Graphing Using ggplot Part-1

```
library(ggplot2)
library(dplyr)
```

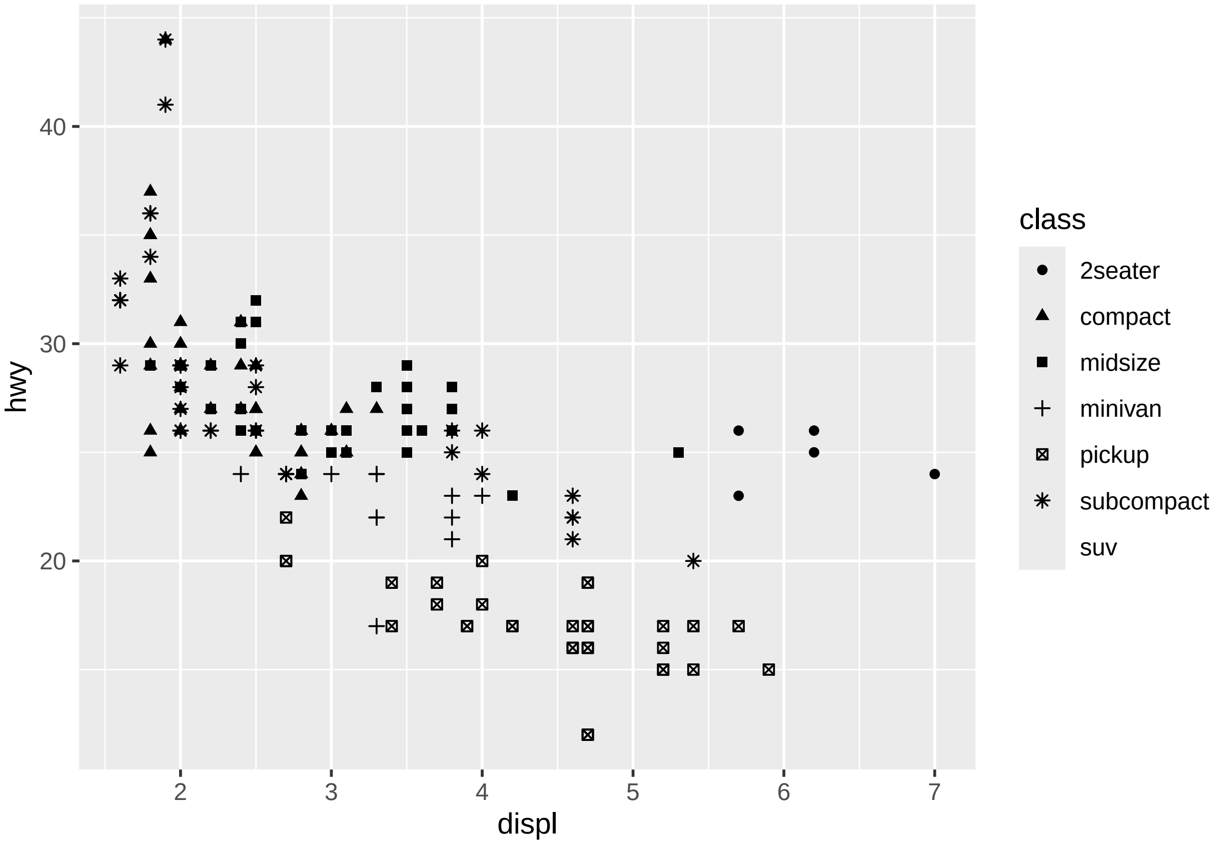
```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
# Load the mpg dataset
data('mpg')
mpgData = mpg

# Print the first five rows (or samples) in the data frame
head(mpgData, 5)
```

```
## # A tibble: 5 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~
```

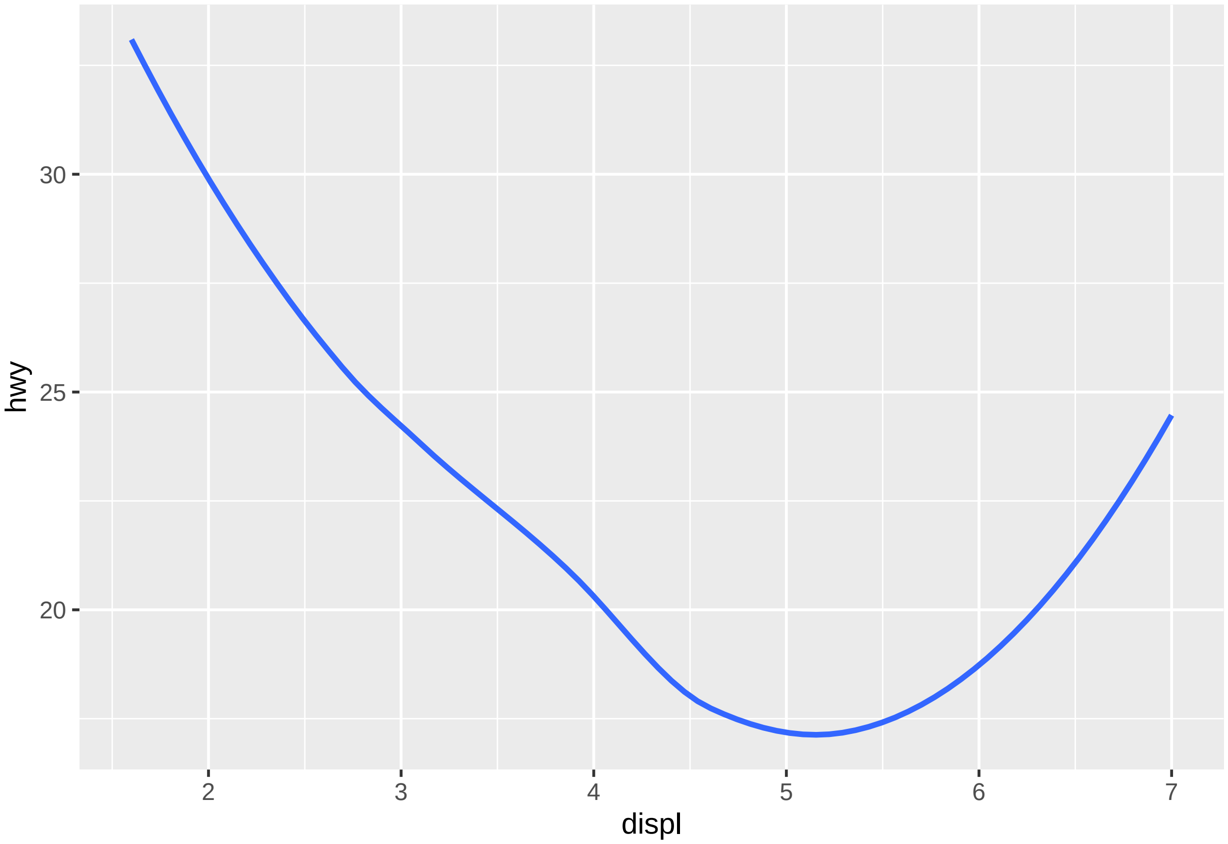
```
# Plot a scatter plot of mileage w.r.t. displacement
p1 = ggplot(data = mpgData) +
  geom_point(mapping = aes(x = displ, y = hwy, shape = class))
p1
```

