 12, face = "bold"),
    axis.text.x = element_text(size = 12, face = "bold"),
    axis.title.x = element_text(size = 12, face = "bold"),
    axis.title.y = element_text(size = 12, face = "bold"))
```

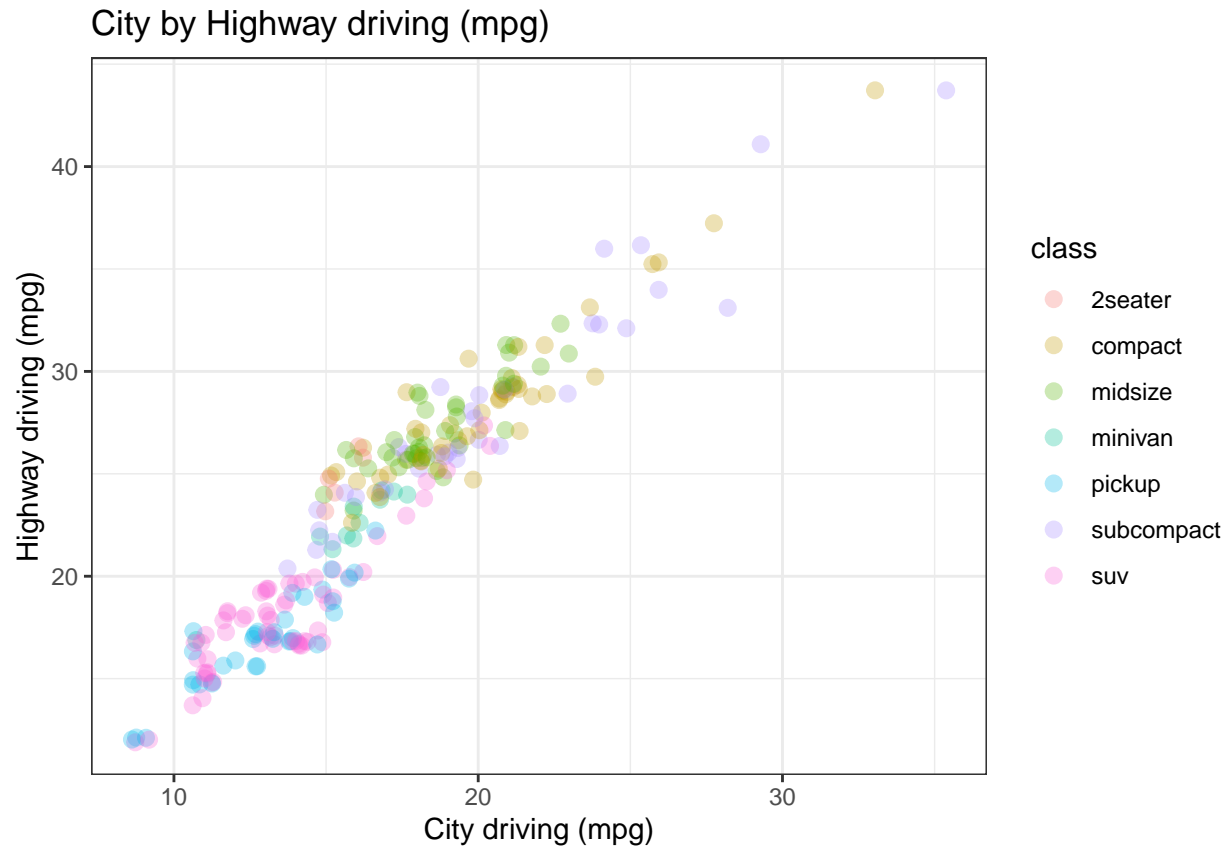


## City by Highway driving (mpg)

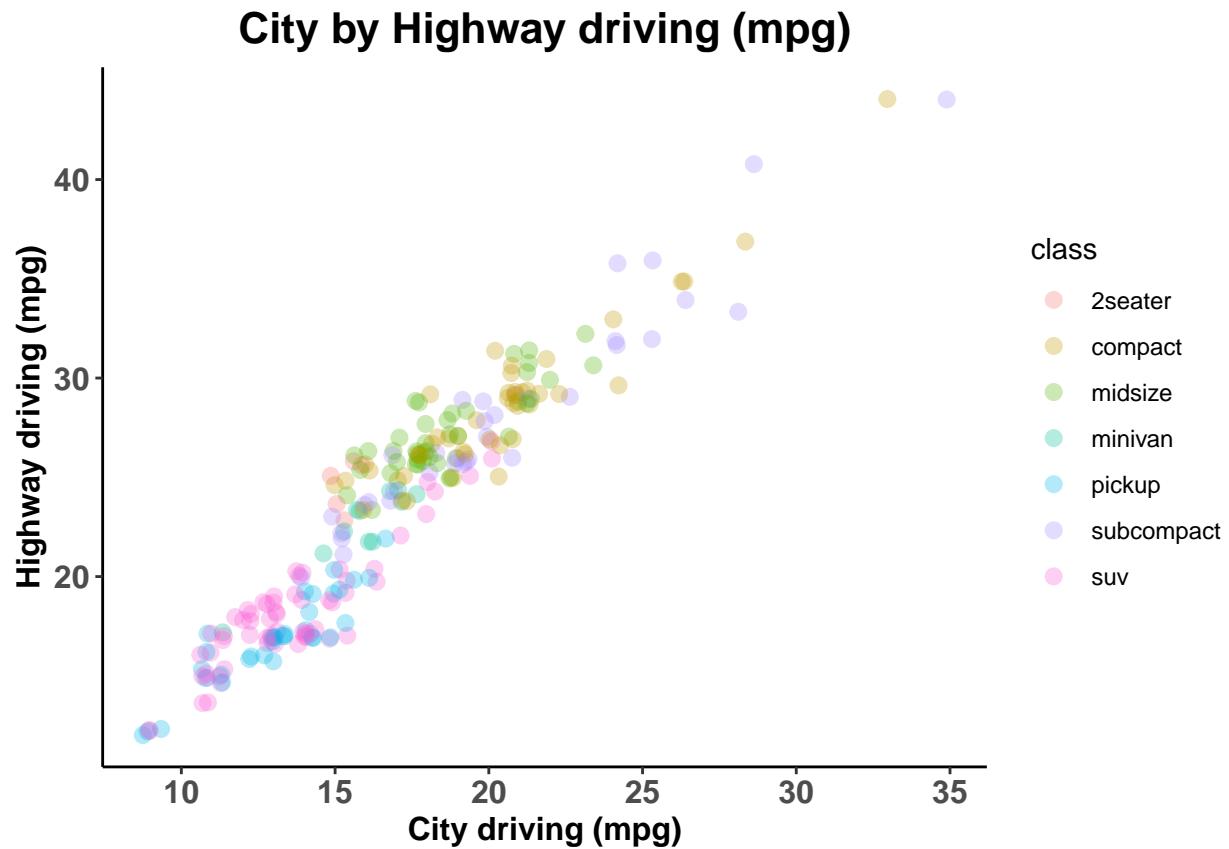


Overall theme

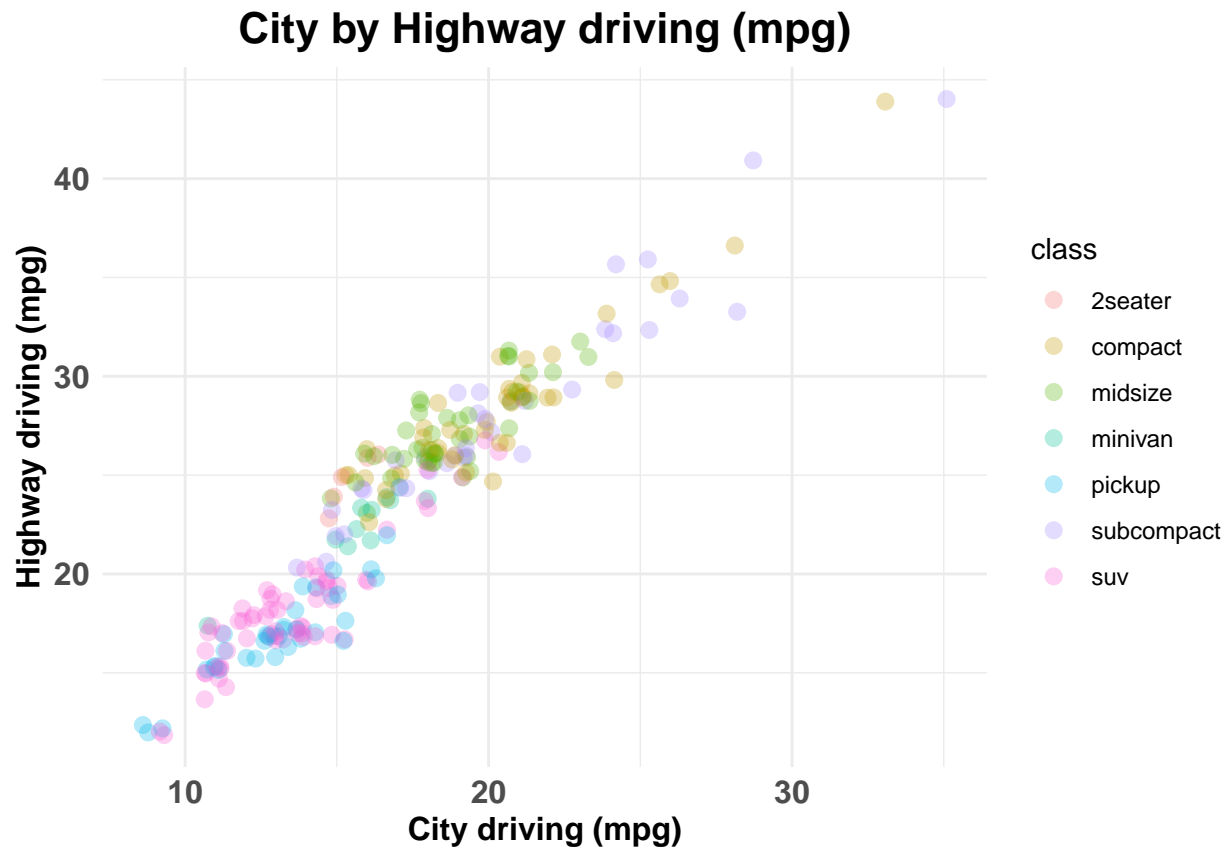
```
# Theme position matters!
ggplot(mpg, aes(cty, hwy, color = class)) +
  geom_jitter(alpha = 0.3, size = 2.5) +
  xlab("City driving (mpg)") +
  ylab("Highway driving (mpg)") +
