ggplot2 그래프

ggplot2 패키지

데이터기반 그래픽 생성

데이터기반으로 기하학적 객체들(점,선,막대 등)에 미적특성(색상,모양,크기)를 매핑하고 적용한 후 바로 화면창에서 결과를 확인할 수 있으며, 추세선 등 통계적 결과를 추가할 수도 있다.

ggplot2 문법을 구성하는 기본 요소.

- 데이터 프레임(data frame)
- 색상, 크기 같은 외적 요소(aes)
- 점, 선, 모양 같은 기하학적 요소(geoms)
- 통계적 처리 방법(stats)
- aes에서 사용할 스케일(scale)

ggplot2을 이용한 그래프 생성 단계

- 1단계 : 배경 설정 (축)
- 2단계 : 그래프 추가(점, 막대, 선)
- 3단계 : 설정 추가 (축범위, 색, 표식)

ggplot 그래프를 작성하기 위해 먼저 다음과 같이 **ggplot2**와 **dyplr** 패키지를 불러온다.

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

데이터 준비

그래프 작성을 연습하기 ggplot2 패키지에서 제공하는 mpg 데이터와 economic 데이터를 사용한다.
mpg 데이터는 미환경보호국에서 제공하는 자동차 연비 데이터로 변수 구성은 다음과 같다.

변수	설명
manufacturer	제조회사
model	모델명
displ	배기량(displcement)
year	생산연도
cyl	실린더 개수(cylinders)
trans	변속기 종류(transmission)
drv	구동 방식(drive wheel)
cty	도시 연비(city)
hwy	고속도로 연비(highway)
fl	연료 종류(fuel)
class	자동차 종류

economic 데이터는 미국의 경제 지표들을 월별로 나타낸 데이터로 변수 구성은 다음과 같다.

변수	설명
date	날짜(월별로 정보를 수집한 시점)
pce	개인소비지출(personal concumption expenditure)
pop	총인구 (1000 단위)
psavert	개인저축률(personal saving rate)
uempmed	평균 실업 기간 (주 단위)
unemploy	실업자수 (1000 단위)

산점도(Scatter Plot)

산점도는 데이터를 x축과 y축에 점으로 표현한 그래프로, 연속 값으로 된 두 변수의 관계를 표현할 때 사용한다.

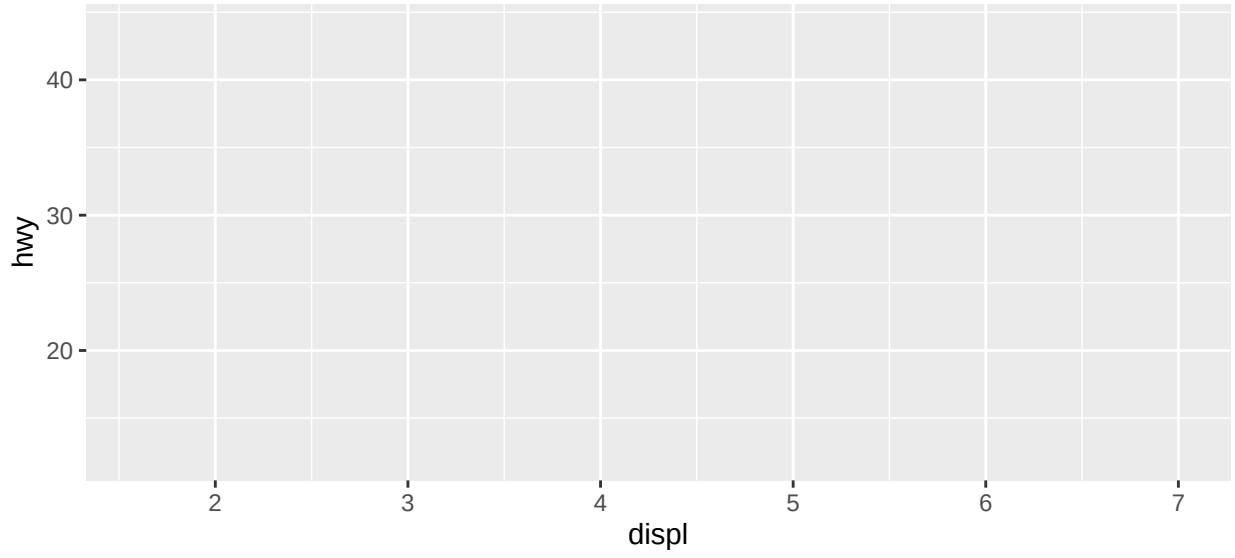
배기량과 고속도료연비 간의 관계

ggplot2 문법에 따라 3단계로 산점도를 그려, mpg 데이터에서 배기량(displ)과 고속도료연비(hwy) 간의 관계를 살펴보자.

(1단계) 배경설정

우선, 그래프를 그릴 배경을 만든다. 데이터를 지정하고 aes에 x축과 y축에 사용한 변수를 지정하면 배경이 만들어 진다.
data는 mpg, x축에는 displ 변수, y축에는 hwy 변수를 지정한다.

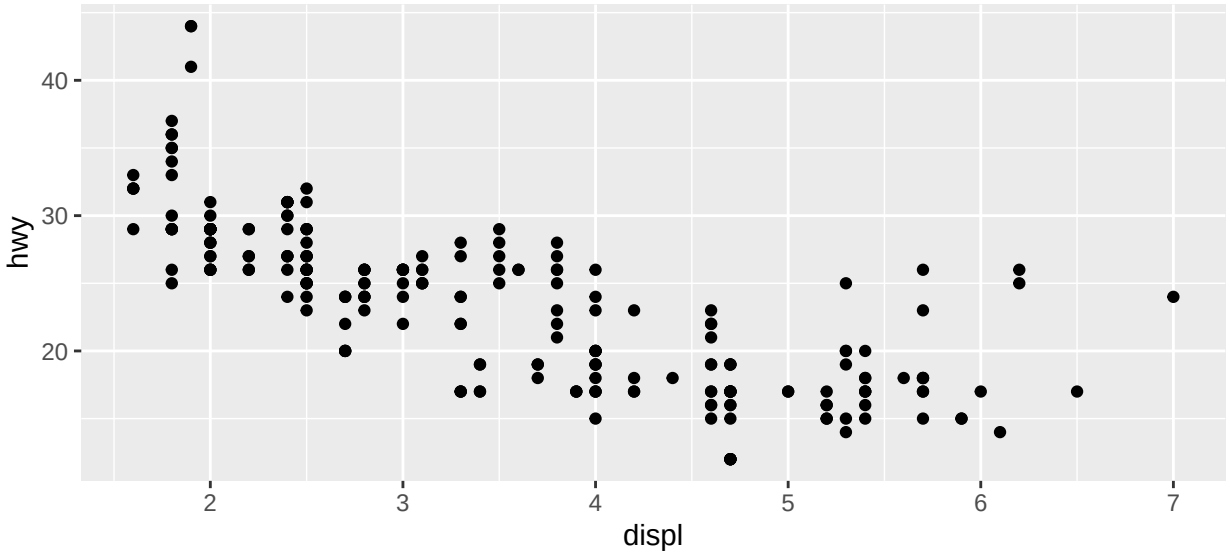
```
ggplot(data = mpg, aes(x = displ, y = hwy))
```



(2단계) 그래프 추가(점,막대,선)

배경 위에 + 기호를 이용해 그래프의 유형을 지정하는 함수를 추가한다. 여기서는 산점도를 그리는 함수 geom_point()를 추가한다.
+ 파이프(%>%) 연산자와 마찬가지로 + 먼저 쓰고 [Enter]키로 행을 바꿔 함수를 추가해야 한다.