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

## Warning: Removed 62 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



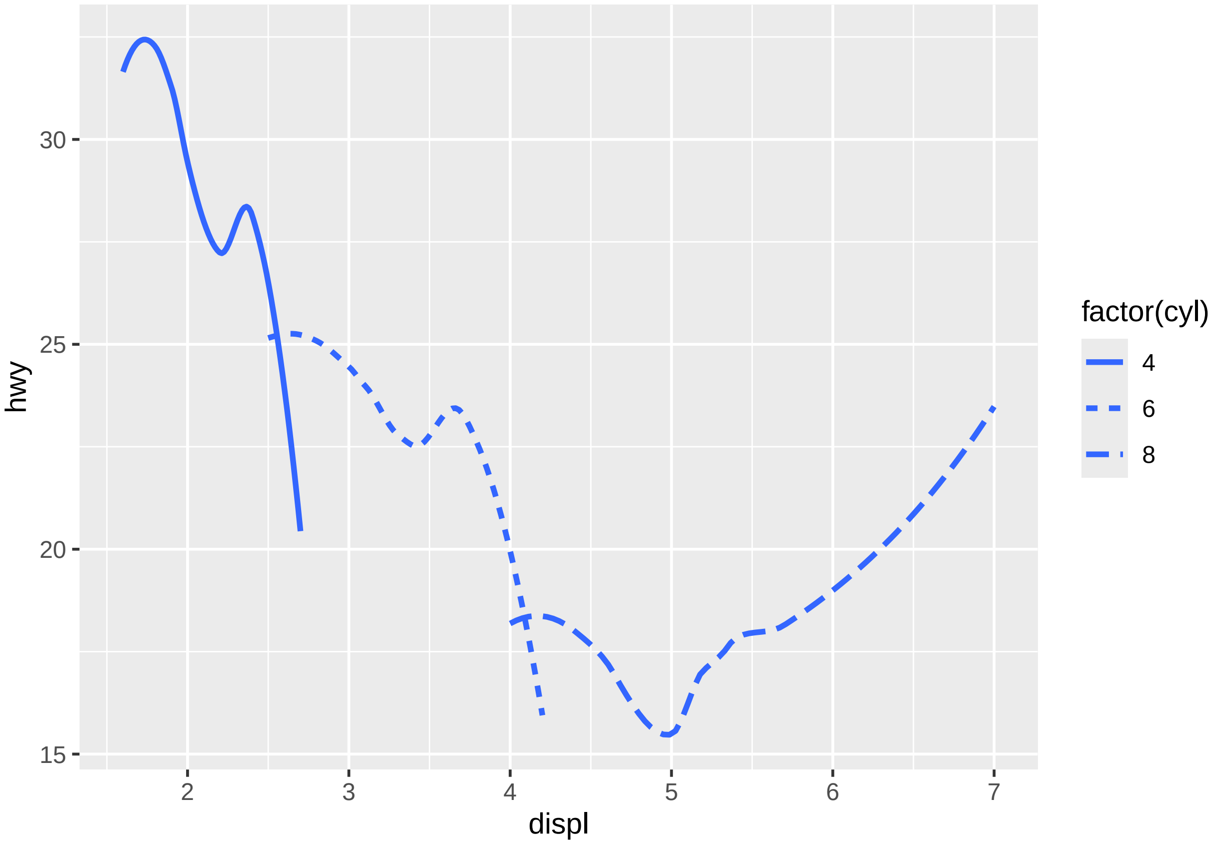
```r
# Plot a smooth lineplot of mileage w.r.t. displacement
p2 = ggplot(data = mpgData) +
  geom_smooth(mapping = aes(x = displ, y = hwy), se = FALSE)
p2
```