  ggtitle("City by Highway driving (mpg)") +
  theme(
    plot.title = element_text(size = 16, face = "bold", hjust = 0.5),
    legend.position = "right",
    axis.text.y = element_text(size = 12, face = "bold"),
    axis.text.x = element_text(size = 12, face = "bold"),
    axis.title.x = element_text(size = 12, face = "bold"),
    axis.title.y = element_text(size = 12, face = "bold")) +
  theme_bw()
```



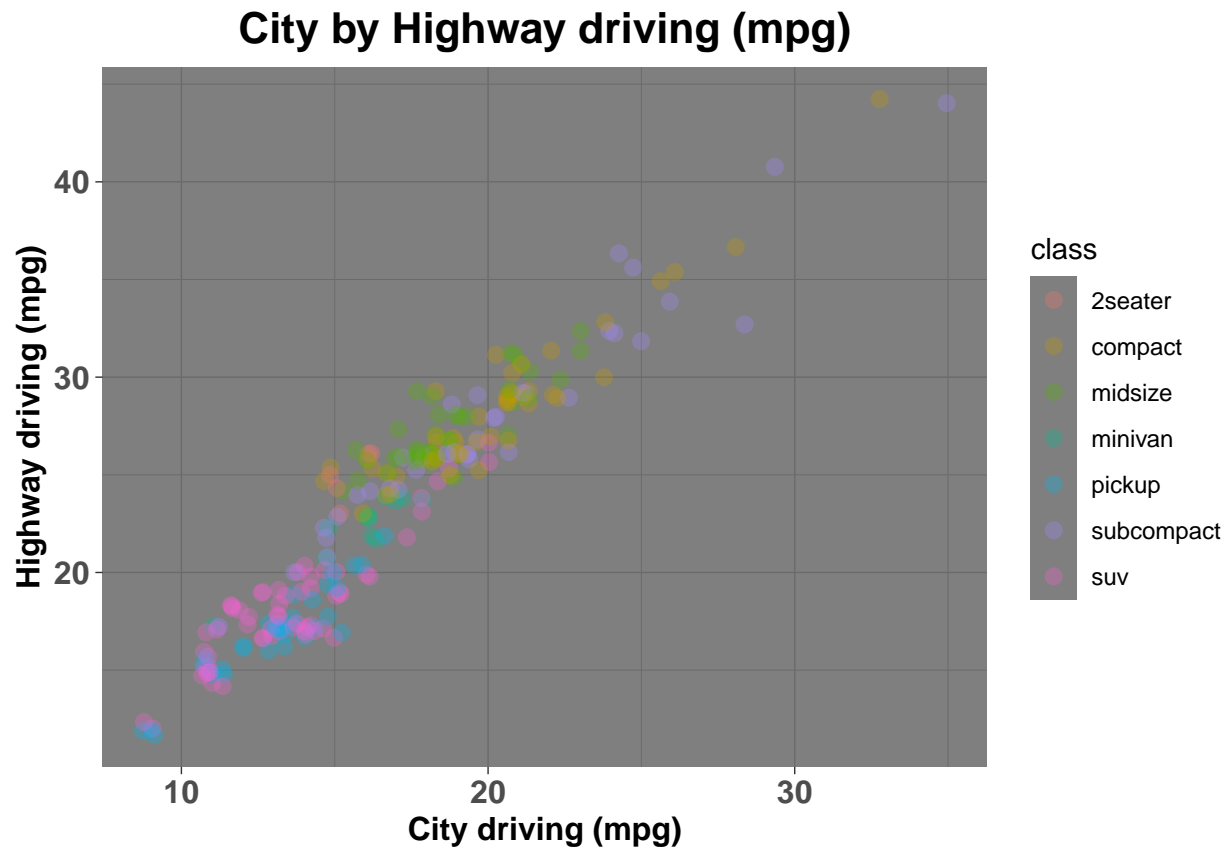
```
# Theme classic
ggplot(mpg, aes(cty, hwy, color = class)) +
  geom_jitter(alpha = 0.3, size = 2.5) +
