Lecture 3. tidyverse-ggplot2

Lazaro Alonso

6/2/2020

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

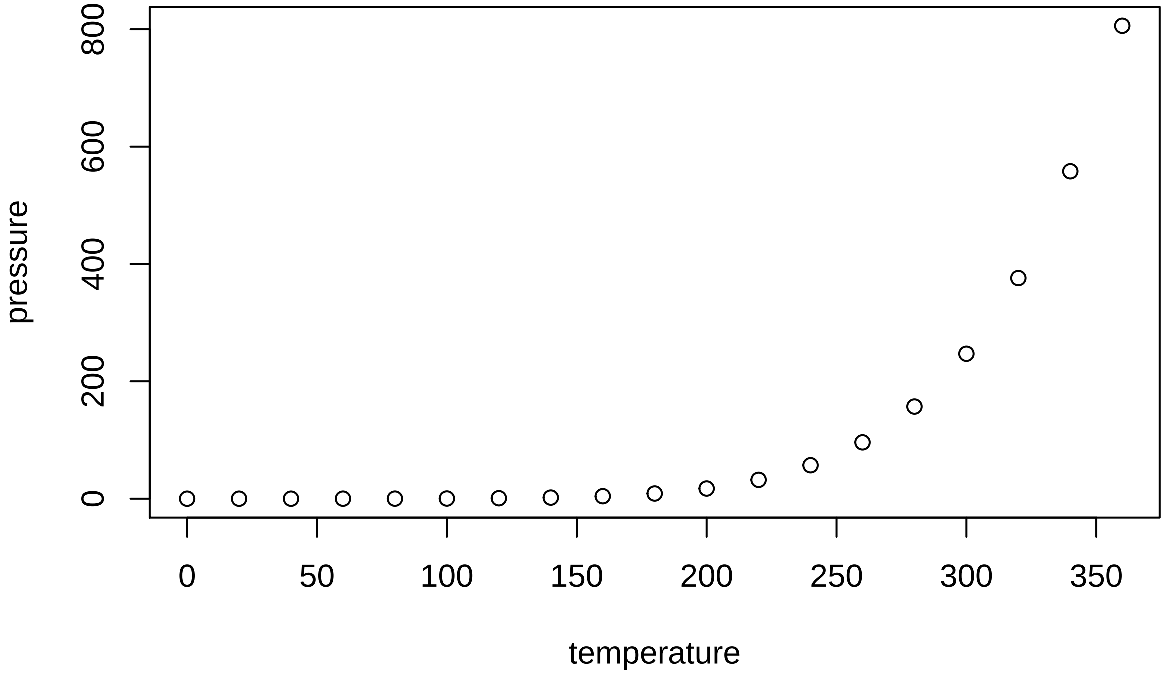
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```r
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00  
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

Data distributions

Let's see one more time the mpg data set.

```r
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.0     v purrr   0.3.4
## v tibble  3.0.1     v dplyr   0.8.5
## v tidyr   1.1.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

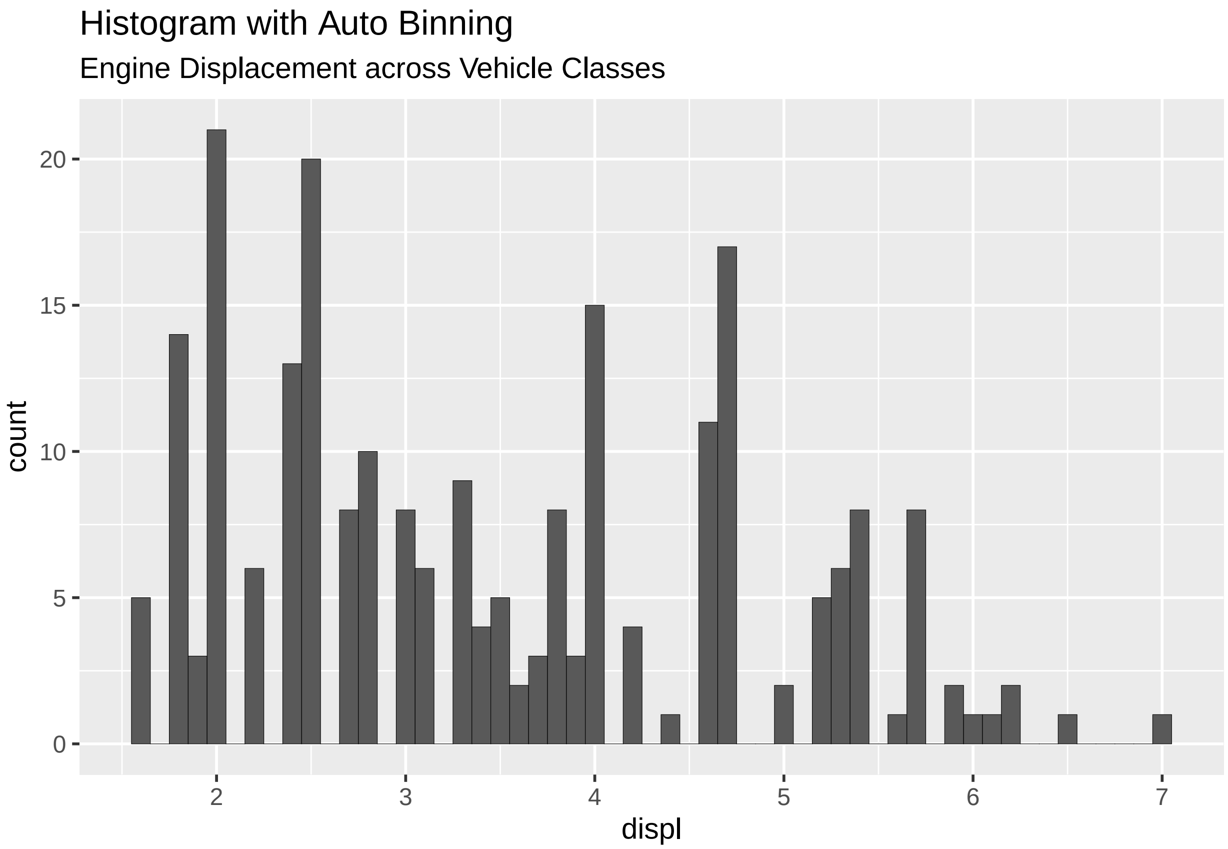
```
## -- Conflicts ------------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
glimpse(mpg)
```

```
## Rows: 234
## Columns: 11
## $ manufacturer <chr> "audi", "audi", "audi", "audi", "audi", "audi", "audi"...
## $ model        <chr> "a4", "a4", "a4", "a4", "a4", "a4", "a4", "a4 quattro"...
## $ displ        <dbl> 1.8, 1.8, 2.0, 2.0, 2.8, 2.8, 3.1, 1.8, 1.8, 2.0, 2.0,...
## $ year         <int> 1999, 1999, 2008, 2008, 1999, 1999, 2008, 1999, 1999, ...
## $ cyl          <int> 4, 4, 4, 4, 6, 6, 6, 4, 4, 4, 4, 6, 6, 6, 6, 6, 6, 8, ...
## $ trans        <chr> "auto(l5)", "manual(m5)", "manual(m6)", "auto(av)", "a...
## $ drv          <chr> "f", "f", "f", "f", "f", "f", "f", "4", "4", "4", "4",...
## $ cty          <int> 18, 21, 20, 21, 16, 18, 18, 18, 16, 20, 19, 15, 17, 17...
## $ hwy          <int> 29, 29, 31, 30, 26, 26, 27, 26, 25, 28, 27, 25, 25, 25...
## $ fl           <chr> "p", "p", "p", "p", "p", "p", "p", "p", "p", "p", "p",...
## $ class        <chr> "compact", "compact", "compact", "compact", "compact",...
```

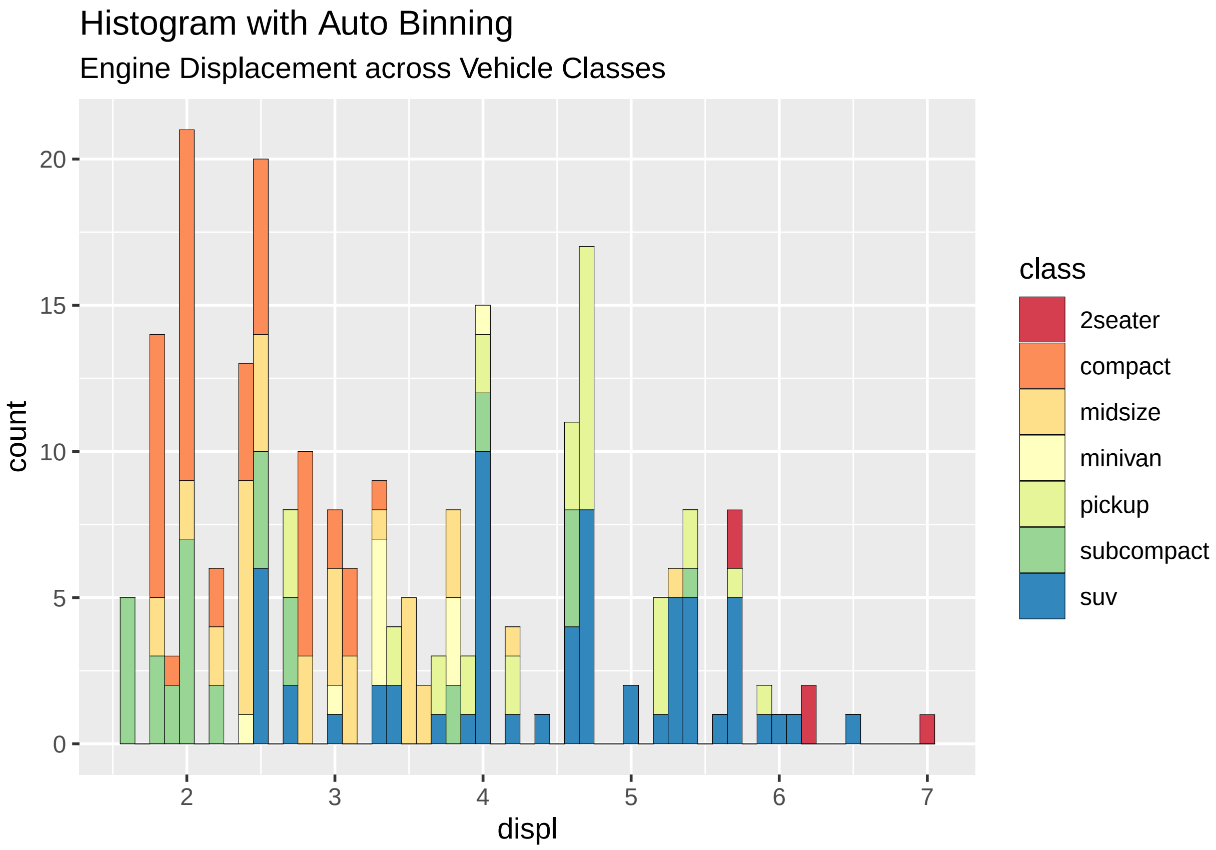
```r
help(mpg)
```