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



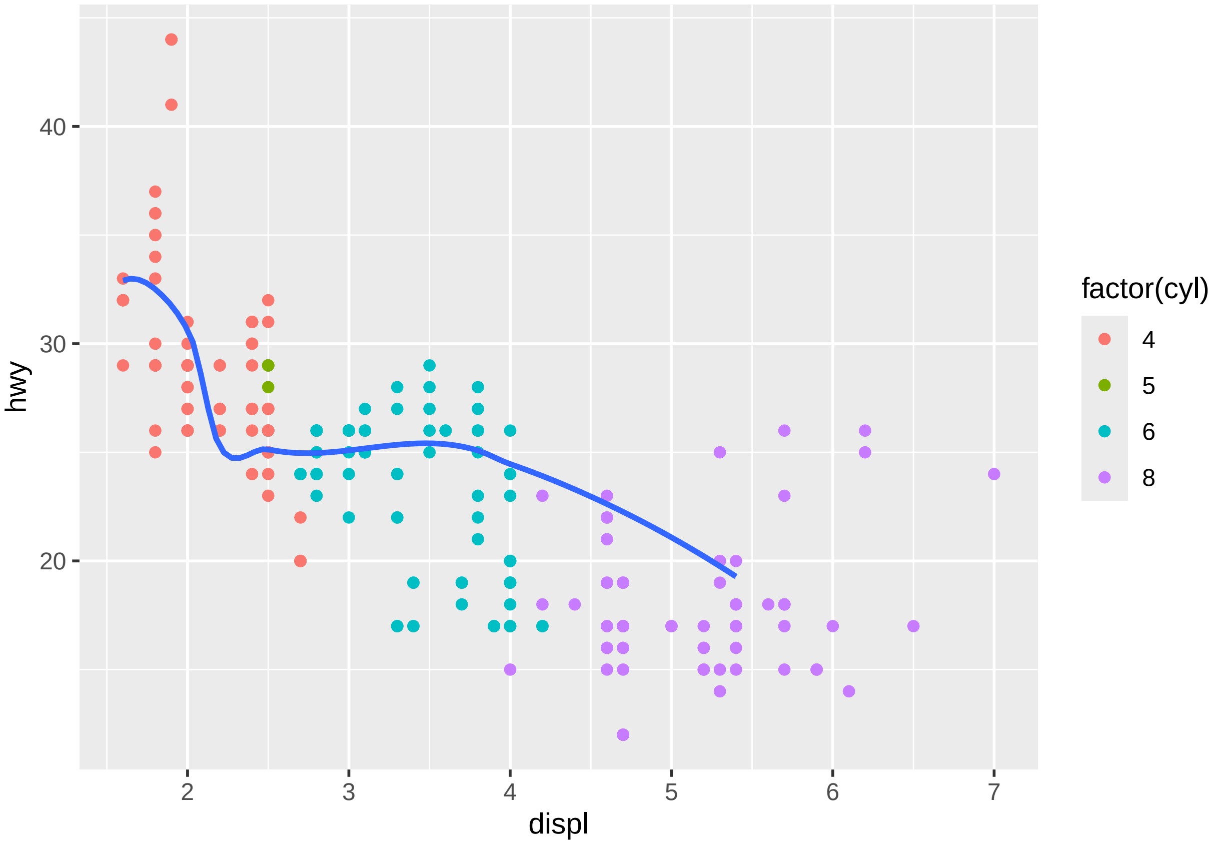
```r
# Plot a smooth lineplot of mileage w.r.t. displacement for each drv type
p3 = ggplot(data = mpgData) +
  geom_smooth(mapping = aes(x = displ, y = hwy, linetype = factor(cyl)), se = FALSE)
p3
```

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

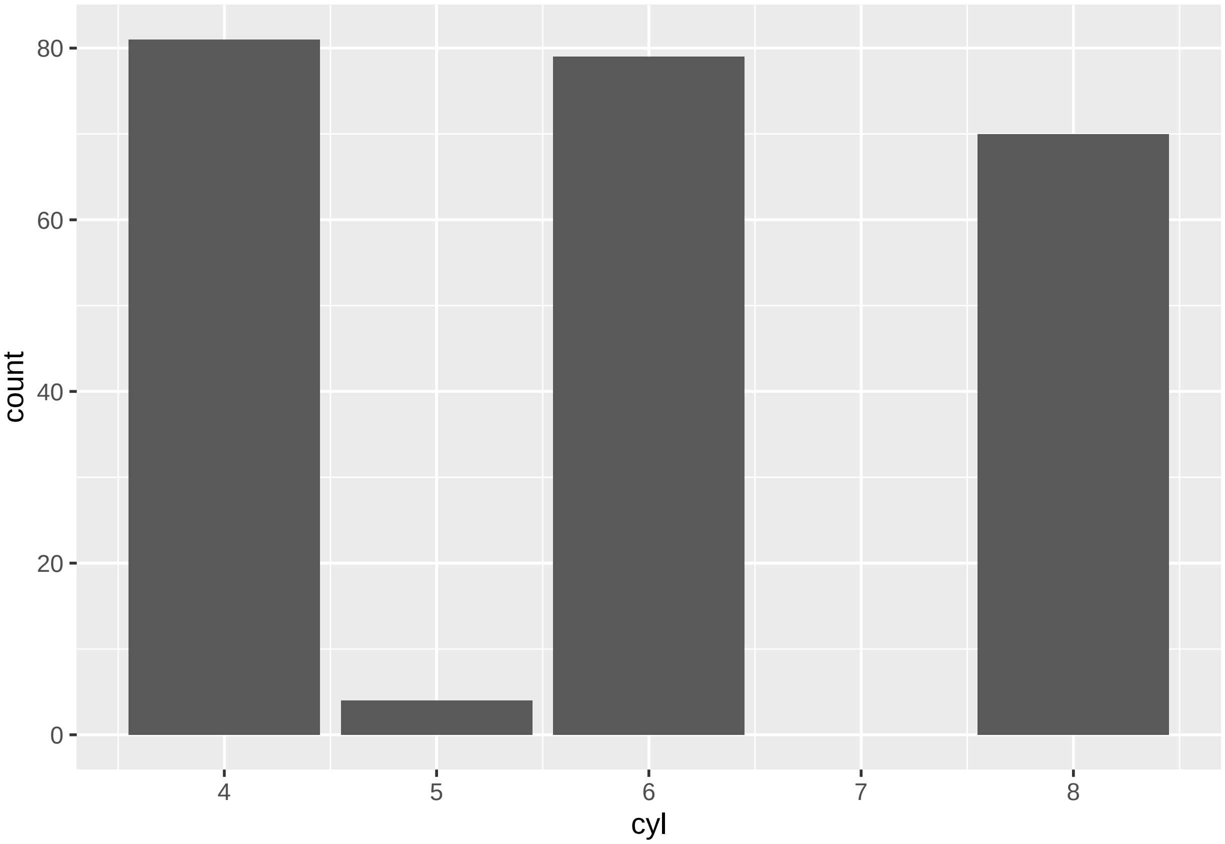


```
# Specify data for layers individually
# Plot mileage w.r.t. displacement for all cars but add a smooth
# line only for subcompact cars by filtering
p4 = ggplot(data = mpgData, mapping = aes(x = displ, y = hwy)) +
  geom_point(mapping = aes(color = factor(cyl))) +
  geom_smooth(data = filter(mpgData, class == 'subcompact'), se = FALSE)
p4
```