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

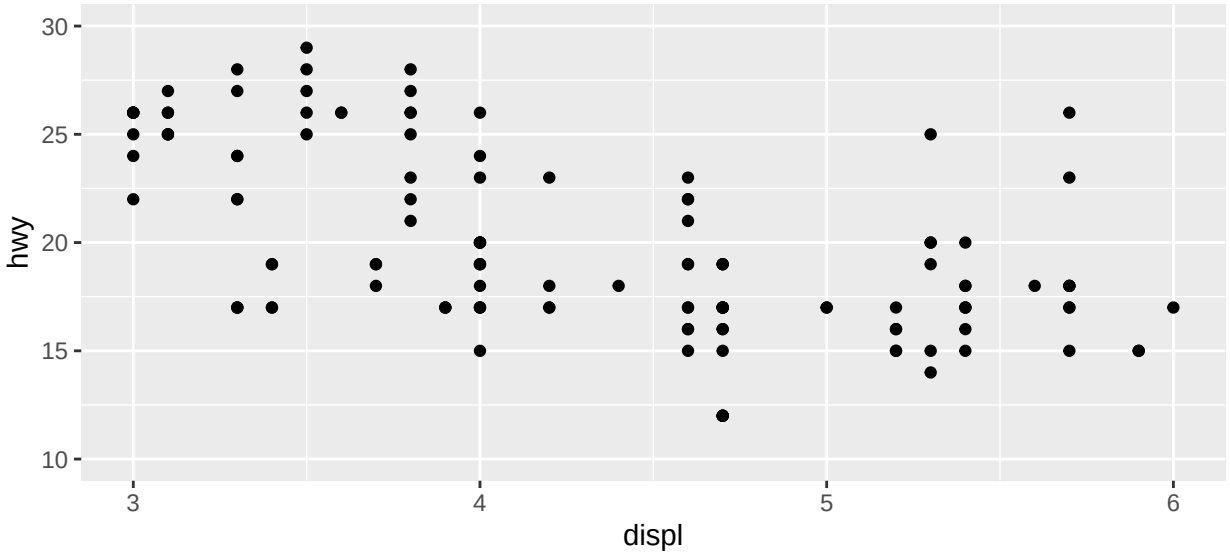


(3단계) 설정 추가

마지막으로 축의 범위나 색, 표식 등의 세부 설정을 지정한다. 여기서는 x축은 3~6까지 표현하고 y축은 10~30까지 표현하자.

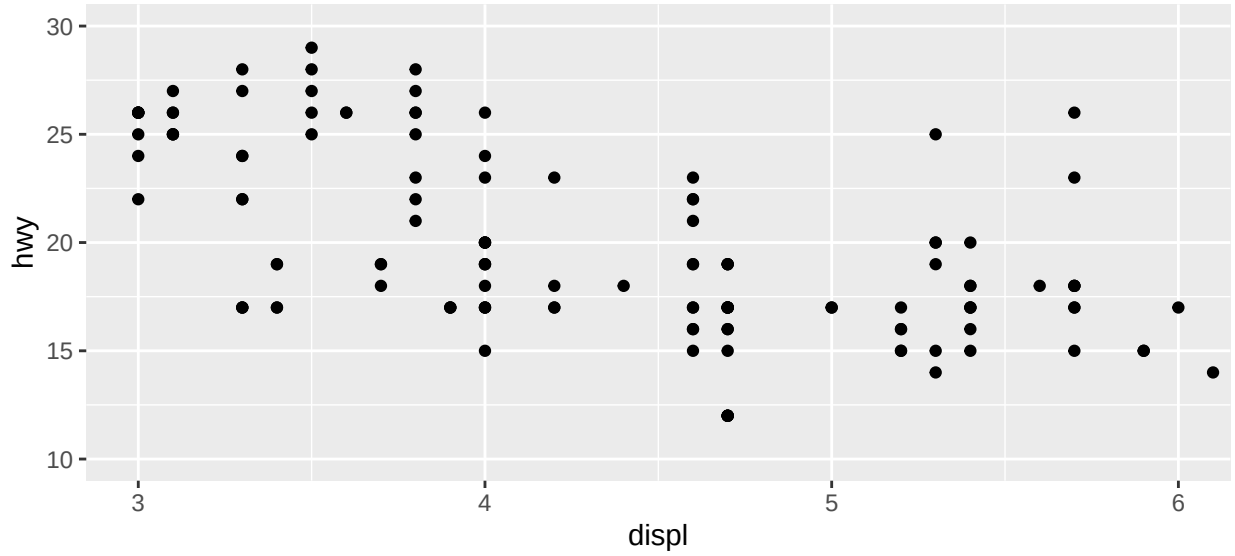
```
ggplot(data = mpg, aes(x = displ, y = hwy)) +
  geom_point() + xlim(3, 6) + ylim(10, 30)
```

```
## Warning: Removed 105 rows containing missing values (geom_point).
```



xlim(), yxlim()은 설정 역역을 벗어난 데이터포인트 삭제되어 추세선을 추가하면 결과가 왜곡되지만, **coord_cartesian()** 를 사용하면 데이터포인트를 삭제하지 않고 차트의 특정영역만 확대하여 나타낸다.

```
ggplot(data=mpg, aes(x = displ, y = hwy)) +
  geom_point() + coord_cartesian(xlim=c(3, 6), ylim=c(10, 30))
```

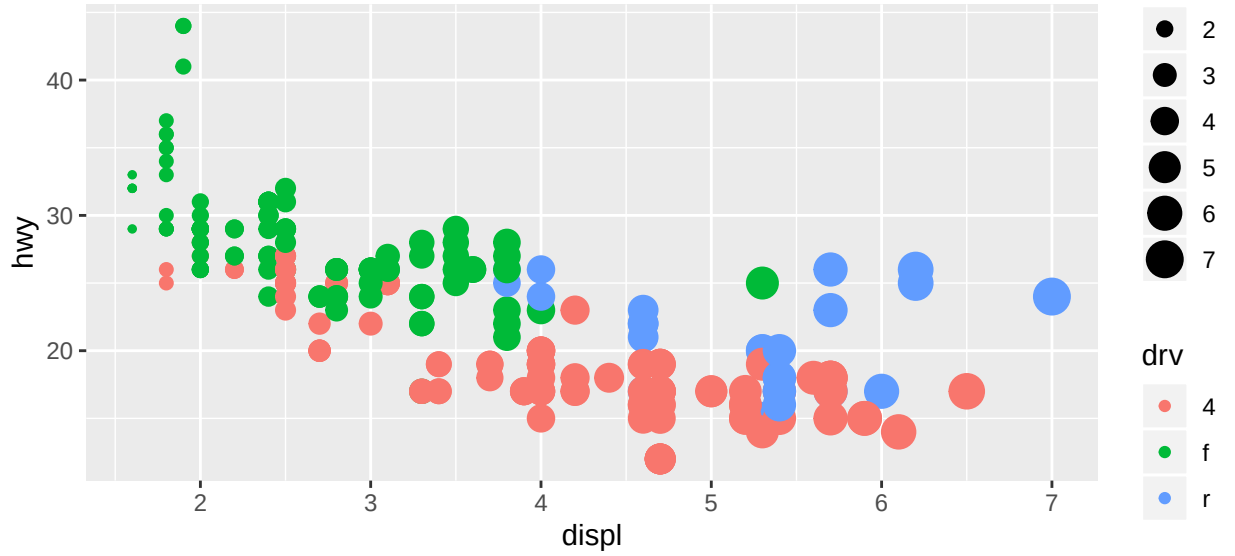


ggplot2를 이용하여 쉽고 짧은 문법으로 보기좋은 그래프를 작성할 수 있다.