```r
g <- ggplot(mpg, aes(displ)) + scale_fill_brewer(palette = "Spectral")

g + geom_histogram(binwidth = .1,
                   col="black",
                   size=.1) +  # change binwidth
  labs(title="Histogram with Auto Binning",
       subtitle="Engine Displacement across Vehicle Classes")
```

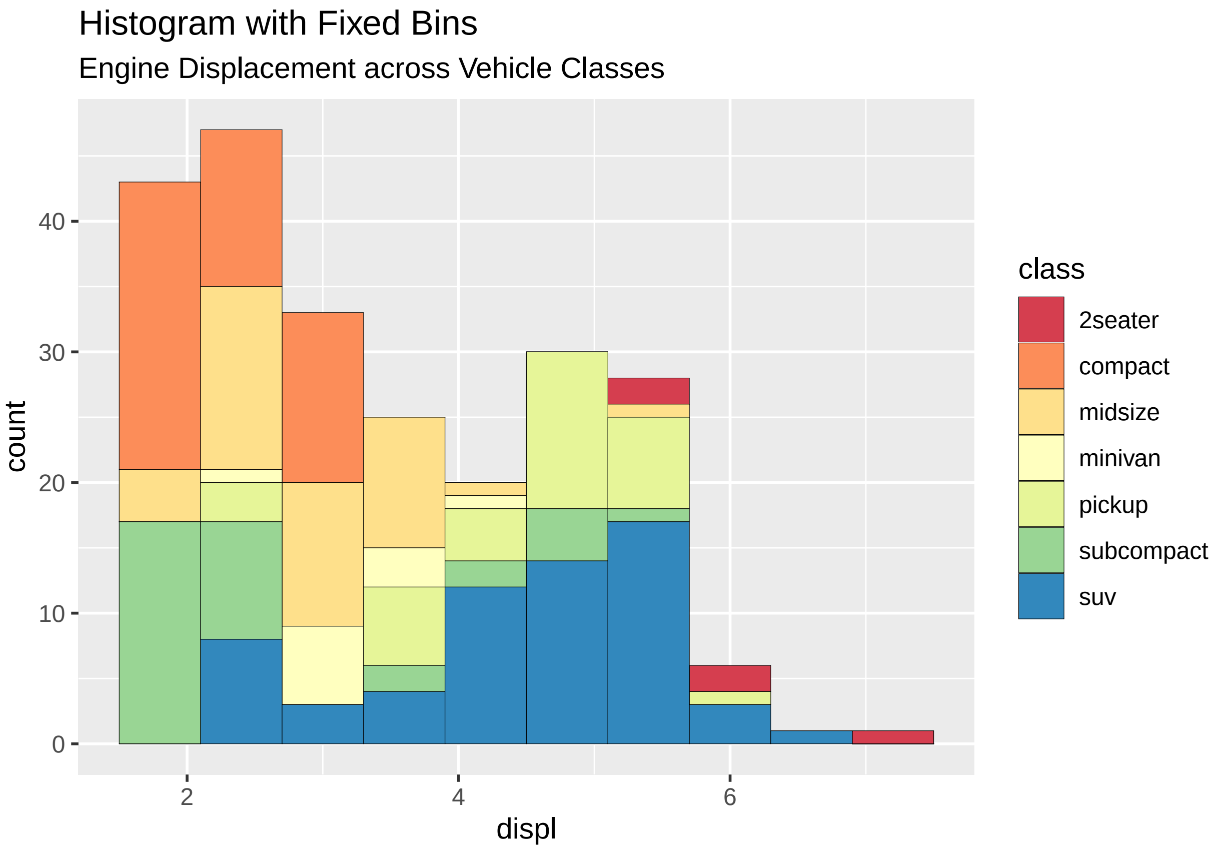


Let's do it per class.

```
library(tidyverse)
g <- ggplot(mpg, aes(displ)) + scale_fill_brewer(palette = "Spectral")

g + geom_histogram(aes(fill=class),
                   binwidth = .1,
                   col="black",
                   size=.1) +  # change binwidth
  labs(title="Histogram with Auto Binning",
       subtitle="Engine Displacement across Vehicle Classes")
```

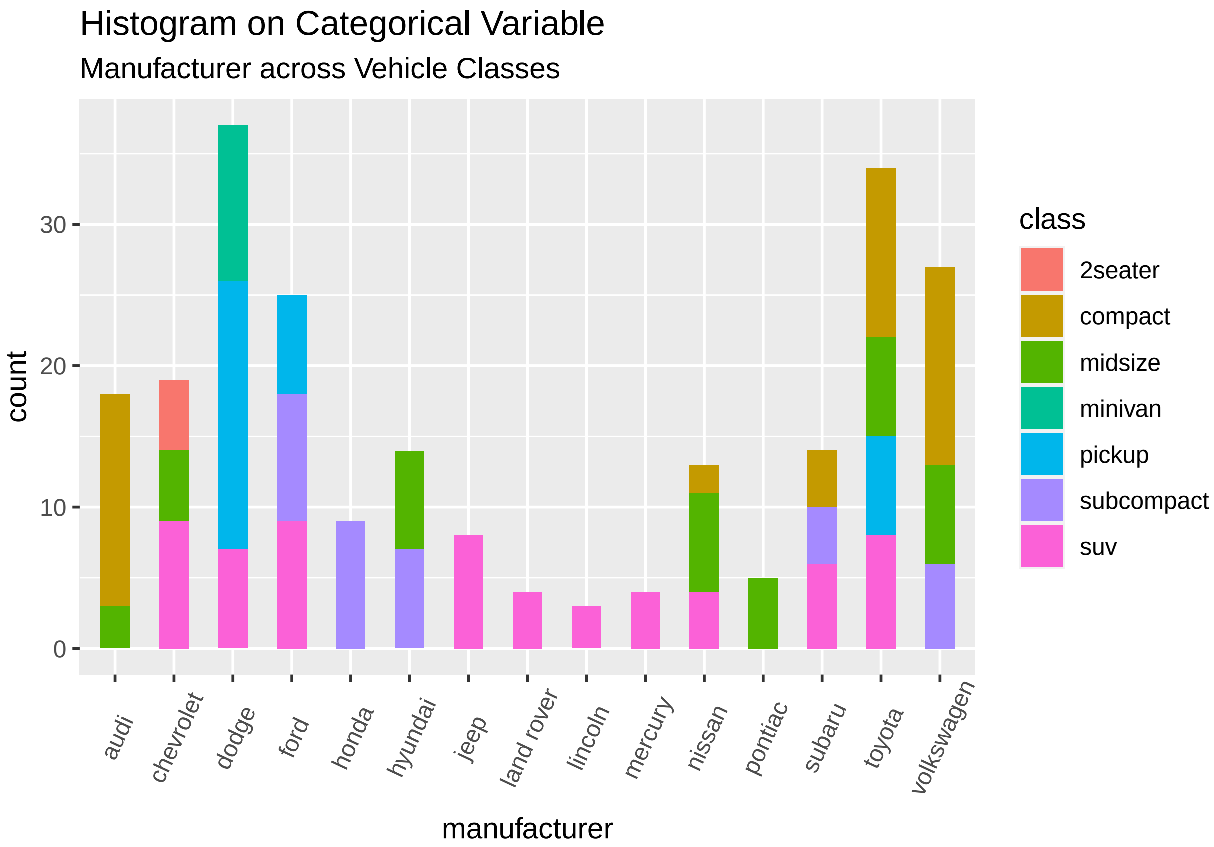


```
g + geom_histogram(aes(fill=class),
                   bins=10,
                   col="black",
                   size=.1) +   # change number of bins
  labs(title="Histogram with Fixed Bins",
       subtitle="Engine Displacement across Vehicle Classes")
```



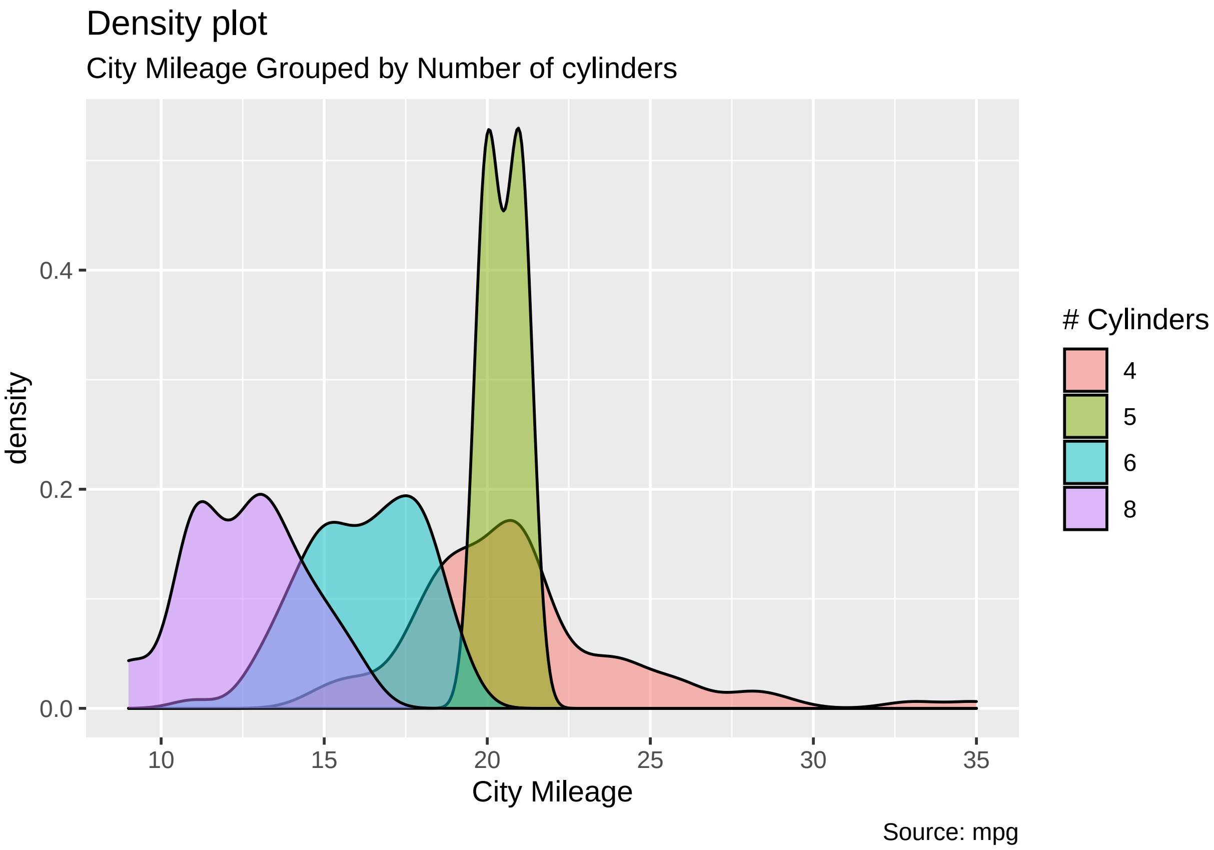
### Histogram on a categorical variable