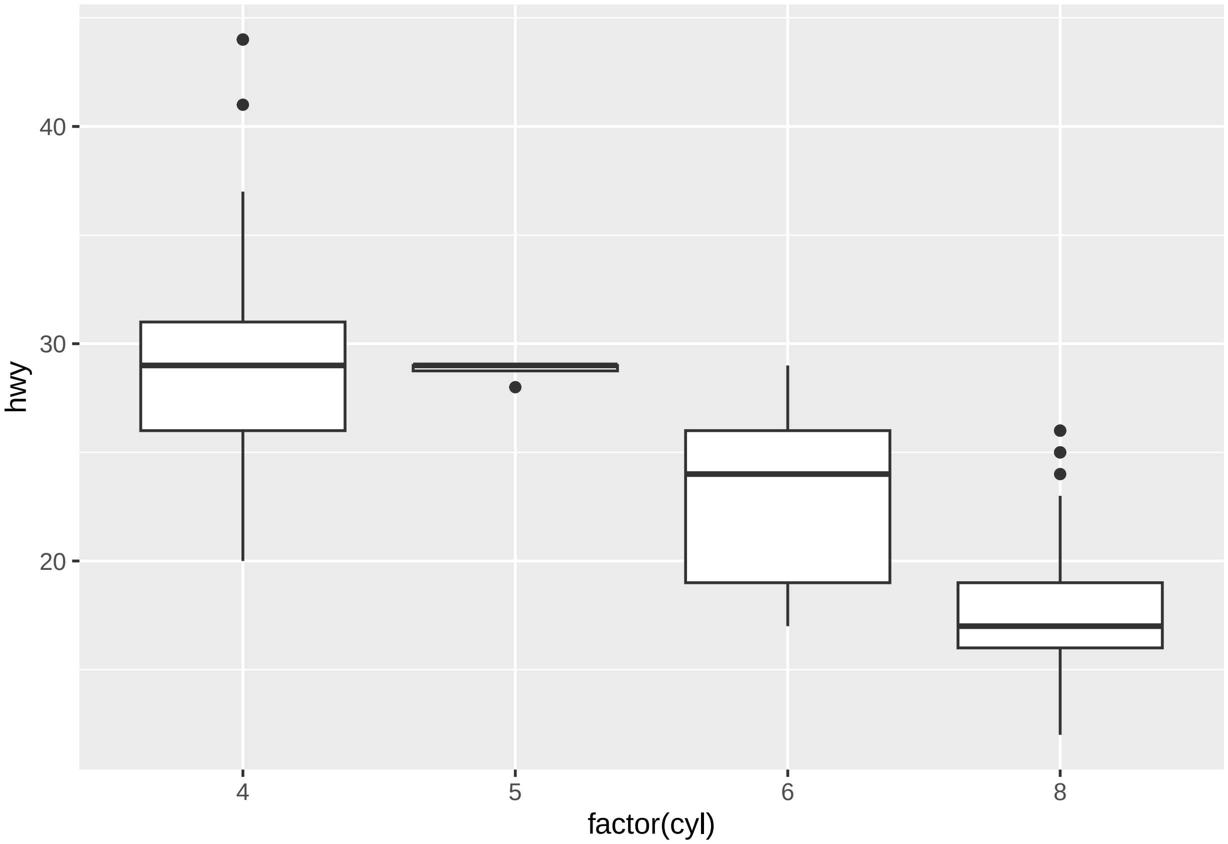
```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



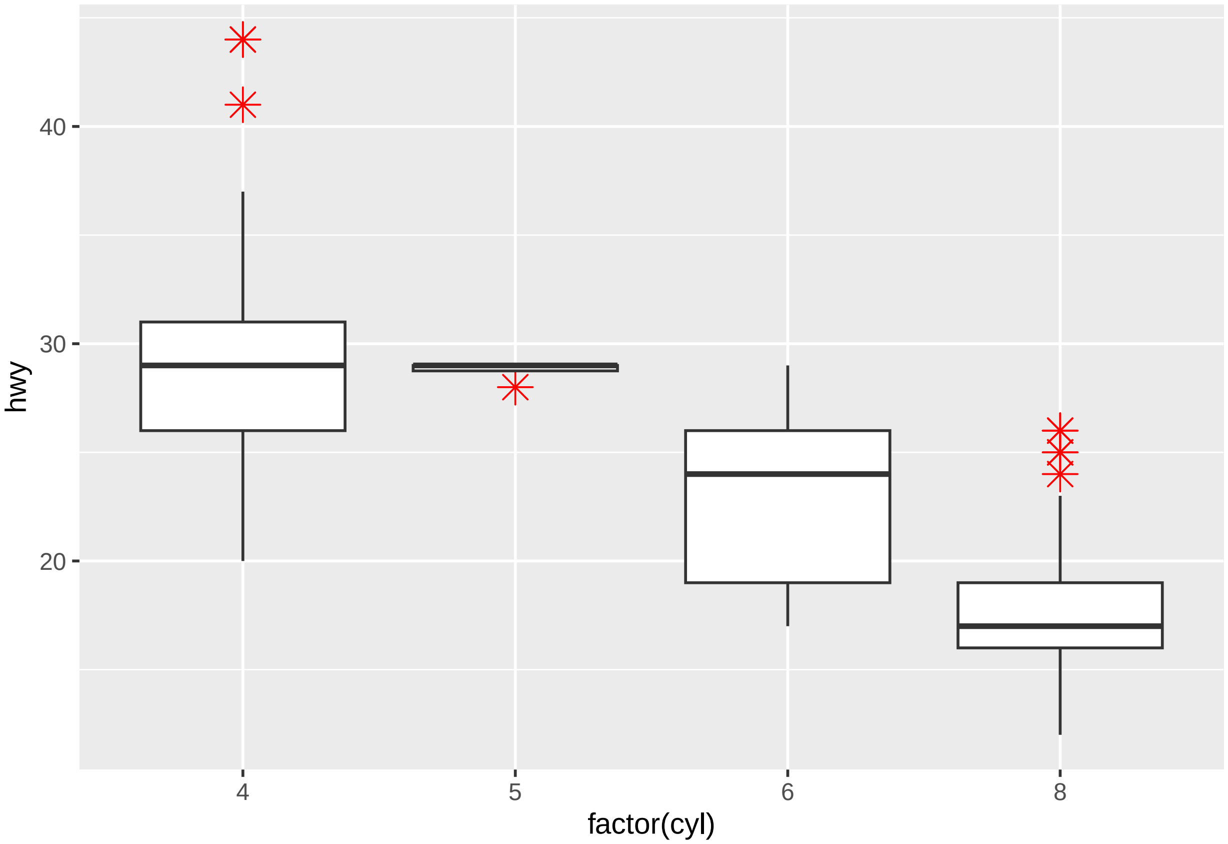
```
# Add a bar chart w.r.t. number of cylinders
p5 = ggplot(data = mpgData) +
  geom_bar(mapping = aes(x = cyl))
p5
```