미적요소 및 추세선 추가

좀더 보기좋게 아래 코드와 같이 산점도 표식의 크기와 색을 지정할 수도 있다.

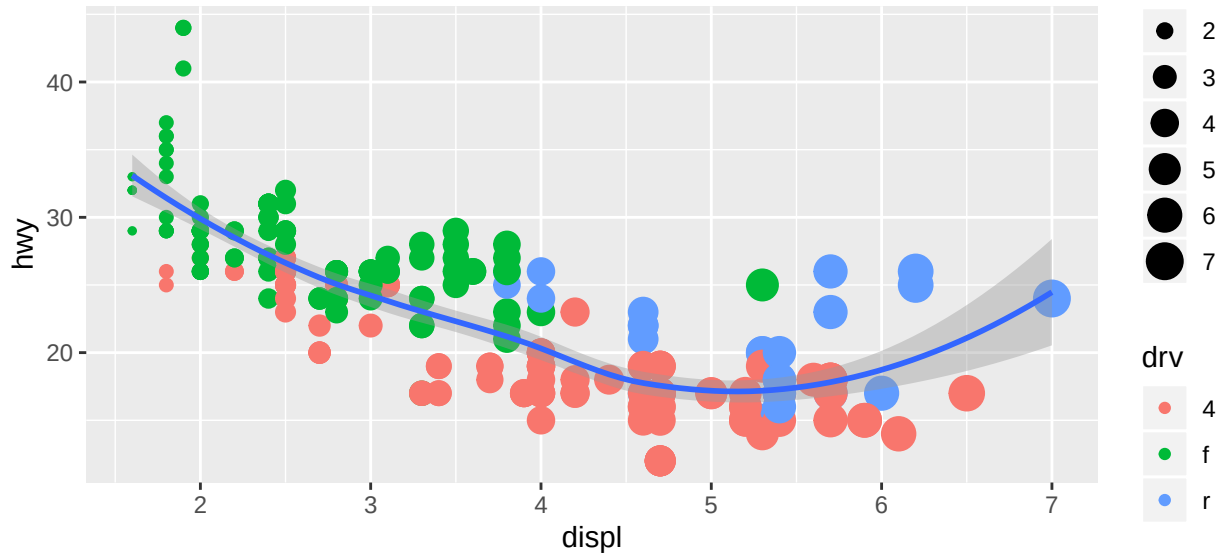
```
ggplot(data = mpg, aes(x = displ, y = hwy)) +
  geom_point(aes(color=drv, size=displ))
```



산점도 위에 추세선을 추가하면 값의 흐름이 어떠한지 경향을 파악할 수 있다.

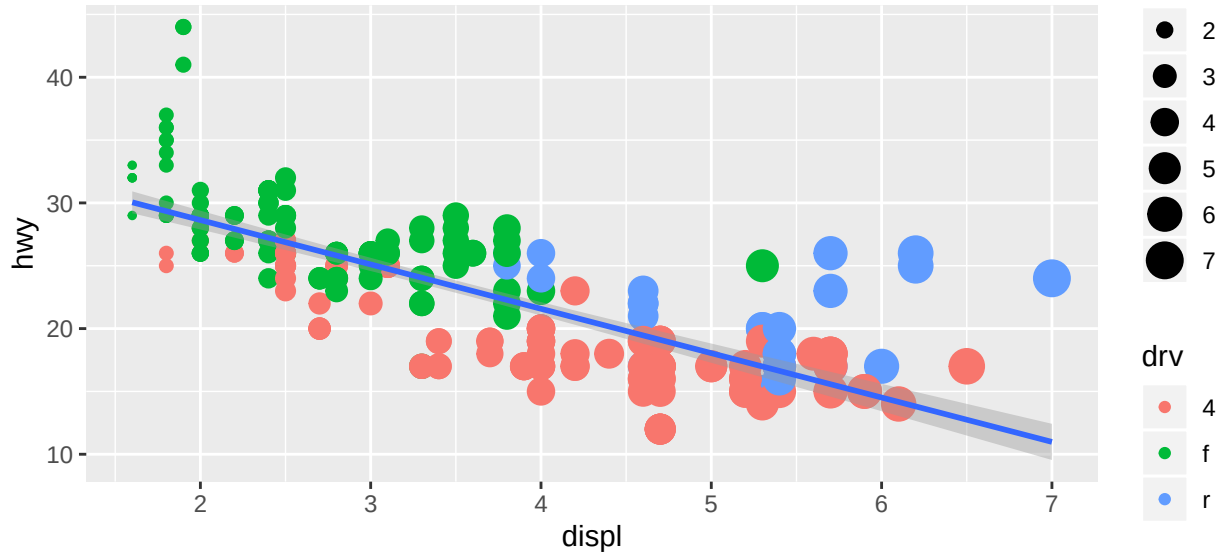
```
ggplot(data = mpg, aes(x = displ, y = hwy)) +
  geom_point(aes(color=drv, size=displ)) +
 stat_smooth()
```

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



ggplot2에서는 추세선을 그릴 때 국소회귀분석(loess) 방식을 사용하기 때문에 위의 그래프와 같이 부드러운 추세선이 나타난다. 직선 형태의 추세선을 그리려면 **stat_smooth(method='lm')** 을 사용하여 선형회귀분석(lm) 방식으로 그린다.

```
ggplot(data = mpg, aes(x = displ, y = hwy)) +
  geom_point(aes(color=drv, size=displ)) +
  stat_smooth(method='lm')
```



산점도와 추세선을 통해 배기량(displ)이 늘어날수록 고속도로연비(hwy)가 줄어드는 경향이 있음을 한눈에 파악할 수 있다

상관분석

상관계수

위의 그래프에서 관측값들의 분포가 추세선 가까운 것을 확인할 수 있다. 상관계수는 선형성의 정도를 나타내는 척도로서 **cor()**을 이용하여 구할 수 있다. 선형계수는 -1과 1 사이의 값으로 1이나 -1에 가까울 수록 두 변수 간의 상관성이 높은 것이다. 일반적으로 상관계수가 0.5보다 크거나 -0.5보다 작으면 상관성이 높다고 판단한다.

```
cor(mpg$displ, mpg$hwy)
```

```
## [1] -0.76602
```

배기량(displ)과 고속도로연비(hwy)의 상관계수가 -0.76602이므로 두 변수 간의 상관성이 높다고 할 수 있다.