  xlab("City driving (mpg)") +
  ylab("Highway driving (mpg)") +
  ggtitle("City by Highway driving (mpg)") +
  theme_classic() +
  theme(
    plot.title = element_text(size = 16, face = "bold", hjust = 0.5),
    legend.position = "right",
    axis.text.y = element_text(size = 12, face = "bold"),
    axis.text.x = element_text(size = 12, face = "bold"),
    axis.title.x = element_text(size = 12, face = "bold"),
    axis.title.y = element_text(size = 12, face = "bold"))
```



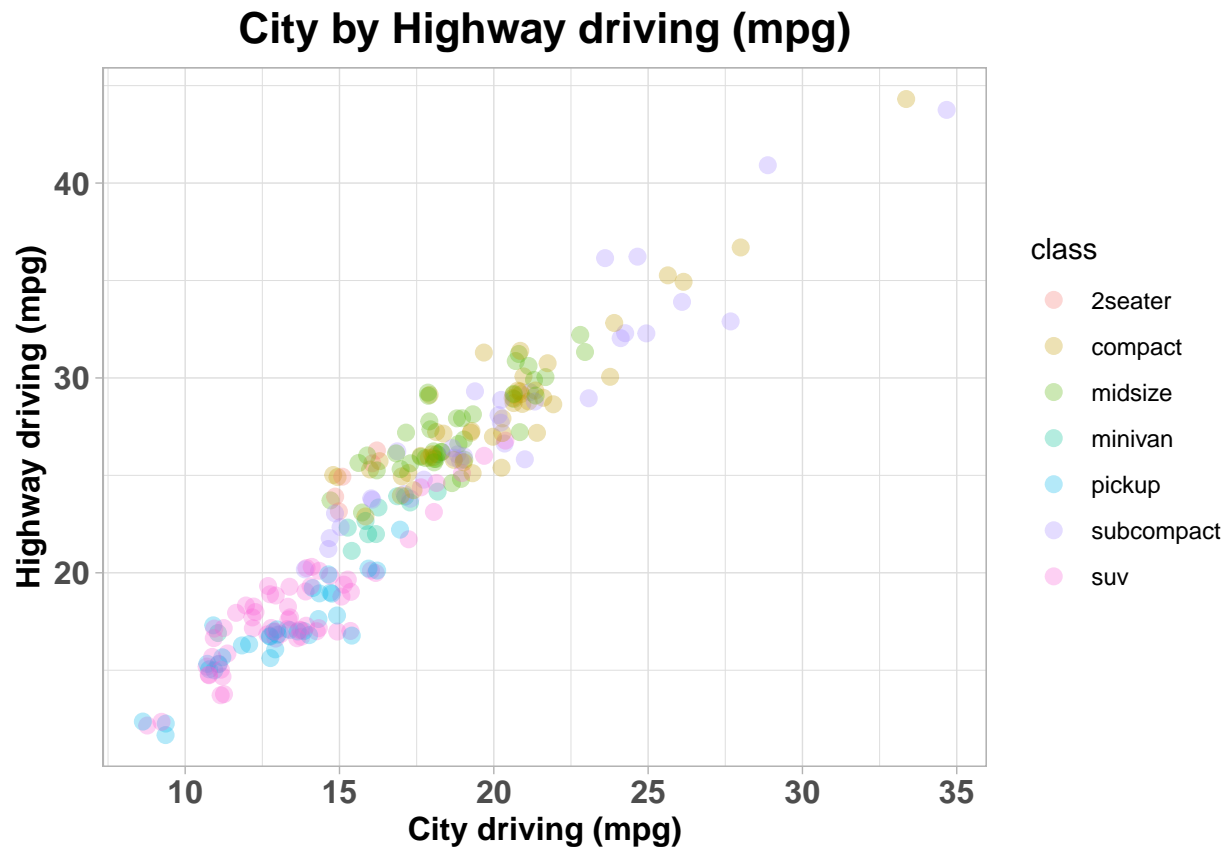
```
# Theme minimal
ggplot(mpg, aes(cty, hwy, color = class)) +
  geom_jitter(alpha = 0.3, size = 2.5) +
  xlab("City driving (mpg)") +
  ylab("Highway driving (mpg)") +
  ggtitle("City by Highway driving (mpg)") +
  theme_minimal() +
  theme(
    plot.title = element_text(size = 16, face = "bold", hjust = 0.5),
    legend.position = "right",
    axis.text.y = element_text(size = 12, face = "bold"),
    axis.text.x = element_text(size = 12, face = "bold"),
    axis.title.x = element_text(size = 12, face = "bold"),
    axis.title.y = element_text(size = 12, face = "bold"))
```