```r
# Add a box plot w.r.t. number of cylinders and mileage
# Notched box plot
p6 = ggplot(data = mpgData) +
    geom_boxplot(mapping = aes(x = factor(cyl), y = hwy))
p6
```



```
# # Change outlier color, shape and size
p6 = ggplot(data = mpgData) +
   geom_boxplot(aes(x = factor(cyl), y = hwy), outlier.colour="red", outlier.shape=8, outlier.size=4)
 p6
```

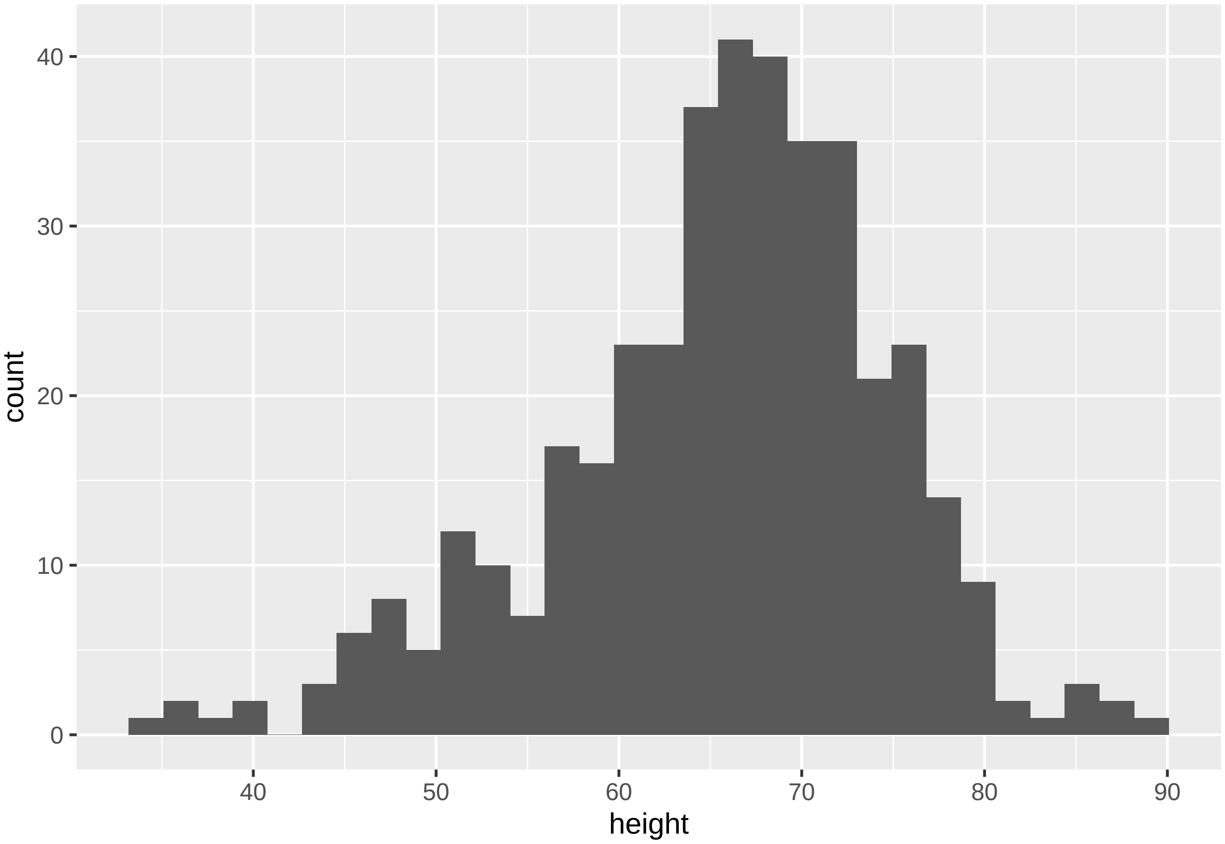