회귀식

추세선을 식으로 나타낸 것을 **회귀식**이라고 하는데, 두 변수의 선형관계를 나타낸 선의 식으로 관측값들의 추세를 가장 잘 나타낼 수 있는 선이다. 회귀식을 이용하면 값을 예측할 수 있다.
회귀식은 **lm()**을 이용하여 구할 수 있다. lm()안에 ~ 를 기준으로 **'종속변수~독립변수'** 을 지정한다.
아래의 코드는 배기량(displ)에 따른 고속도로연비(hwy)의 회귀식을 구한 것이다.

```
lm( mpg$hwy~mpg$displ)
```

```
##
## Call:
## lm(formula = mpg$hwy ~ mpg$displ)
##
## Coefficients:
## (Intercept)    mpg$displ
##      35.698       -3.531
```

위의 결과를 통해 회귀식이 다음과 같음을 알 수 있다.

$y = -3.531x + 35.698$

배기량(displ)이 x에 해당하고 고속도로연비(hwy)가 y에 해당하므로, 배기량이 8, 9, 10인 자동차의 고속도로연비가 7.45, 3.919, 0.388 될 것이라고 예측할 수 있다.

```
(y <- -3.531*8 + 35.698)
```

```
## [1] 7.45
```

```
(y <- -3.531*9 + 35.698)
```

```
## [1] 3.919
```

```
(y <- -3.531*10 + 35.698)
```

```
## [1] 0.388
```

ggplot 그래프 저장

ggplot 그래프는 ggsave()를 사용하여 그림 파일로 저장할 수 있다.

```
gpt <- ggplot(mpg, aes(x=cyl)) + geom_bar()

ggsave("myggplot.png")            #
```

```
## Saving 6.5 x 3 in image
```

```
ggsave("myggplot.png", plot=gpt)  # gpt
```

```
## Saving 6.5 x 3 in image
```

막대 그래프(Bar Chart)

막대그래프는 데이터의 크기를 막대의 길이로 표현한 그래프로서 성별이나 소득의 차이처럼 집단 간 차이를 표현할 때 주로 사용한다.

차종별 빈도그래프

자동차 종류(class)별 빈도 즉, 값의 갯수를 막대그래프로 그려보자. 먼저, 빈도표를 통해 차종별로 자동차 대수를 살펴보면 다음과 같다.

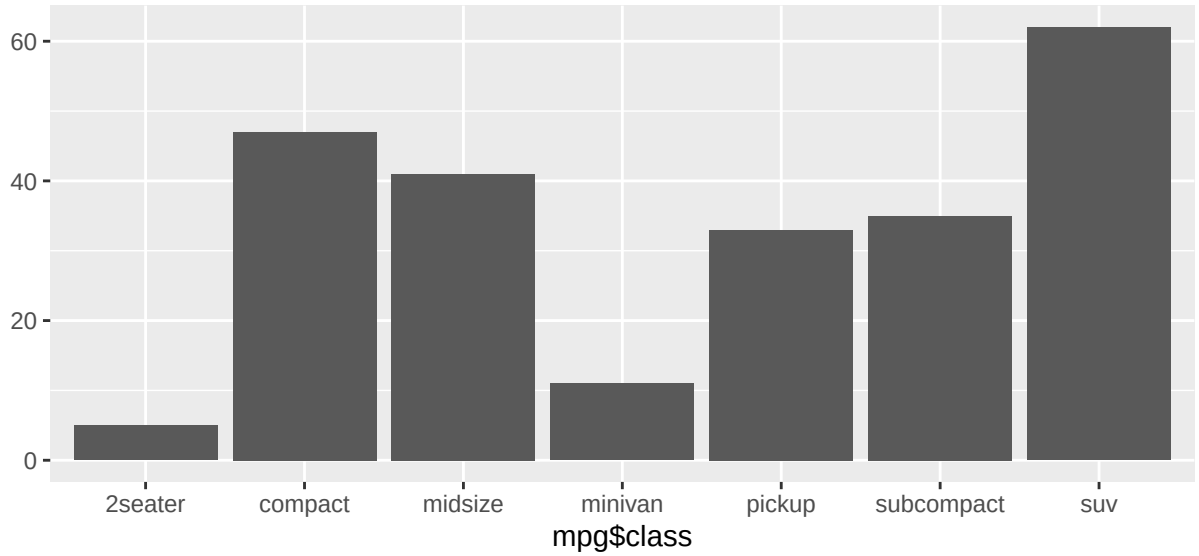
```
table(mpg$class)
```

```
##
##    2seater    compact    midsize    minivan     pickup subcompact
##          5         47         41         11         33         35
##        suv
##         62
```

(방법1) qplot() 사용

앞에서 배웠던 **qplot()**을 사용하여 차종(class)별 빈도그래프를 그려보자.