```
g <- ggplot(mpg, aes(manufacturer))
g + geom_bar(aes(fill=class), width = 0.5) +
  theme(axis.text.x = element_text(angle=65, vjust=0.6)) +
  labs(title="Histogram on Categorical Variable",
       subtitle="Manufacturer across Vehicle Classes")
```



### Density plot

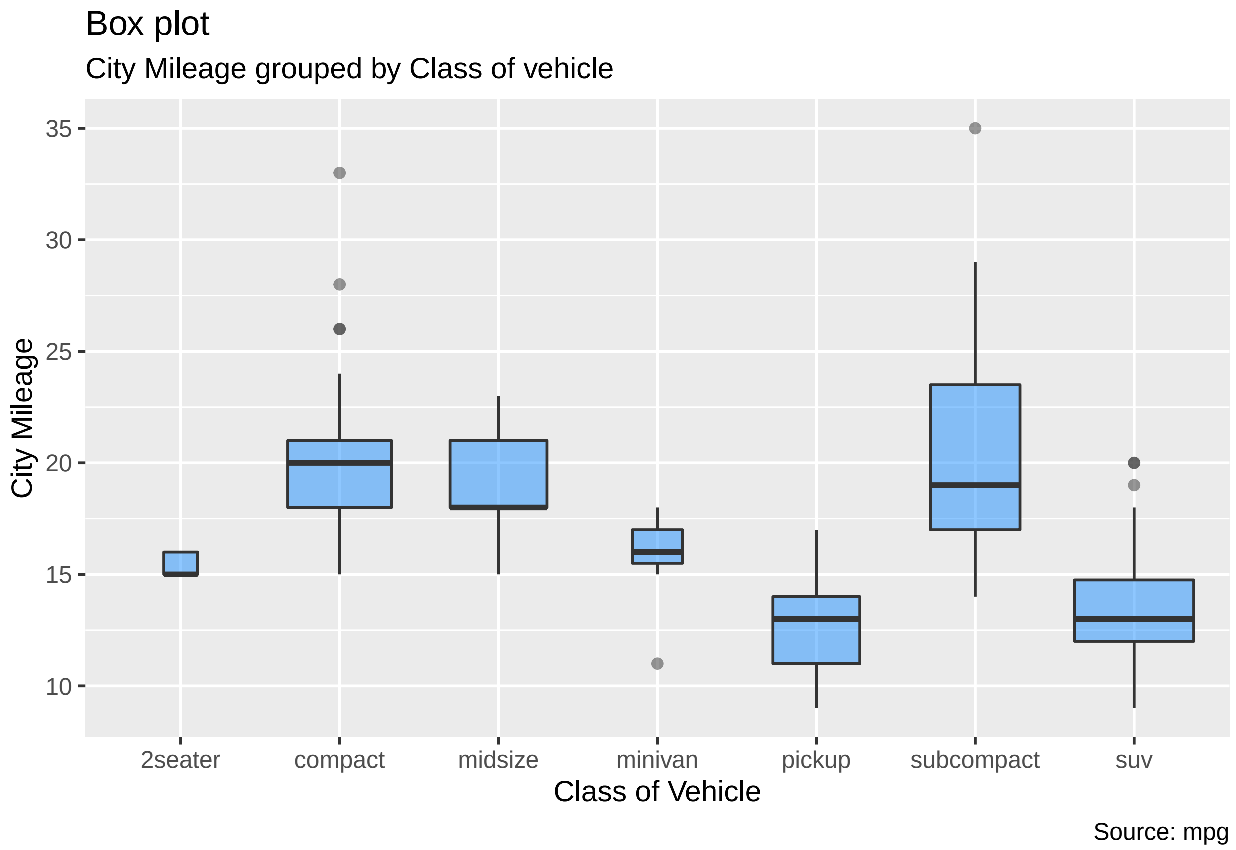
```
g <- ggplot(mpg, aes(cty))
g + geom_density(aes(fill=factor(cyl)), alpha=0.5) +
    labs(title="Density plot",
         subtitle="City Mileage Grouped by Number of cylinders",
         caption="Source: mpg",
         x="City Mileage",
         fill="# Cylinders")
```



## Box plots

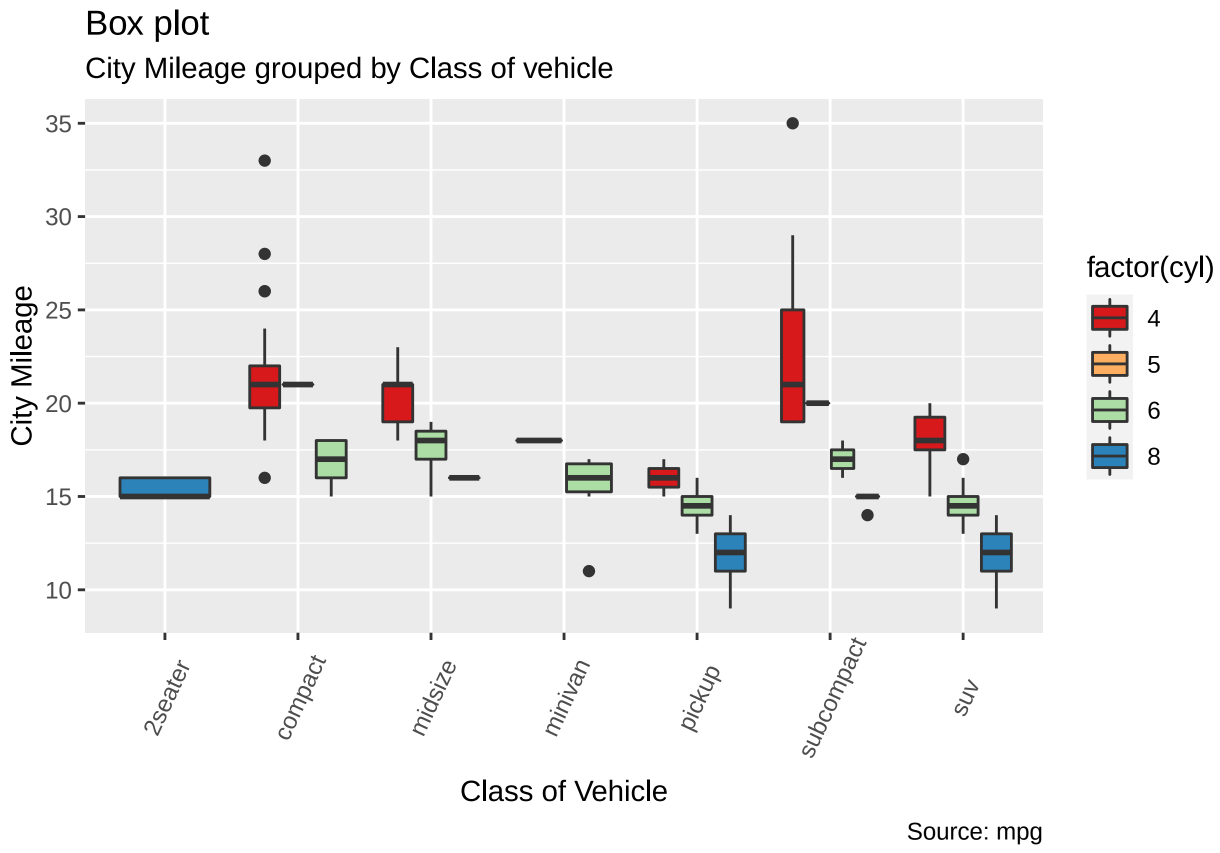
The dark line inside the box represents the median. The top of box is 75%ile and bottom of box is 25%ile. The end points of the lines (aka whiskers) is at a distance of 1.5*IQR, where IQR or Inter Quartile Range is the distance between 25th and 75th percentiles. The points outside the whiskers are marked as dots and are normally considered as extreme points.

```r
g <- ggplot(mpg, aes(class, cty))
g + geom_boxplot(varwidth=T, fill="dodgerblue", alpha = 0.5) +
    labs(title="Box plot",
         subtitle="City Mileage grouped by Class of vehicle",
         caption="Source: mpg",
         x="Class of Vehicle",
         y="City Mileage")
```



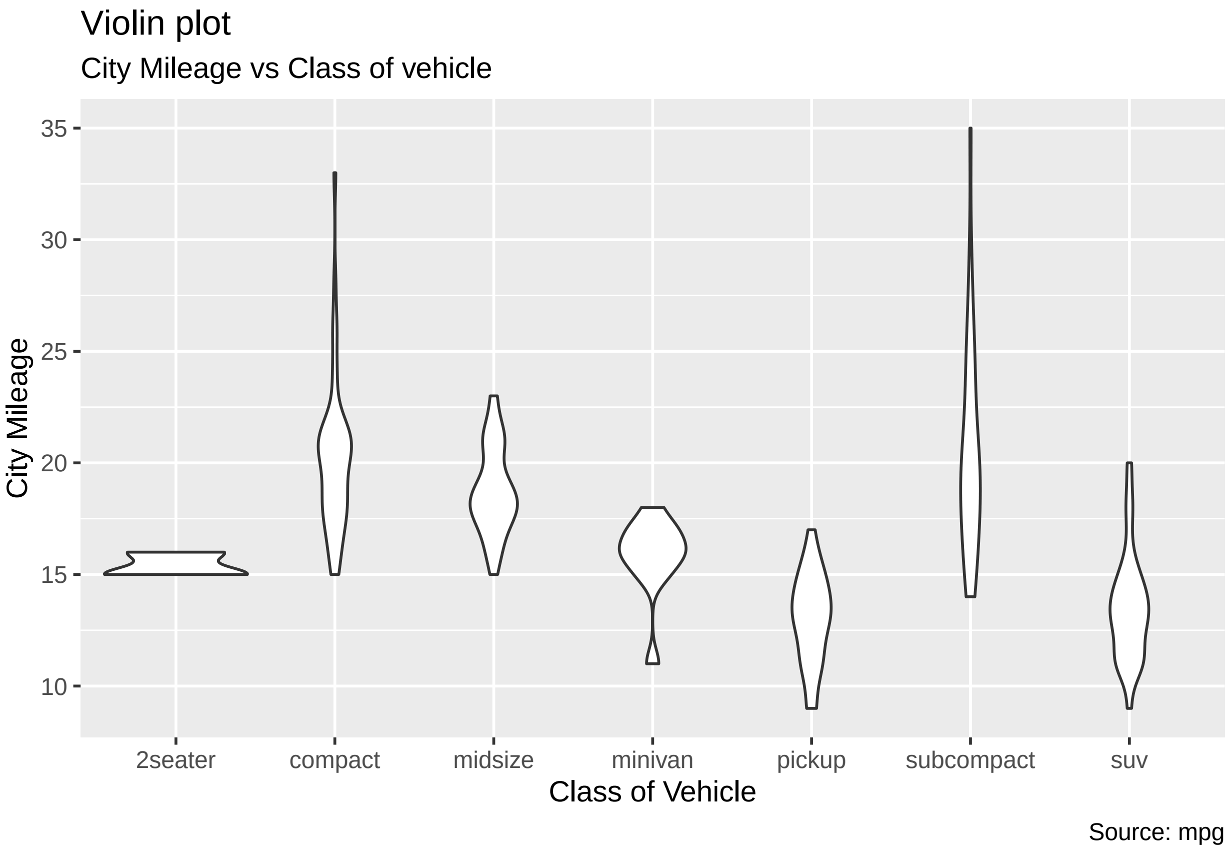
Setting varwidth=T adjusts the width of the boxes to be proportional to the number of observation it contains.