```
# Simulate a data set and store in a data frame
df = data.frame(
  gender = factor(rep(c("F", "M"), each = 200)),
  height = round(c(rnorm(200, mean = 60, sd = 10), rnorm(200, mean = 70, sd = 6)))
  )
head(df, 5)
```

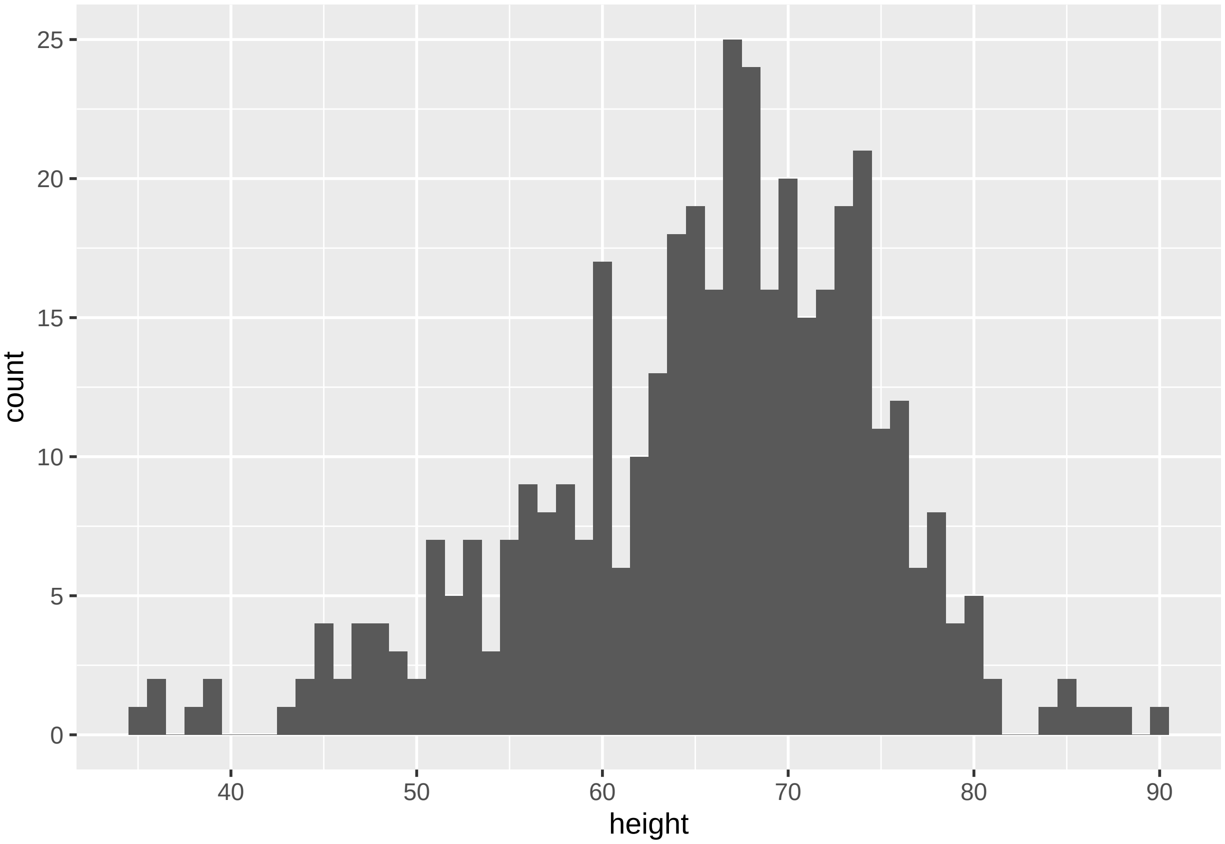
```
##   gender height
## 1      F     63
## 2      F     66
## 3      F     56
## 4      F     73
## 5      F     64
```