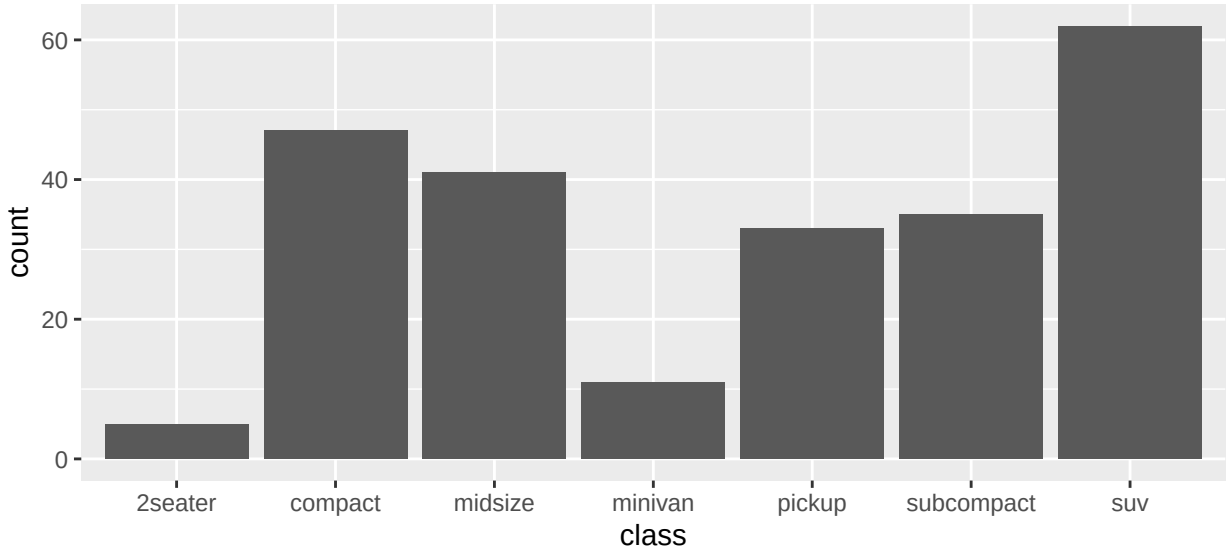
```
qplot(mpg$class)
```



(방법2) qqplot() + geom_bar() 사용

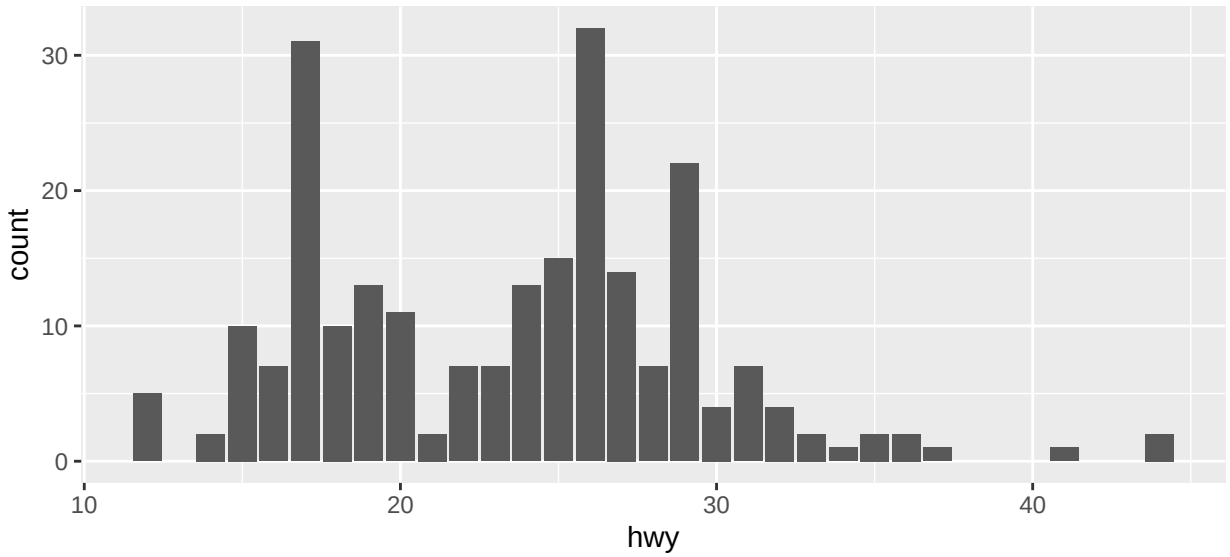
ggplot2 패키지를 사용하면 ggolot() 함수에 **geom_bar()**을 추가하면 쉽게 빈도그래프를 작성할 수 있다. 이때 y축없이 x축만 지정한다.
x축에 자동차 종류(class)와 같은 범주형 변수가 지정되어 있는 경우 해당 변수 값의 개수를 나타내는 빈도그래프가 만들어진다.

```
ggplot(mpg, aes(x=class)) + geom_bar()
```



x축에 고속도로연비(hwy)와 같은 연속형 변수가 지정되어 있는 경우는 해당 변수 값의 분포를 나타내는 히스토그램이 만들어진다.

```
ggplot(mpg, aes(x=hwy)) + geom_bar()
```



자동차 종류(class)에 따른 고속도로연비(hwy)의 빈도표를 살펴보면 다음과 같다.

```
table(mpg$hwy)
```

```
##
## 12 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37
##  5  2 10  7 31 10 13 11  2  7  7 13 15 32 14  7 22  4  7  4  2  1  2  2  1
## 41 44
##  1  2
```

(방법3) ggplot() + geom_col() 사용

geom_bar() 를 사용하지 않고 **geom_col()** 함수를 사용할 수도 있다. 단순히 geom_bar()만 연결하면 아래와 같이 **오류**가 발생한다.
geom_col()는 **x축과 y축 모두 변수를 지정**해야 한다.

```
ggplot(mpg, aes(x=class)) + geom_col()
 # Error :
```

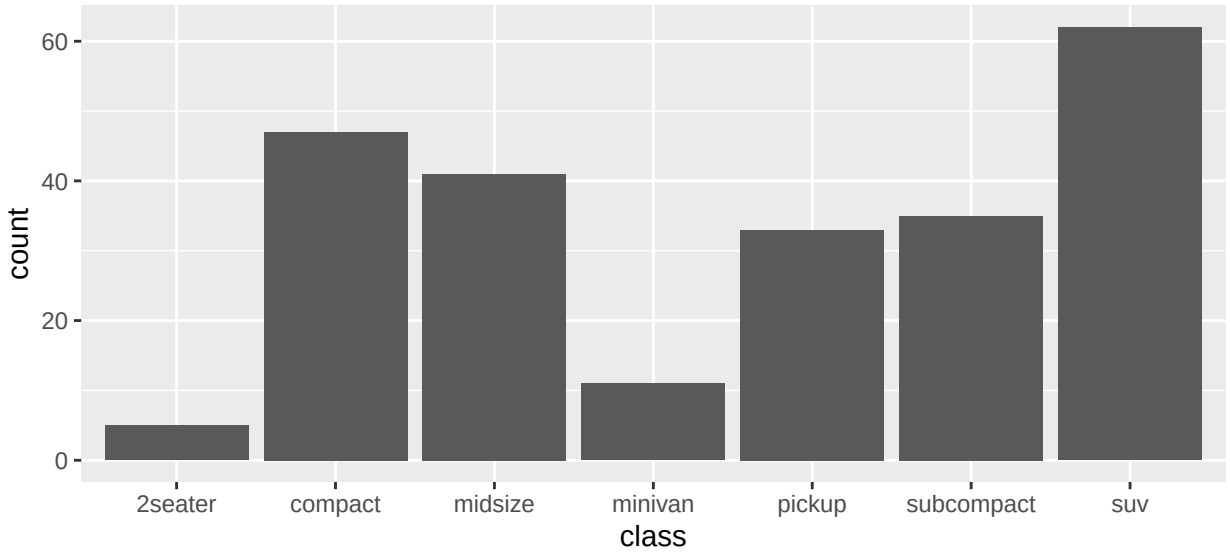
geom_col()을 이용하여 빈도를 나타내는 막대그래프를 작성하기 위해서는 먼저, 그래프로 작성하고자 하는 변수에 대한 요약통계를 계산한다.

```
df_mpg <- mpg %>% group_by(class) %>% summarise(count=n())
df_mpg
```

```
## # A tibble: 7 x 2
##   class      count
##   <chr>      <int>
## 1 2seater        5
## 2 compact       47
## 3 midsize       41
## 4 minivan       11
## 5 pickup        33
## 6 subcompact    35
## 7 suv           62
```

계산된 값을 **y축**의 변수로 지정하여 빈도 막대그래프를 작성한다.

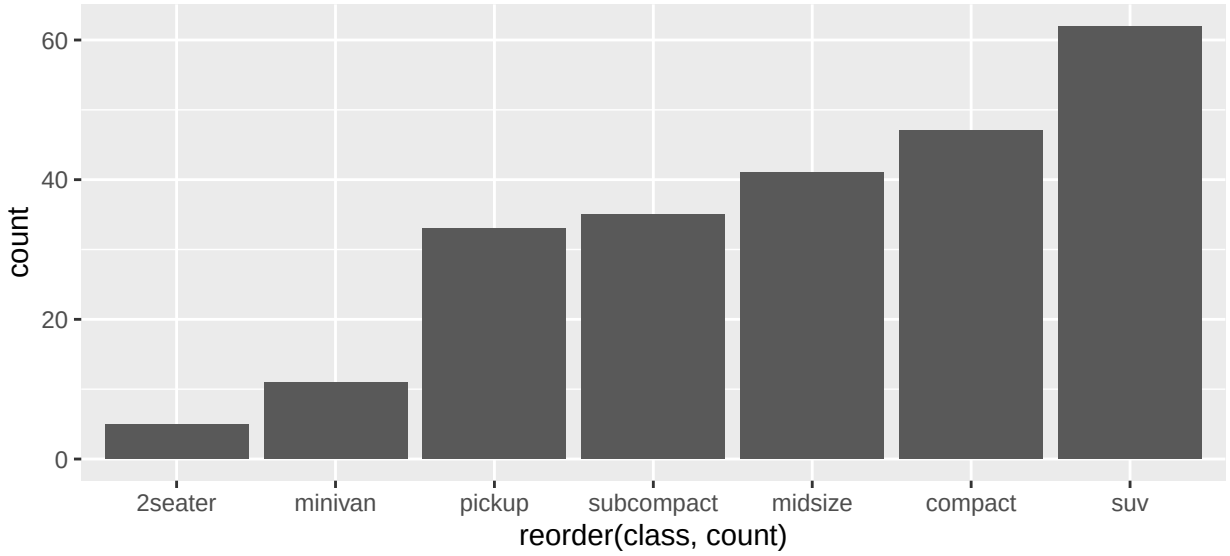
```
ggplot(df_mpg, aes(x=class, y=count)) + geom_col()
```