```
g <- ggplot(mpg, aes(class, cty))
g + geom_boxplot(aes(fill=factor(cyl))) +  scale_fill_brewer(palette = "Spectral") +
  theme(axis.text.x = element_text(angle=65, vjust=0.6)) +
  labs(title="Box plot",
       subtitle="City Mileage grouped by Class of vehicle",
       caption="Source: mpg",
       x="Class of Vehicle",
       y="City Mileage")
```

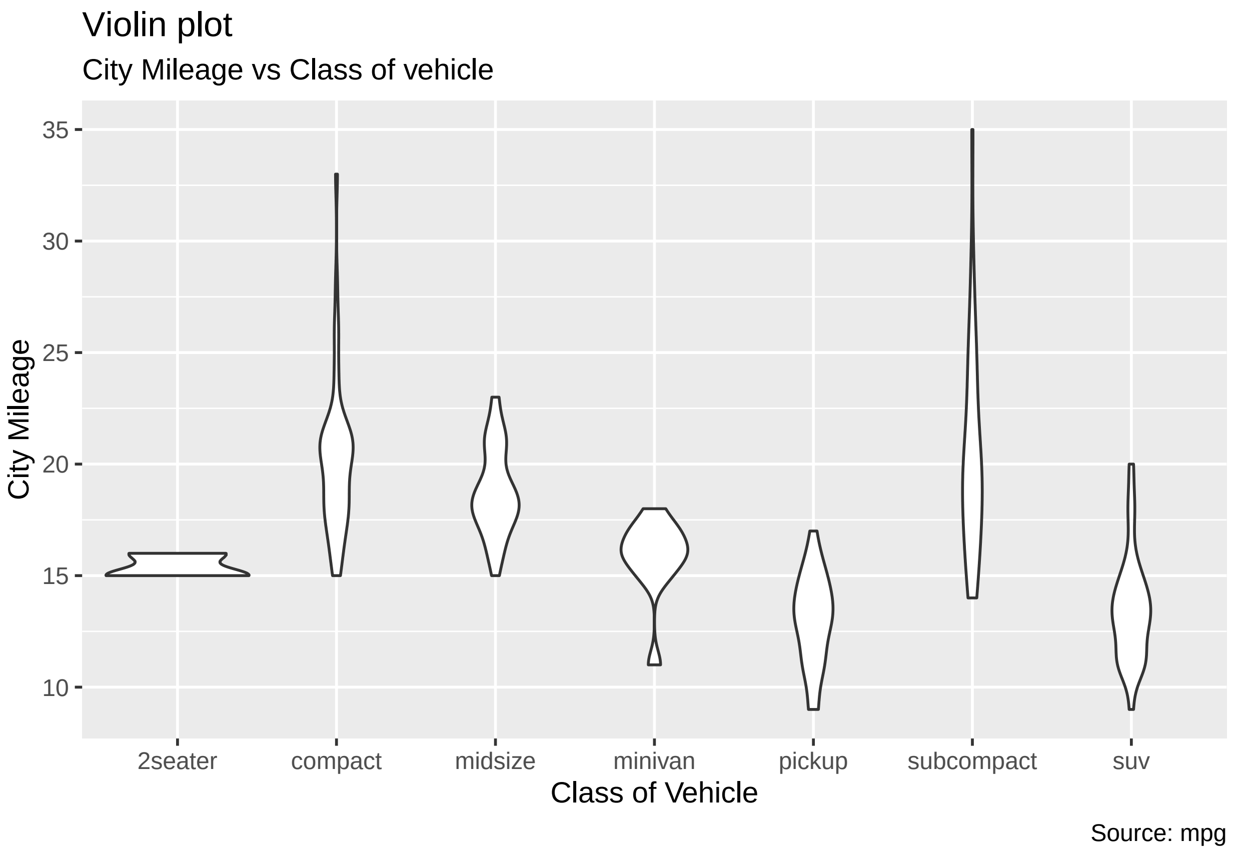


### Violin plot

```
g <- ggplot(mpg, aes(class, cty))
g + geom_violin() +
  labs(title="Violin plot",
       subtitle="City Mileage vs Class of vehicle",
       caption="Source: mpg",
       x="Class of Vehicle",
       y="City Mileage")
```

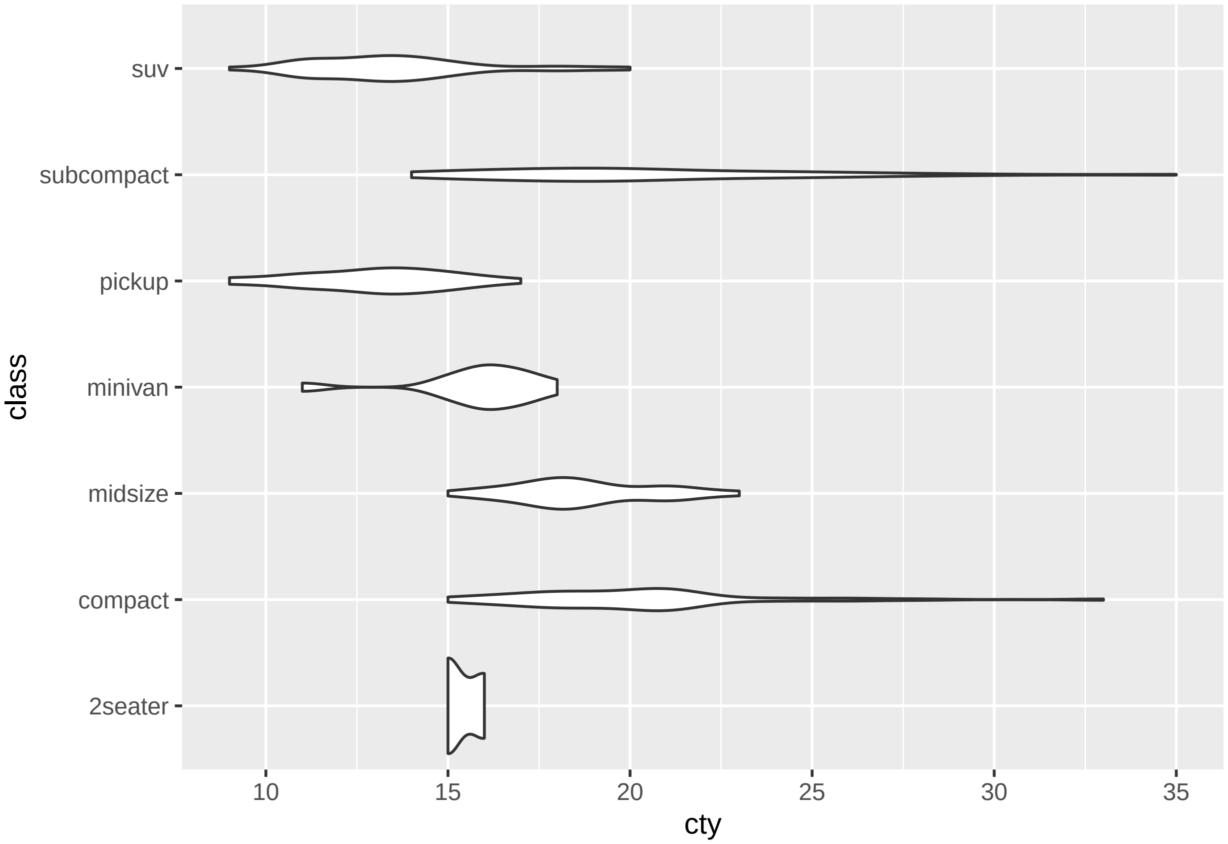


```r
g <- ggplot(mpg, aes(class, cty))
p <- g + geom_violin() +
  labs(title="Violin plot",
       subtitle="City Mileage vs Class of vehicle",
       caption="Source: mpg",
       x="Class of Vehicle",
       y="City Mileage")
p
```