```
# Plot a basic histogram
ggplot(df, aes(x = height)) +
  geom_histogram(mapping = aes(x = height))
```

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

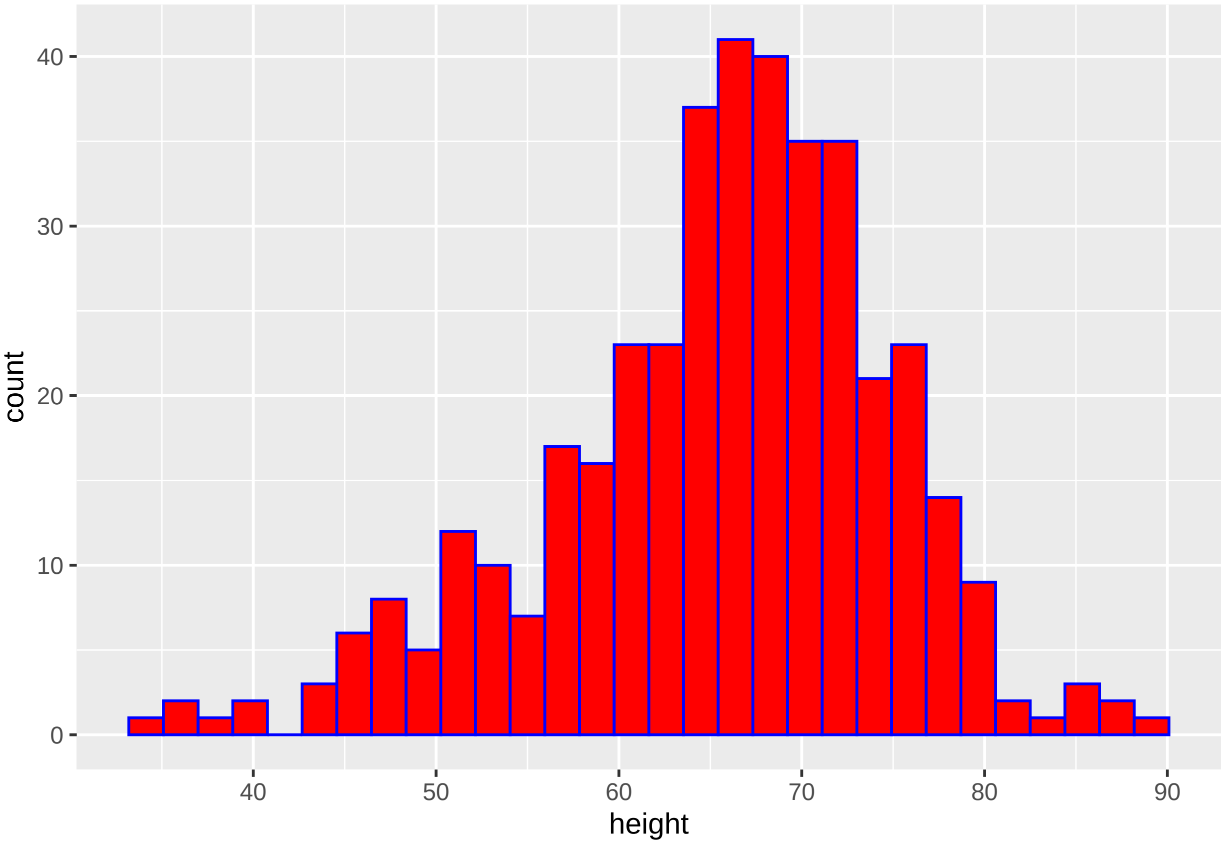