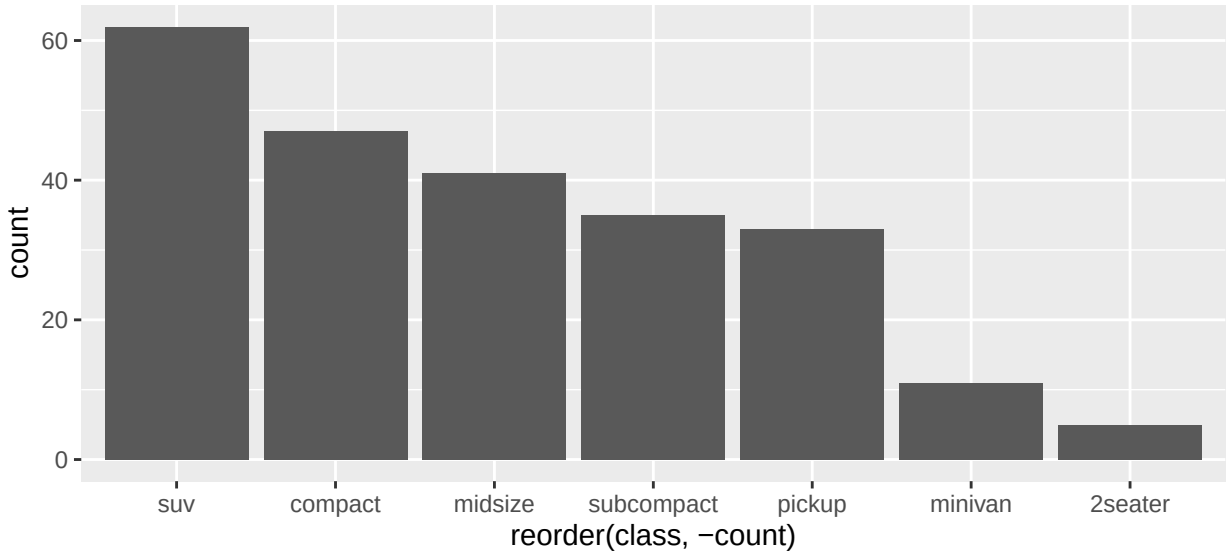
크기순으로 정렬하여 그래프 그리기

막대그래프는 기본적으로 x축 값의 크기순(알파벳순)으로 정렬된다. **reorder()**를 사용하면 막대를 크기순으로 정렬할 수 있다. reoder()에 x축 변수, 정렬기준이 되는 변수를 차례대로 지정한다. 정렬기준 변수 앞에 - 기호를 붙이면 내림차순으로 정렬된다.

```
ggplot(df_mpg, aes(x=reorder(class, count), y=count)) + geom_col()
```




```
ggplot(df_mpg, aes(x=reorder(class, -count), y=count)) + geom_col()
```



※ geom_bar() .vs. geom_col()

geom_bar() 원자료를 이용해 빈도를 그래프로 작성
geom_col() 요약값 계산 후 그래프로 작성(크기순 정렬 가능)

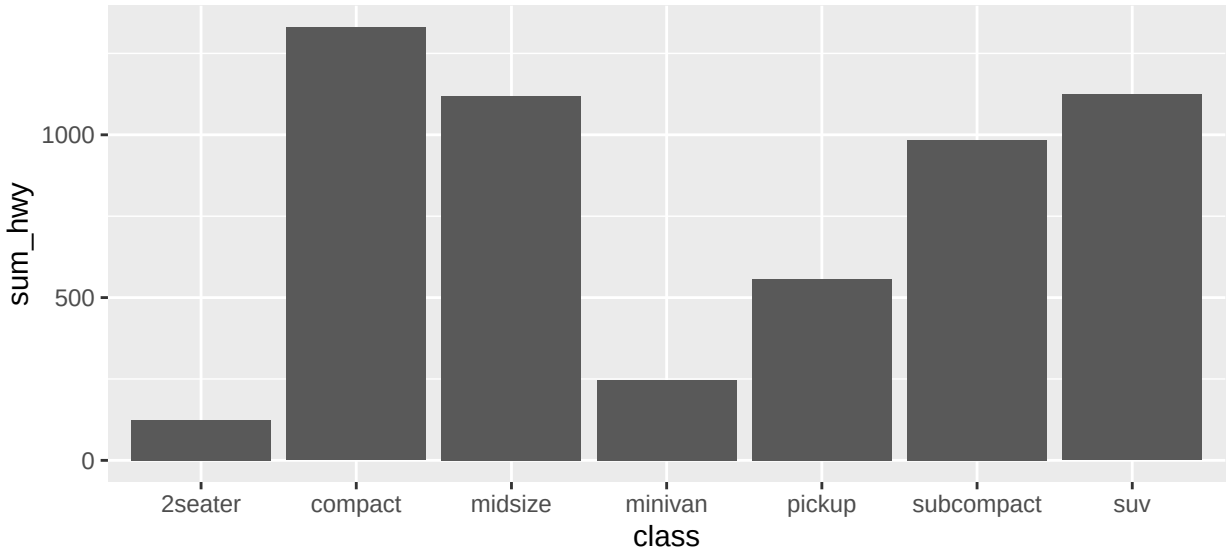
차종별 고속도로연비 합계를 나타내는 막대그래프

먼저, **geom_col()** 사용하여 차종(class) 별 고속도로연비(hwy)의 합계를 막대그래프를 그려본다. geom_col()은 x축과 y축 모두에 변수를 지정해야 한다.

```
df_mpg <- mpg %>% group_by(class) %>% summarise(sum_hwy=sum(hwy))
df_mpg
```