```
g + geom_violin(TRIM=FALSE) + coord_flip()
```

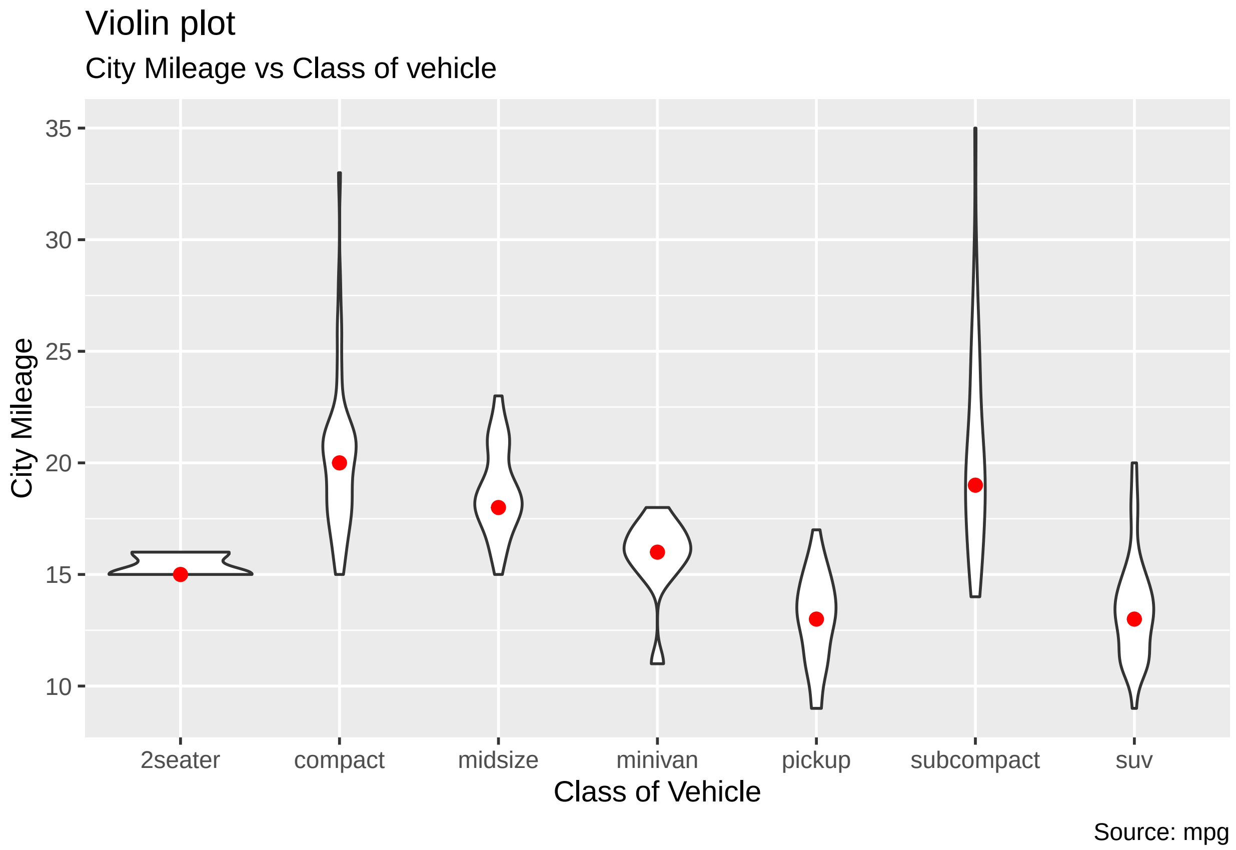
```
## Warning: Ignoring unknown parameters: TRIM
```



### Add summary statistics on a violin plot

```
p + stat_summary(fun.y=median, geom="point", size=2, color="red")
```

```
## Warning: `fun.y` is deprecated. Use `fun` instead.
```

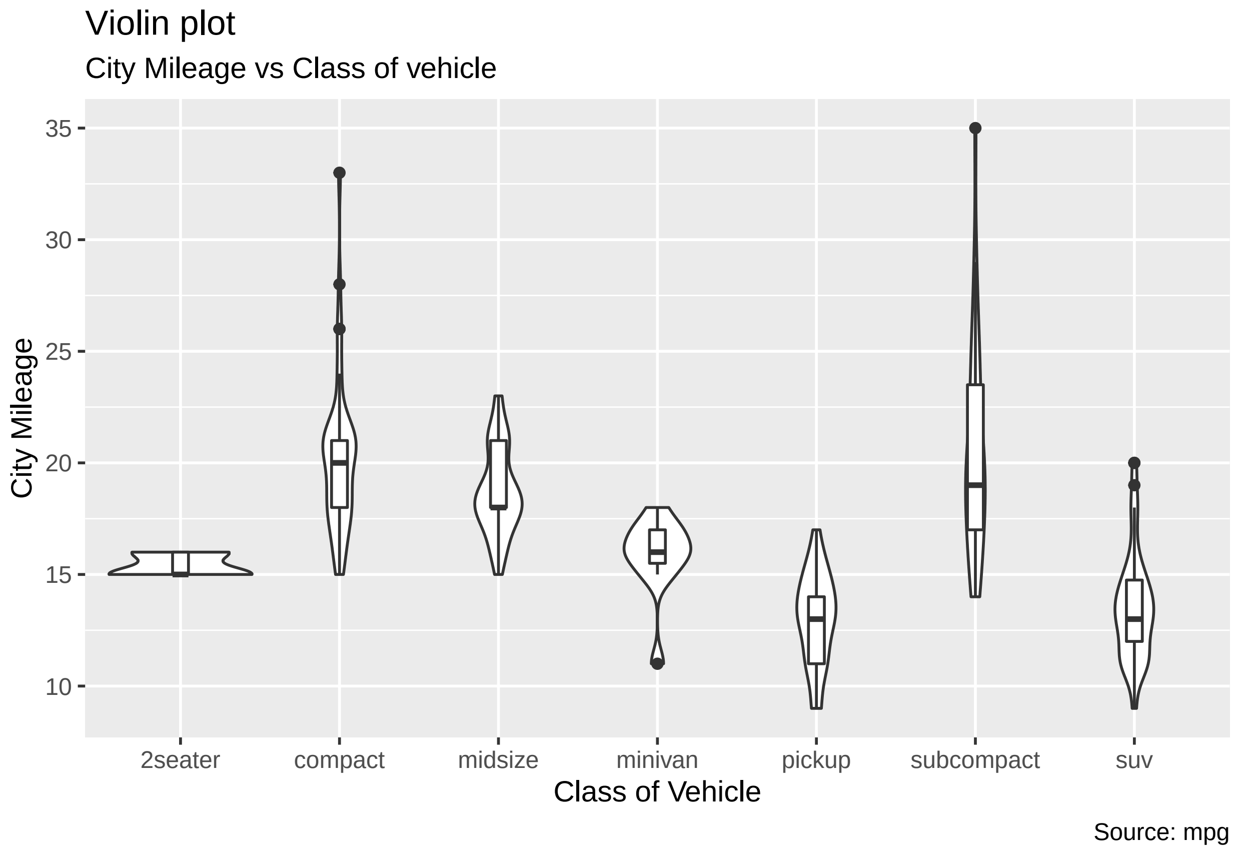

Violin plot

City Mileage vs Class of vehicle

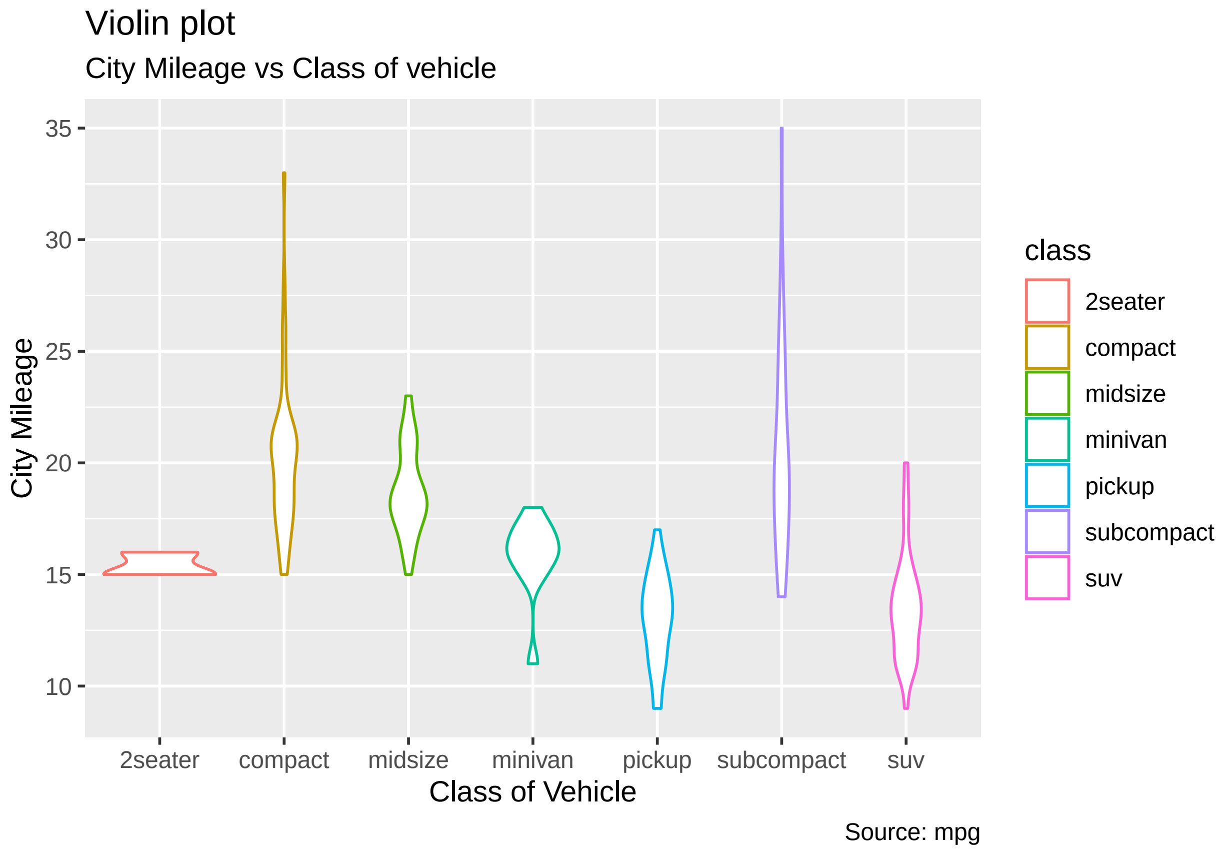
Source: mpg

## Add median and quartile

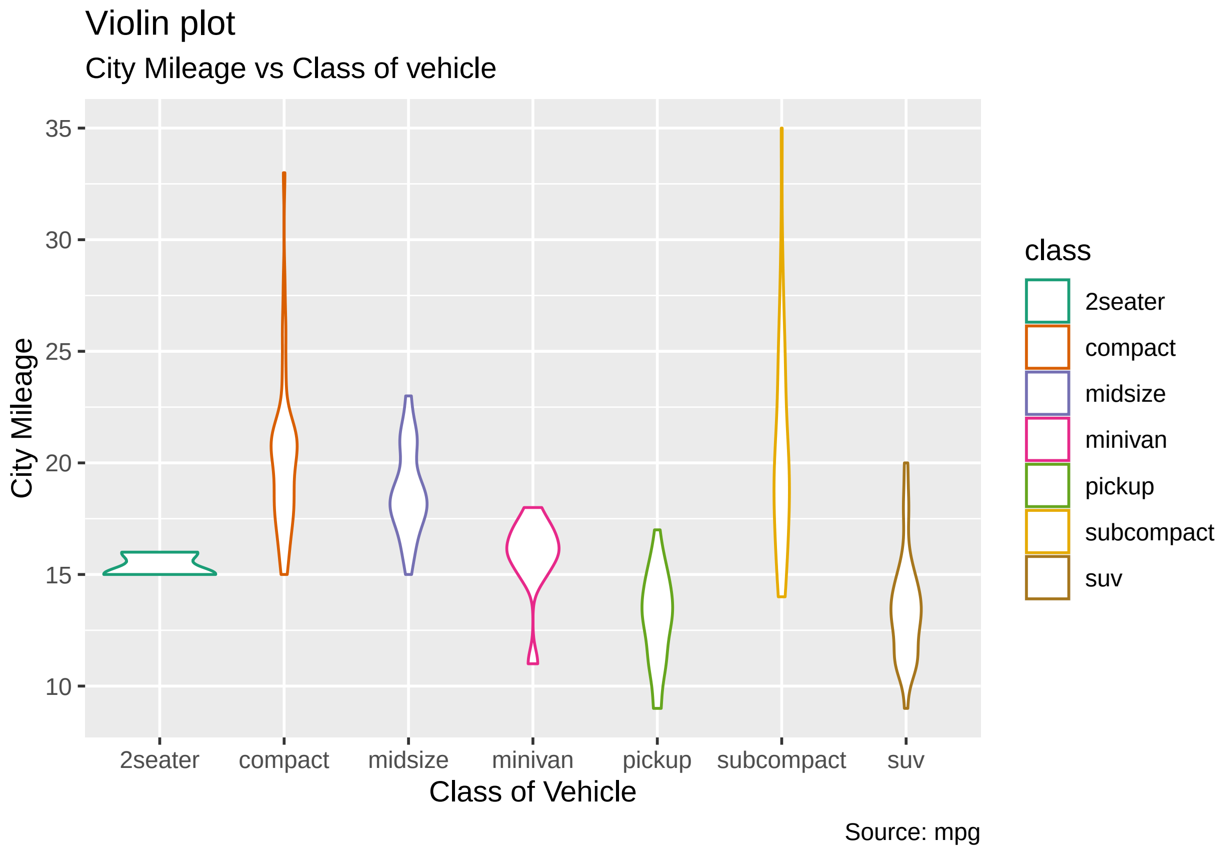
```
p + geom_boxplot(width=0.1)
```