```
# Theme dark
ggplot(mpg, aes(cty, hwy, color = class)) +
  geom_jitter(alpha = 0.3, size = 2.5) +
  xlab("City driving (mpg)") +
  ylab("Highway driving (mpg)") +
  ggtitle("City by Highway driving (mpg)") +
  theme_dark() +
  theme(
    plot.title = element_text(size = 16, face = "bold", hjust = 0.5),
    legend.position = "right",
    axis.text.y = element_text(size = 12, face = "bold"),
    axis.text.x = element_text(size = 12, face = "bold"),
    axis.title.x = element_text(size = 12, face = "bold"),
    axis.title.y = element_text(size = 12, face = "bold"))
```



```
# Theme light
ggplot(mpg, aes(cty, hwy, color = class)) +
  geom_jitter(alpha = 0.3, size = 2.5) +
  xlab("City driving (mpg)") +
  ylab("Highway driving (mpg)") +
  ggtitle("City by Highway driving (mpg)") +
  theme_light() +
  theme(
    plot.title = element_text(size = 16, face = "bold", hjust = 0.5),
    legend.position = "right",
    axis.text.y = element_text(size = 12, face = "bold"),
    axis.text.x = element_text(size = 12, face = "bold"),
    axis.title.x = element_text(size = 12, face = "bold"),
    axis.title.y = element_text(size = 12, face = "bold"))
```



#### Appendix