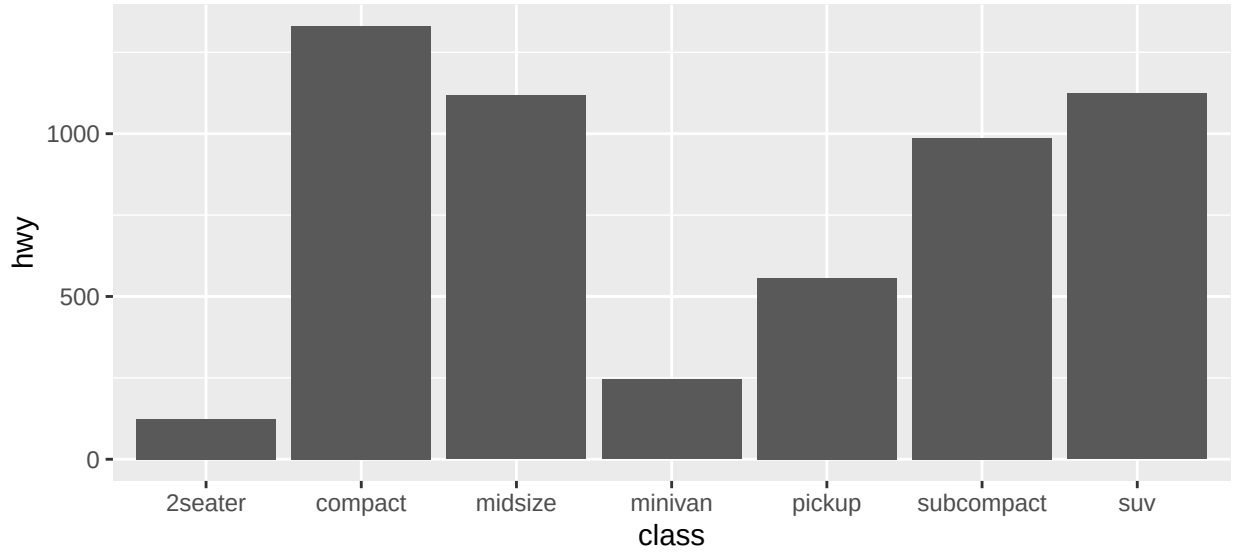
```
## # A tibble: 7 x 2
##   class      sum_hwy
##   <chr>        <int>
## 1 2seater        124
## 2 compact       1330
## 3 midsize       1119
## 4 minivan        246
## 5 pickup         557
## 6 subcompact     985
## 7 suv           1124
```

```
ggplot(df_mpg, aes(x=class, y=sum_hwy)) + geom_col()
```



평균, 갯수, 최대값 등의 요약통계가 아닌 **합계**에 대한 막대그래프를 작성할 경우 아래와 같이 y축에 hwy 변수만 지정해도 된다.

```
ggplot(mpg, aes(x=class, y=hwy)) + geom_col()
```



geom_bar()를 사용하여 합계와 같은 막대그래프를 그리려면 **stat='identity'** 옵션을 사용해야 한다. geom_bar()는 그래프를 히스토그램화하는 기본 속성을 가지고 있기 때문이다.

```
ggplot(mpg, aes(x=class, y=hwy)) + geom_bar(stat='identity')
```

※ 사용하기 편한 것 선택

goem_col() 빈도(범주) 또는 값을 막대그래프 표현하기 수월

제조회사별 평균 고속도로연비를 나타내는 막대그래프

geom_col()을 사용하여 제조회사(manufacturer)별 평균 고속도로연비(hwy)에 대한 막대그래프를 그리려면, 먼저 제조회사별 평균 고속도로연비를 계산해야 한다.

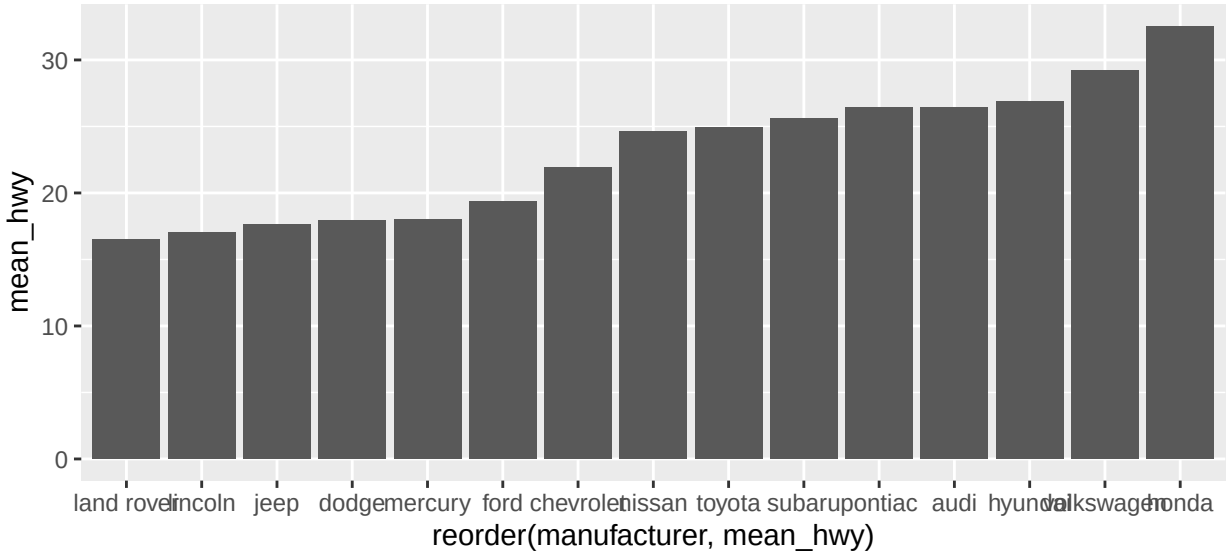
```
df_mpg <- mpg %>% group_by(manufacturer) %>%
  summarise(mean_hwy=mean(hwy))
df_mpg
```

```
## # A tibble: 15 x 2
##    manufacturer mean_hwy
##    <chr>           <dbl>
##  1 audi             26.4
##  2 chevrolet        21.9
##  3 dodge            17.9
##  4 ford             19.4
##  5 honda            32.6
##  6 hyundai          26.9
##  7 jeep             17.6
##  8 land rover       16.5
##  9 lincoln          17
## 10 mercury          18
```

```
## 11 nissan           24.6
## 12 pontiac          26.4
## 13 subaru           25.6
## 14 toyota           24.9
## 15 volkswagen       29.2
```

계산된 평균값을 y축 변수에 지정하고 값의 크기 순으로 오름차순으로 정렬하여 그래프를 그리면 다음과 같다.

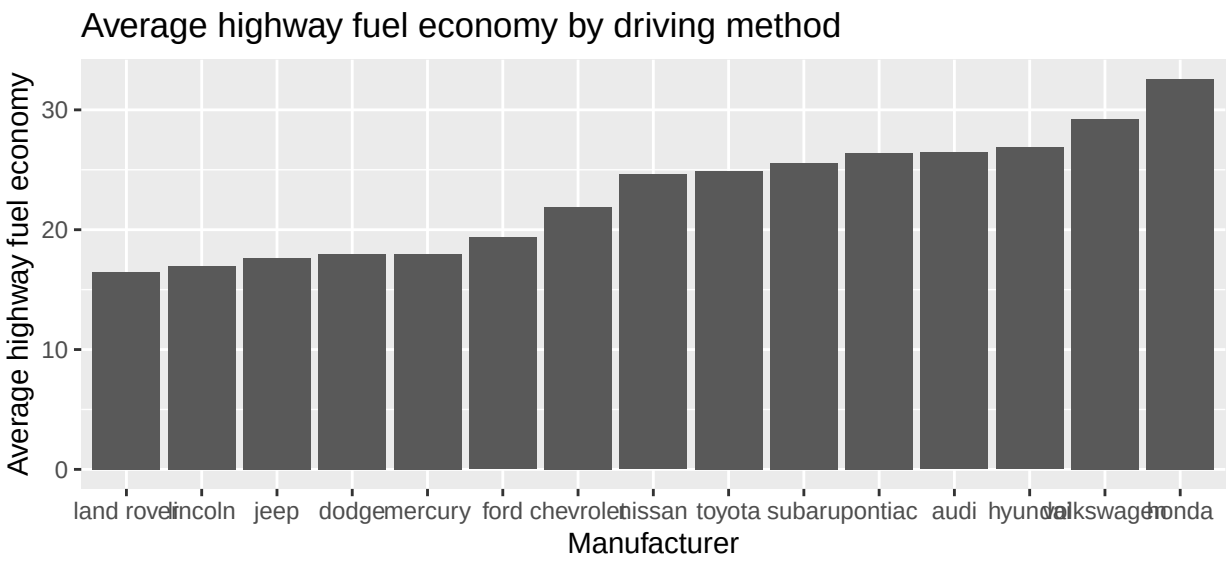
```
ggplot(df_mpg, aes(x=reorder(manufacturer, mean_hwy), y=mean_hwy)) +
  geom_col()
```



좀 더 그래프를 꾸며보자.