```
g <- ggplot(mpg, aes(class, cty, color = class))
pc <- g + geom_violin() +
  labs(title="Violin plot",
       subtitle="City Mileage vs Class of vehicle",
       caption="Source: mpg",
       x="Class of Vehicle",
       y="City Mileage")
pc
```



```
pc + scale_color_brewer(palette="Dark2")
```

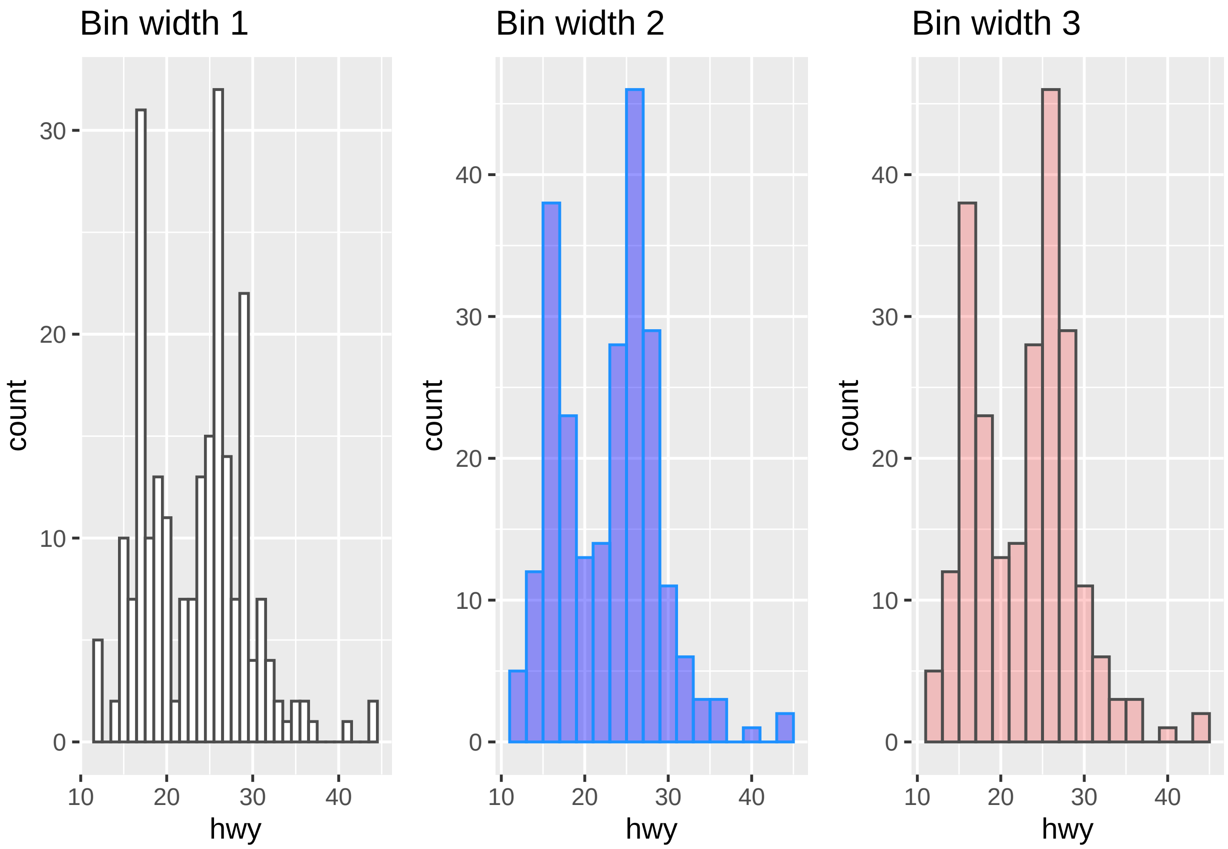


### More than one plot

```
# install.packages("gridExtra")
library("gridExtra")
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

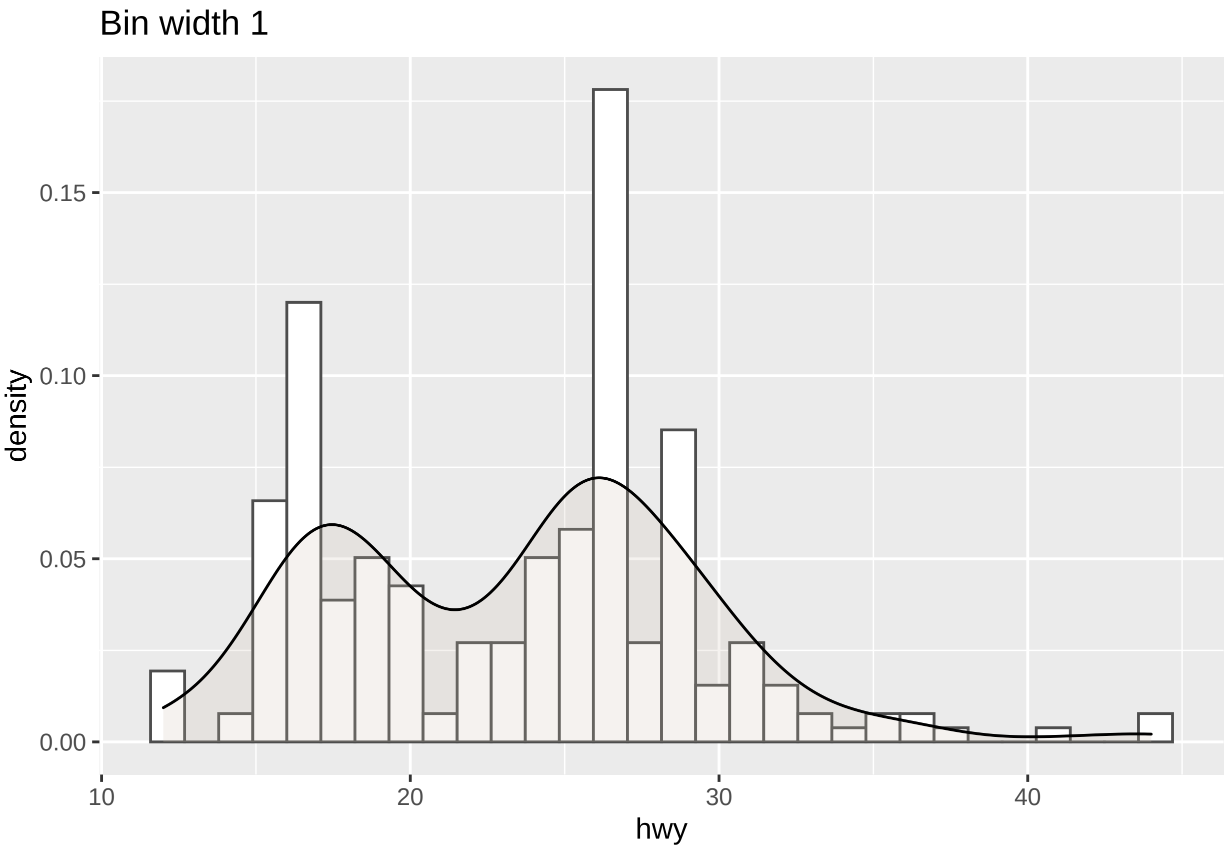
```
p_data <- ggplot(mpg, aes(x = hwy))
p1 <- p_data + geom_histogram(binwidth = 1, color = "grey30", fill = "white") +
  labs(title="Bin width 1")
p2 <-  p_data + geom_histogram(binwidth = 2, color = "dodgerblue",
                               fill = "blue", alpha = 0.4) +
  labs(title="Bin width 2")

p3 <- p_data + geom_histogram(binwidth = 2, color = "grey30",
                              fill = "red", alpha = 0.2) +
  labs(title="Bin width 3")

grid.arrange(p1, p2, p3, ncol = 3)
```



### overlap density and histogram

```
ggplot(mpg, aes(x = hwy)) + geom_histogram(aes(y=..density..), color = "grey30", fill = "white") +geom_
```