```
library(datasets)
library(tidyverse)
library(knitr)
library(ggplot2)
mpg <- mpg
# What are 5 functions we could use to explore the mpg dataset?
str(mpg)

summary(mpg)

colnames(mpg)

?mpg

head(mpg)

# Which manufacturer has the most models in this dataset?
mpg %>%
  count(model) %>%
  arrange(n)

mpg %>%
  count(model) %>%
```

```

  arrange(desc(n))
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point()

# Fill in the following:
# Data: mpg
# Aesthetic: engine oil displacement to fuel economy
# Layers: point
# Plus sign location: end of first line
ggplot(mpg, aes(displ, hwy)) +
  geom_point()

# What conclusions can you make from this graph?
# Do these data look similar when comparing engine displacement to city miles per gallon?
ggplot(mpg, aes(displ, cty)) +
  geom_point()

# We can more easily compare the highway and city miles per gallon using the gridExtra package.
library(gridExtra)

p1 <- ggplot(mpg, aes(displ, hwy)) +
  geom_point()

p2 <- ggplot(mpg, aes(displ, cty)) +
  geom_point()

grid.arrange(p1, p2, ncol = 2)
ggplot(mpg, aes(displ, hwy)) +
  geom_point(aes(color = "blue"))

# Value "blue" is scaled to red color
ggplot(mpg, aes(displ, hwy)) +
  geom_point(color = "blue")

# Points are given color
ggplot(mpg, aes(displ, hwy, color = class)) +
  geom_point()

# Unique color based on class of vehicle

# What additional information does the color add? What patterns do you see?
ggplot(mpg, aes(displ, hwy)) +
  geom_point() +
  facet_wrap(~ class)

# Read the documentation for facet_wrap. What arguments can you use to control how many rows and columns?
?facet_wrap

ggplot(mpg, aes(displ, hwy)) +
  geom_point() +
  facet_wrap(~ class, nrow = 1)

```

*#What does the scales argument to facet\_wrap() do? When might you use it?*

```
ggplot(mpg, aes(displ, hwy)) +  
  geom_point() +  
  facet_wrap(~ class, scales = "free")  
ggplot(mpg, aes(displ, hwy)) +  
  geom_point() +  
  geom_smooth()
```

*# Smoothed line shows dominant pattern and level of uncertainty  
# Method = "loess" is default for small n (>1,000 points)*

```
ggplot(mpg, aes(displ, hwy)) +  
  geom_point() +  
  geom_smooth(span = 0.2)
```

```
ggplot(mpg, aes(displ, hwy)) +  
  geom_point() +  
  geom_smooth(span = 1)
```

*# Span control "wiggleness of line"*

```
ggplot(mpg, aes(displ, hwy)) +  
  geom_point() +  
  geom_smooth(method = "lm")
```

*# Linear model gives the line of best fit*

```
ggplot(mpg, aes(hwy)) +  
  geom_histogram()
```

```
ggplot(mpg, aes(hwy)) +  
  geom_freqpoly()
```

*# Binwidth controls the width of bins (probably NOT ideal for your data, try experimenting)*

```
ggplot(mpg, aes(hwy)) +  
  geom_freqpoly(binwidth = 1)
```

```
ggplot(mpg, aes(hwy)) +  
  geom_freqpoly(binwidth = 2.5)
```

```
p3 <- ggplot(mpg, aes(hwy)) +  
  geom_freqpoly(binwidth = 1)
```

```
p4 <- ggplot(mpg, aes(hwy)) +  
  geom_freqpoly(binwidth = 2.5)
```

```
grid.arrange(p3, p4, ncol = 2)
```

```
ggplot(mpg, aes(manufacturer)) +  
  geom_bar()  
ggplot(mpg, aes(drv, hwy)) +  
  geom_boxplot()
```