```
# # Change the width of bins
ggplot(df, aes(x = height)) +
   geom_histogram(binwidth = 1)
```



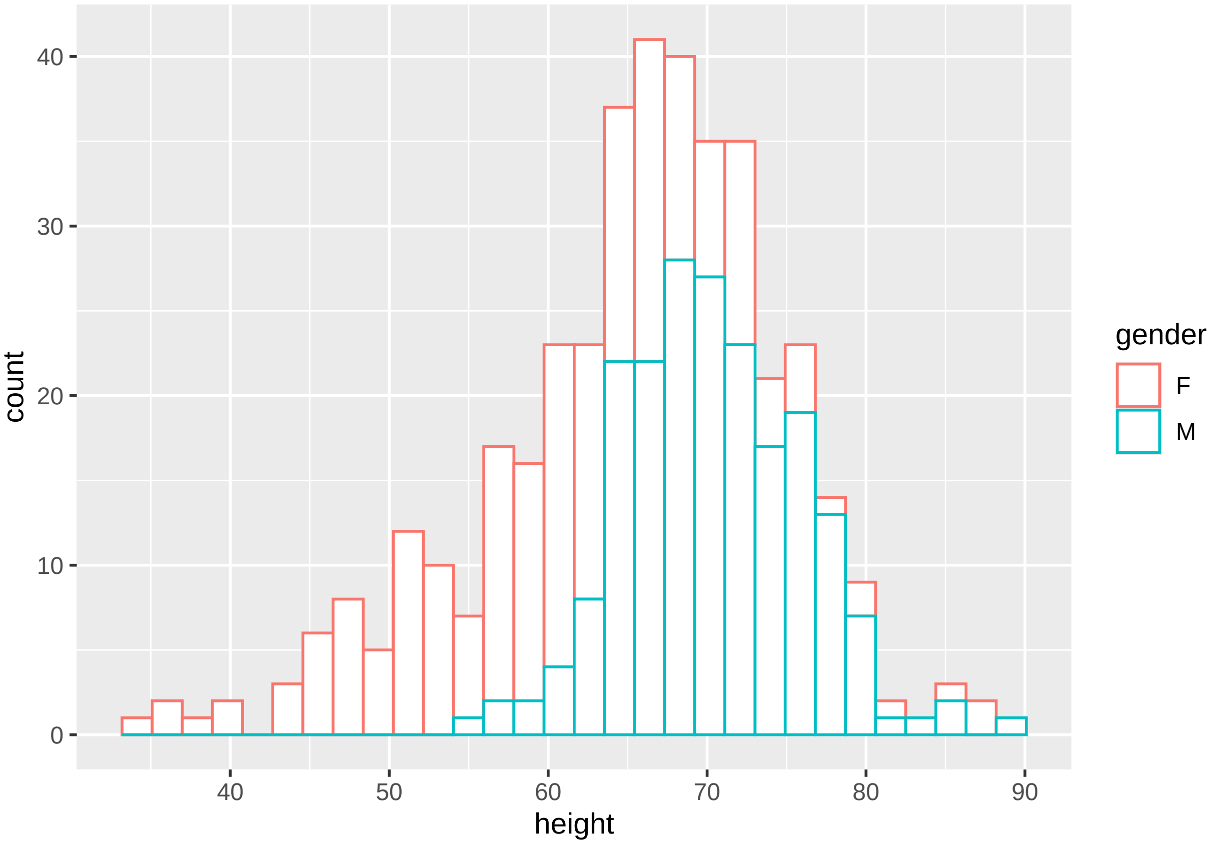
```
# # Change colors
 ggplot(df, aes(x = height)) +
   geom_histogram(color = 'blue', fill = 'red')
```

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



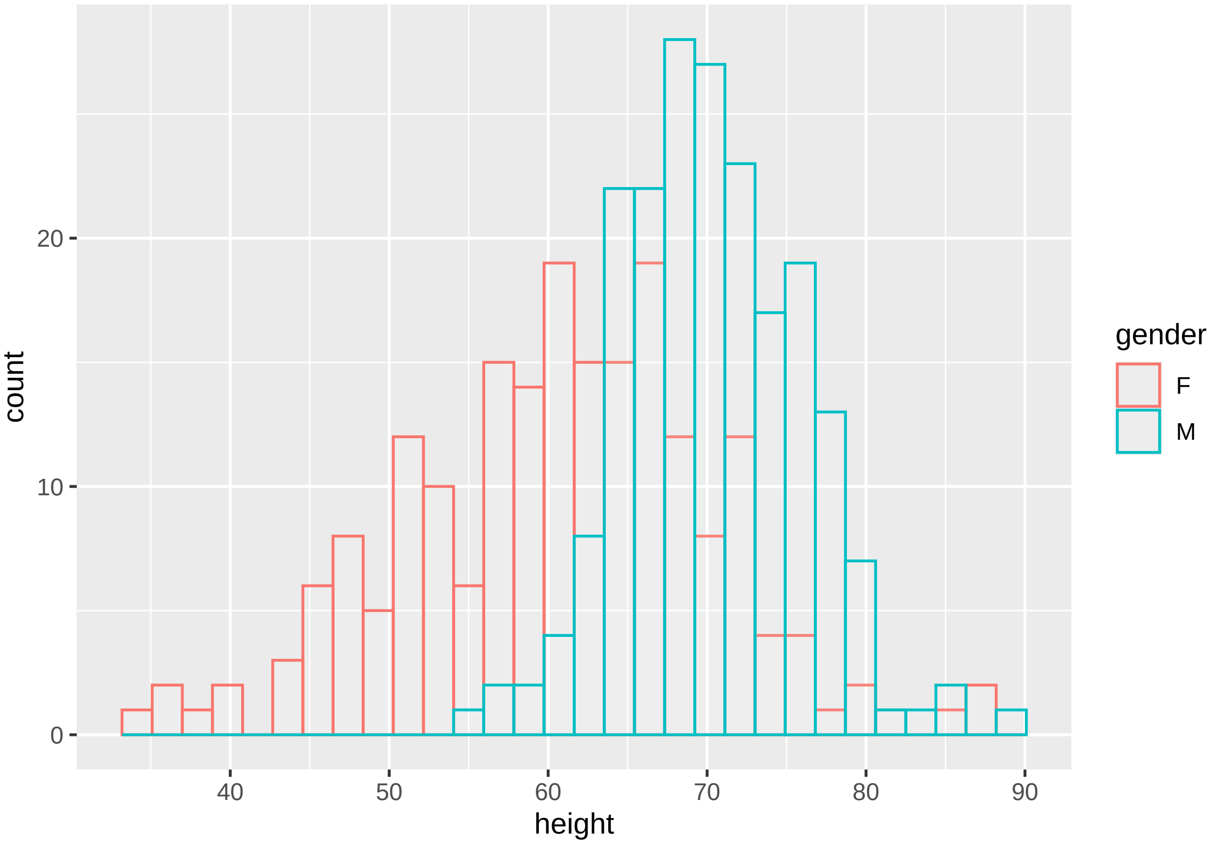
```
# Change histogram plot line colors by gender
ggplot(df, aes(x = height, color = gender)) +
  geom_histogram(fill = 'white')
```

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



```
#
# # Overlay histograms for both genders
ggplot(df, aes(x = height, color = gender)) +
  geom_histogram(fill = 'white', alpha = 0.1, position = 'identity')
```

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


count
20
10
0
40
50
60
70
80
90
height
gender
F
M