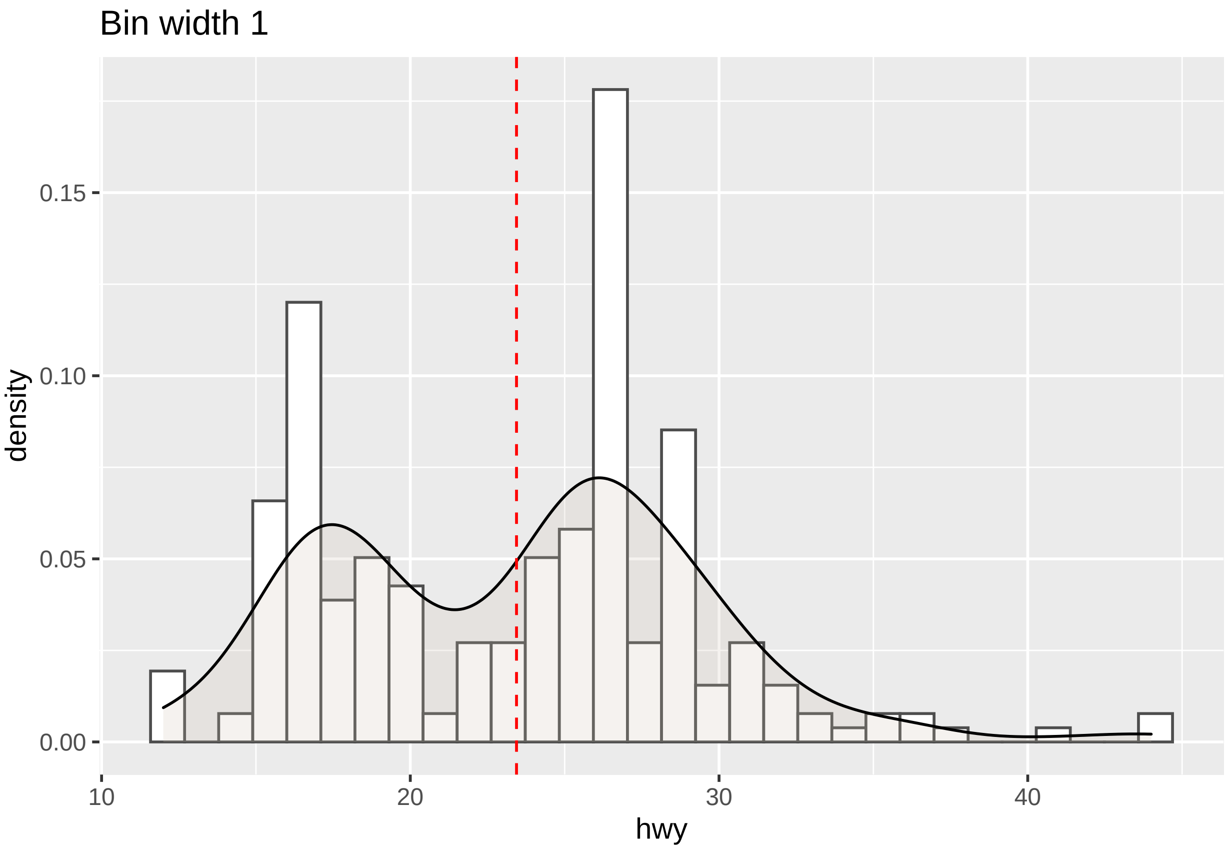
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



### Add Value Markers

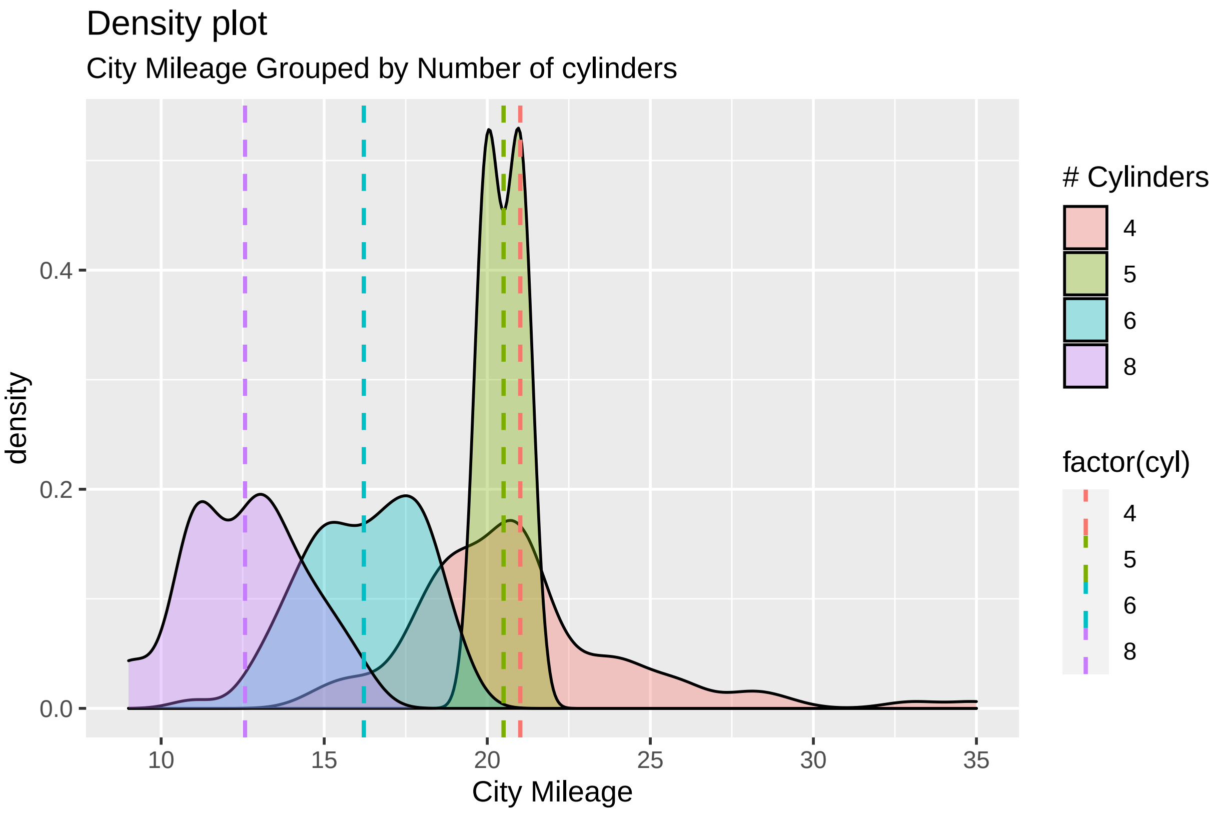
```
ggplot(mpg, aes(x = hwy)) + geom_histogram(aes(y=..density..), color = "grey30", fill = "white") +geom_
```

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



To add lines for grouped data we need to do a little computation prior to plotting.

```
compare_mean <- mpg %>%
        group_by(cyl) %>%
        summarise(Mean = mean(cty))

g <- ggplot(mpg, aes(cty))
g + geom_density(aes(fill=factor(cyl)), alpha=0.35) +
    labs(title="Density plot",
         subtitle="City Mileage Grouped by Number of cylinders",
         caption="Source: mpg",
         x="City Mileage",
         fill="# Cylinders") +
  geom_vline(data = compare_mean, aes(xintercept = Mean, color = factor(cyl)),
                   linetype = "dashed", size = 0.75)
```



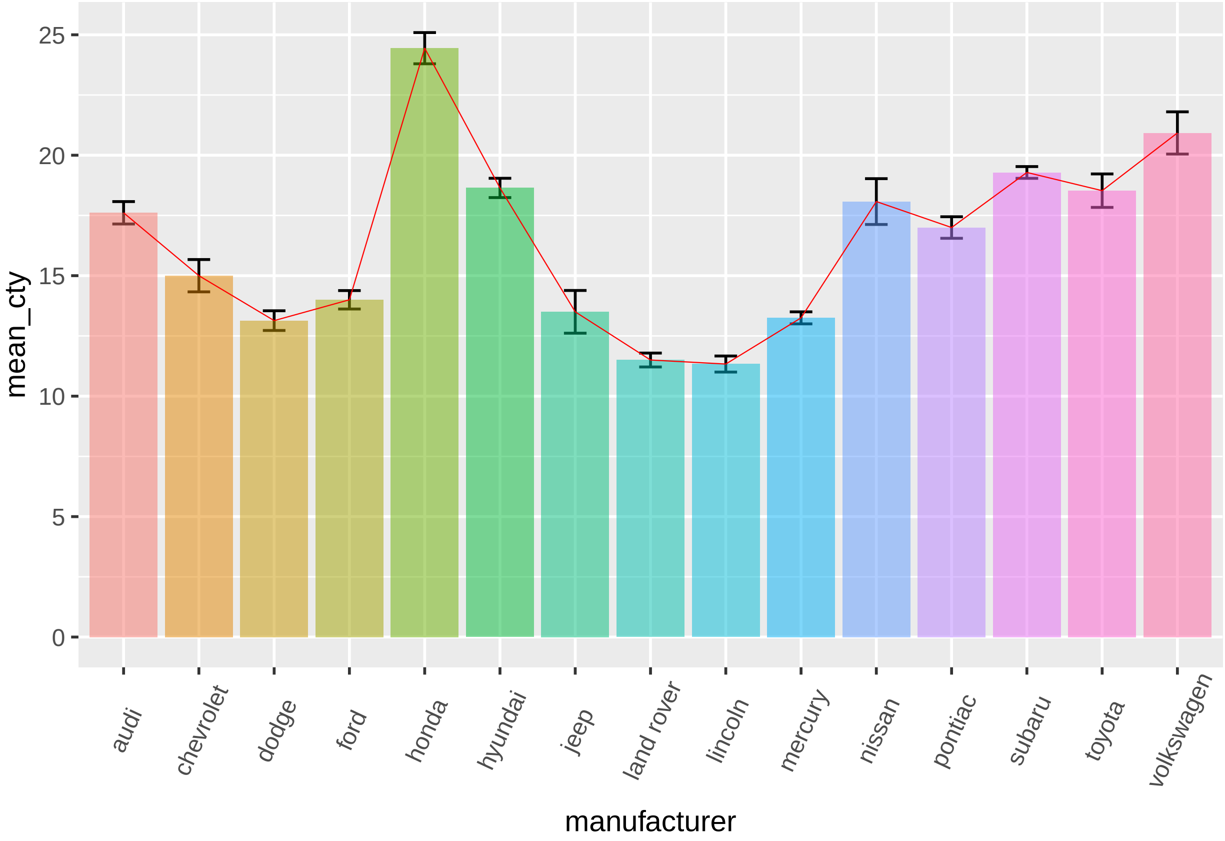
Source: mpg

### Error bars

```
# Create new dataframe mpg_means_se
mpg_means_se <- mpg %>%
  group_by(manufacturer) %>% # Group the data by manufacturer
  summarize(mean_cty=mean(cty), # Create variable with mean of cty per group
            sd_cty=sd(cty), # Create variable with sd of cty per group
            N_cty=n(), # Create new variable N of cty per group
            se=sd_cty/sqrt(N_cty), # Create variable with se of cty per group
            upper_limit=mean_cty+se, # Upper limit
            lower_limit=mean_cty-se # Lower limit
            )
mpg_means_se
```