```

ggplot(mpg, aes(drv, hwy)) +
  geom_boxplot() +
  geom_point()

ggplot(mpg, aes(drv, hwy)) +
  geom_boxplot() +
  geom_jitter()
ggplot(mpg, aes(drv, hwy)) +
  geom_boxplot() +
  geom_jitter(alpha = 0.2)

ggplot(mpg, aes(drv, hwy)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(alpha = 0.3)

ggplot(mpg, aes(drv, hwy)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(alpha = 0.3, size = 2.5)

ggplot(mpg, aes(drv, hwy)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(alpha = 0.3, size = 2.5, width = 0.1)
ggplot(mpg, aes(drv, hwy)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(width = 0.25)

ggplot(mpg, aes(drv, hwy)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(width = 0.25) +
  xlim("f", "r")

ggplot(mpg, aes(drv, hwy)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(width = 0.25) +
  xlim("f", "r")+
  ylim(20, 30)
#> Warning: Removed 138 rows containing missing values (`geom_point()`).

ggplot(mpg, aes(cty, hwy, color = class)) +
  geom_jitter(alpha = 0.3, size = 2.5)

# Change axis titles
ggplot(mpg, aes(cty, hwy, color = class)) +
  geom_jitter(alpha = 0.3, size = 2.5) +
  xlab("City driving (mpg)") +
  ylab("Highway driving (mpg)")

ggplot(mpg, aes(cty, hwy, color = class)) +
  geom_jitter(alpha = 0.3, size = 2.5) +
  xlab("City driving (mpg)") +
  ylab("Highway driving (mpg)") +
  ggtitle("City by Highway driving (mpg)")

```

```

# Remove the axis labels with NULL
ggplot(mpg, aes(cty, hwy, color = class)) +
  geom_jitter(alpha = 0.3, size = 2.5) +
  xlab(NULL) +
  ylab(NULL) +
  ggtitle("City by Highway driving (mpg)")

# Title
ggplot(mpg, aes(cty, hwy, color = class)) +
  geom_jitter(alpha = 0.3, size = 2.5) +
  xlab("City driving (mpg)") +
  ylab("Highway driving (mpg)") +
  ggtitle("City by Highway driving (mpg)") +
  theme(
    plot.title = element_text(size = 16, face = "bold", hjust = 0.5))

# Legend
ggplot(mpg, aes(cty, hwy, color = class)) +
  geom_jitter(alpha = 0.3, size = 2.5) +
  xlab("City driving (mpg)") +
  ylab("Highway driving (mpg)") +
  ggtitle("City by Highway driving (mpg)") +
  theme(
    plot.title = element_text(size = 16, face = "bold", hjust = 0.5),
    legend.position = "left")

# Axis
ggplot(mpg, aes(cty, hwy, color = class)) +
  geom_jitter(alpha = 0.3, size = 2.5) +
  xlab("City driving (mpg)") +
  ylab("Highway driving (mpg)") +
  ggtitle("City by Highway driving (mpg)") +
  theme(
    plot.title = element_text(size = 16, face = "bold", hjust = 0.5),
    legend.position = "right",
    axis.text.y = element_text(size = 12, face = "bold"),
    axis.text.x = element_text(size = 12, face = "bold"))

# Axis name
ggplot(mpg, aes(cty, hwy, color = class)) +
  geom_jitter(alpha = 0.3, size = 2.5) +
  xlab("City driving (mpg)") +
  ylab("Highway driving (mpg)") +
  ggtitle("City by Highway driving (mpg)") +
  theme(
    plot.title = element_text(size = 16, face = "bold", hjust = 0.5),
    legend.position = "right",
    axis.text.y = element_text(size = 12, face = "bold"),
    axis.text.x = element_text(size = 12, face = "bold"),
    axis.title.x = element_text(size = 12, face = "bold"),
    axis.title.y = element_text(size = 12, face = "bold"))

# Theme black and white
ggplot(mpg, aes(cty, hwy, color = class)) +

```

```

geom_jitter(alpha = 0.3, size = 2.5) +
xlab("City driving (mpg)") +
ylab("Highway driving (mpg)") +
ggtitle("City by Highway driving (mpg)") +
theme_bw() +
theme(
  plot.title = element_text(size = 16, face = "bold", hjust = 0.5),
  legend.position = "right",
  axis.text.y = element_text(size = 12, face = "bold"),
  axis.text.x = element_text(size = 12, face = "bold"),
  axis.title.x = element_text(size = 12, face = "bold"),
  axis.title.y = element_text(size = 12, face = "bold"))

# Theme position matters!
ggplot(mpg, aes(cty, hwy, color = class)) +
geom_jitter(alpha = 0.3, size = 2.5) +
xlab("City driving (mpg)") +
ylab("Highway driving (mpg)") +
ggtitle("City by Highway driving (mpg)") +
theme(
  plot.title = element_text(size = 16, face = "bold", hjust = 0.5),
  legend.position = "right",
  axis.text.y = element_text(size = 12, face = "bold"),
  axis.text.x = element_text(size = 12, face = "bold"),
  axis.title.x = element_text(size = 12, face = "bold"),
  axis.title.y = element_text(size = 12, face = "bold")) +
theme_bw()

# Theme classic
ggplot(mpg, aes(cty, hwy, color = class)) +
geom_jitter(alpha = 0.3, size = 2.5) +
xlab("City driving (mpg)") +
ylab("Highway driving (mpg)") +
ggtitle("City by Highway driving (mpg)") +
theme_classic() +
theme(
  plot.title = element_text(size = 16, face = "bold", hjust = 0.5),
  legend.position = "right",
  axis.text.y = element_text(size = 12, face = "bold"),
  axis.text.x = element_text(size = 12, face = "bold"),
  axis.title.x = element_text(size = 12, face = "bold"),
  axis.title.y = element_text(size = 12, face = "bold"))

# Theme minimal
ggplot(mpg, aes(cty, hwy, color = class)) +
geom_jitter(alpha = 0.3, size = 2.5) +
xlab("City driving (mpg)") +
ylab("Highway driving (mpg)") +
ggtitle("City by Highway driving (mpg)") +
theme_minimal() +
theme(
  plot.title = element_text(size = 16, face = "bold", hjust = 0.5),
  legend.position = "right",

```

```

axis.text.y = element_text(size = 12, face = "bold"),
axis.text.x = element_text(size = 12, face = "bold"),
axis.title.x = element_text(size = 12, face = "bold"),
axis.title.y = element_text(size = 12, face = "bold"))

# Theme dark
ggplot(mpg, aes(cty, hwy, color = class)) +
  geom_jitter(alpha = 0.3, size = 2.5) +
  xlab("City driving (mpg)") +
  ylab("Highway driving (mpg)") +
  ggtitle("City by Highway driving (mpg)") +
  theme_dark() +
  theme(
    plot.title = element_text(size = 16, face = "bold", hjust = 0.5),
    legend.position = "right",
    axis.text.y = element_text(size = 12, face = "bold"),
    axis.text.x = element_text(size = 12, face = "bold"),
    axis.title.x = element_text(size = 12, face = "bold"),
    axis.title.y = element_text(size = 12, face = "bold"))

# Theme light
ggplot(mpg, aes(cty, hwy, color = class)) +
  geom_jitter(alpha = 0.3, size = 2.5) +
  xlab("City driving (mpg)") +
  ylab("Highway driving (mpg)") +
  ggtitle("City by Highway driving (mpg)") +
  theme_light() +
  theme(
    plot.title = element_text(size = 16, face = "bold", hjust = 0.5),
    legend.position = "right",
    axis.text.y = element_text(size = 12, face = "bold"),
    axis.text.x = element_text(size = 12, face = "bold"),
    axis.title.x = element_text(size = 12, face = "bold"),
    axis.title.y = element_text(size = 12, face = "bold"))

```