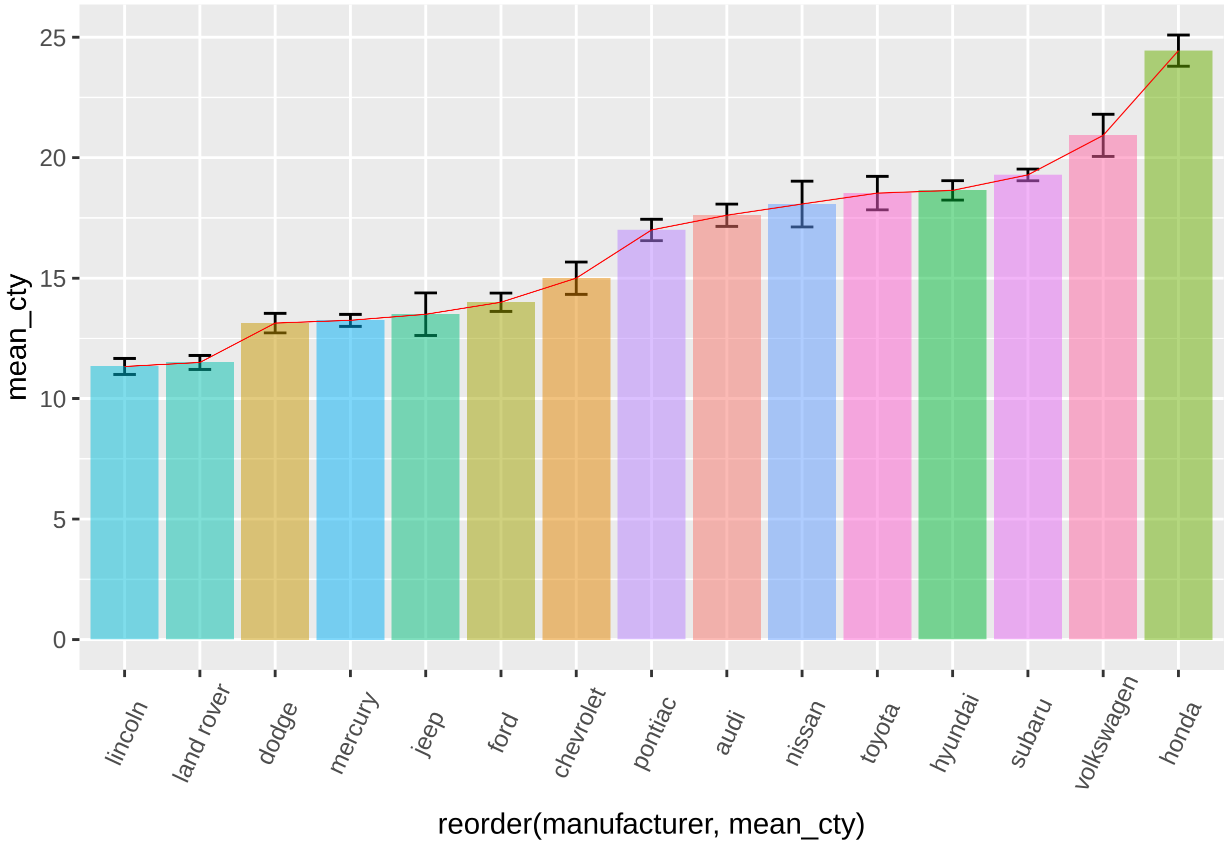
```
## # A tibble: 15 x 7
##    manufacturer mean_cty sd_cty N_cty    se upper_limit lower_limit
##    <chr>           <dbl>  <dbl> <int> <dbl>       <dbl>       <dbl>
##  1 audi             17.6  1.97     18 0.465        18.1        17.1
##  2 chevrolet        15    2.92     19 0.671        15.7        14.3
##  3 dodge            13.1  2.49     37 0.409        13.5        12.7
##  4 ford             14    1.91     25 0.383        14.4        13.6
##  5 honda            24.4  1.94      9 0.648        25.1        23.8
##  6 hyundai          18.6  1.50     14 0.401        19.0        18.2
##  7 jeep             13.5  2.51      8 0.886        14.4        12.6
##  8 land rover       11.5  0.577     4 0.289        11.8        11.2
##  9 lincoln          11.3  0.577     3 0.333        11.7        11
## 10 mercury          13.2  0.5       4 0.25         13.5        13
## 11 nissan           18.1  3.43     13 0.950        19.0        17.1
## 12 pontiac           17     1          5 0.447         17.4        16.6
## 13 subaru            19.3   0.914     14 0.244         19.5        19.0
## 14 toyota            18.5   4.05      34 0.694         19.2        17.8
## 15 volkswagen        20.9   4.56      27 0.877         21.8        20.0
```

```
ggplot(mpg_means_se, aes(x=manufacturer, y=mean_cty, fill=manufacturer)) +
  geom_errorbar(aes(ymin=lower_limit, ymax=upper_limit), width = 0.3) +
   geom_bar(stat="identity", alpha= 0.5) +
  theme(axis.text.x = element_text(angle=65, vjust=0.6), legend.position = "none") +
  geom_line(aes(group = 1), size = 0.2, color = "red")
```



Reordering…

```
ggplot(mpg_means_se, aes(x=reorder(manufacturer, mean_cty), y=mean_cty, fill=manufacturer)) +
  geom_errorbar(aes(ymin=lower_limit, ymax=upper_limit), width = 0.3) +
   geom_bar(stat="identity", alpha= 0.5) +
  theme(axis.text.x = element_text(angle=65, vjust=0.6), legend.position = "none") +
  geom_line(aes(group = 1), size = 0.2, color = "red")
```



Now, use all these examples to explore the nycflights data set.

EXAMPLE:

```
library(nycflights13)
head(flights)
```

```
## # A tibble: 6 x 19
##    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
##   <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
## 1  2013     1     1      517            515         2      830            819
## 2  2013     1     1      533            529         4      850            830
## 3  2013     1     1      542            540         2      923            850
## 4  2013     1     1      544            545        -1     1004           1022
## 5  2013     1     1      554            600        -6      812            837
## 6  2013     1     1      554            558        -4      740            728
## # ... with 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>,
## #   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>,
## #   hour <dbl>, minute <dbl>, time_hour <dttm>
```

Data sets

```
ls("package:nycflights13")
```

```
## [1] "airlines" "airports" "flights"  "planes"   "weather"
```

```
airlines_data <- airlines
airports_data <- airports
flights_data <- flights
```

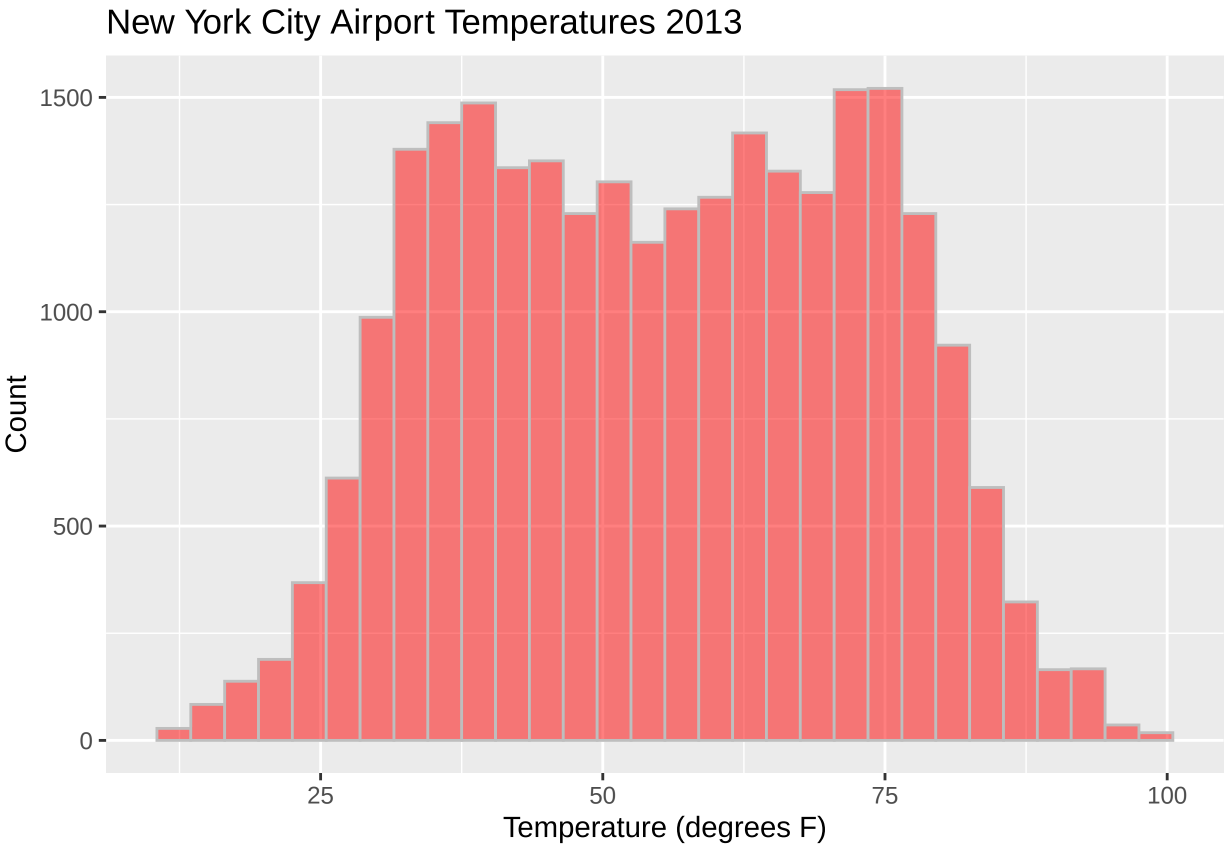
```
planes_data <- planes
weather_data <- weather
# Variables in flights dataset
?flights
```

```
head(weather)
```

```
## # A tibble: 6 x 15
##   origin  year month   day  hour  temp  dewp humid wind_dir wind_speed wind_gust
##   <chr>  <int> <int> <int> <int> <dbl> <dbl> <dbl>    <dbl>      <dbl>     <dbl>
## 1 EWR     2013     1     1     1  39.0  26.1  59.4      270      10.4         NA
## 2 EWR     2013     1     1     2  39.0  27.0  61.6      250       8.06        NA
## 3 EWR     2013     1     1     3  39.0  28.0  64.4      240      11.5         NA
## 4 EWR     2013     1     1     4  39.9  28.0  62.2      250      12.7         NA
## 5 EWR     2013     1     1     5  39.0  28.0  64.4      260      12.7         NA
## 6 EWR     2013     1     1     6  37.9  28.0  67.2      240      11.5         NA
## # ... with 4 more variables: precip <dbl>, pressure <dbl>, visib <dbl>,
## #   time_hour <dttm>
```

```
ggplot(weather, aes(x = temp)) +
  geom_histogram(binwidth = 3, color = "grey", fill = "red", alpha= 0.5) +
  labs(x = "Temperature (degrees F)", y = "Count",
       title = "New York City Airport Temperatures 2013")
```

```
## Warning: Removed 1 rows containing non-finite values (stat_bin).
```


New York City Airport Temperatures 2013
Count
Temperature (degrees F)