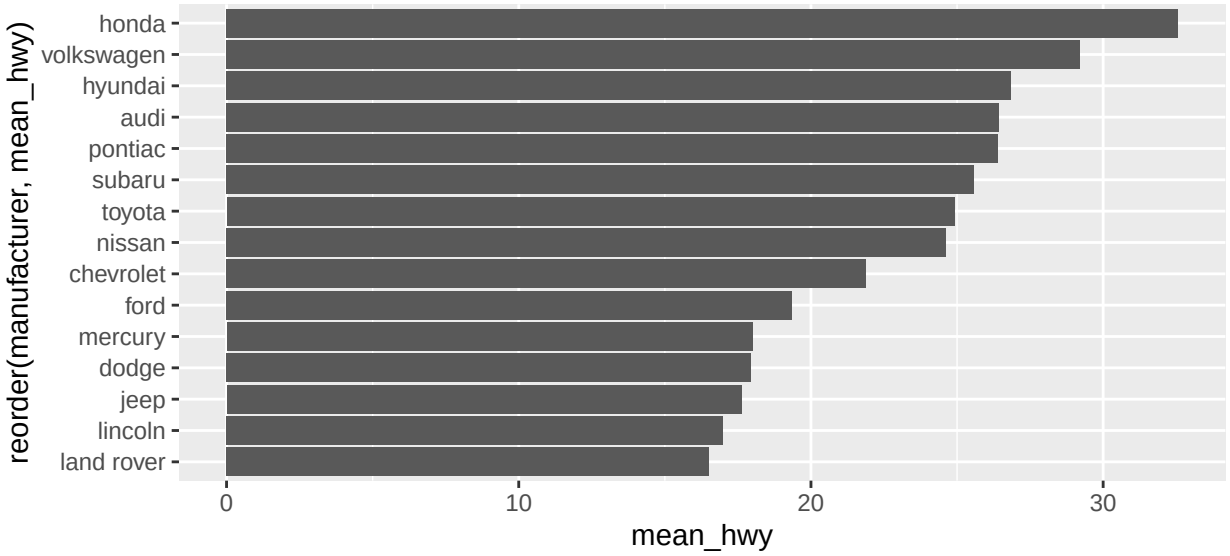
그래프의 제목과 x축 및 y축의 제목을 달기 위해 **labs()**를 사용한다.

```
ggplot(df_mpg, aes(x=reorder(manufacturer, mean_hwy), y=mean_hwy)) +
  geom_col() +
  labs(title="Average highway fuel economy by driving method",
       x="Manufacturer", y="Average highway fuel economy")
```



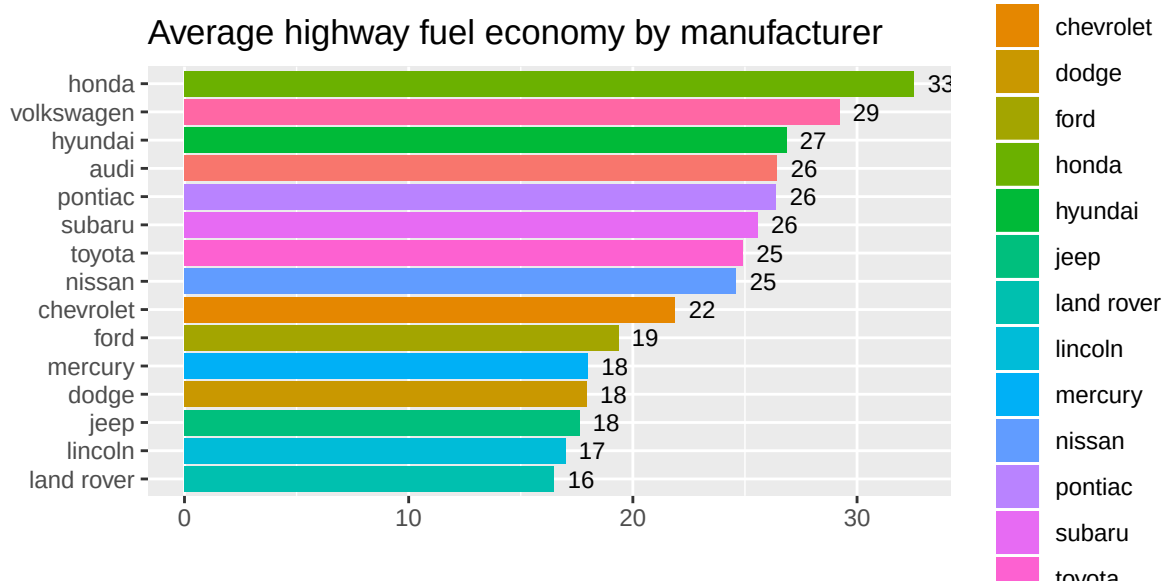
위의 그래프를 보면 y축의 제조회사명이 겹쳐져 잘 보이지 않는다. coord_flip()를 추가하여 가로 막대그래프를 그릴 수 있다.

```
ggplot(df_mpg, aes(x=reorder(manufacturer, mean_hwy), y=mean_hwy)) +
  geom_col() +
  coord_flip()
```



geom_text()을 추가하면 각 막대그래프의 값을 그래프에 나타낼 수 있다. 아래 R코드에서 x축과 y축 제목을 보이지 않도록 하기 위해 ""을 지정했다.

```
ggplot(df_mpg, aes(x=reorder(manufacturer, mean_hwy),y=mean_hwy)) +
  geom_col(aes(fill=manufacturer)) +
  coord_flip() +
  geom_text(aes(label=round(mean_hwy)), hjust=-0.5, size=3) +
  labs(title="Average highway fuel economy by manufacturer", x="", y="")
```



선 그래프(Line Chart)

선그래프는 데이터를 선으로 표현한 그래프로, 환율이나 주가지수와 같이 시간에 따라 달라지는 값을 그래프를 표현할 때 이용한다. 일정 시간 간격을 두고 나열된 데이터를 **시계열 데이터**라고 하고, 시계열 데이터를 선으로 표현한 그래프를 **시계열 그래프**라고 한다.

시간에 따른 실업자수의 변화

시간의 변화에 따라 실업자 수가 어떻게 변하는지 시계열 그래프를 작성하여 살펴보자.
x축은 시간을 의미하는 date 변수, y축은 실업자 수를 의미하는 unemploy 변수를 지정하고 선그래프로 표현하기 위해 geom_line()을 추가한다.
먼저, economics 자료구조를 데이터프레임 형태로 변경하자. economics의 자료구조는 **tibble**이다. tibble은 data.frame을 현대적으로 재구성한 것으로 대용량 데이터를 다루는데 보다 편리하다.

```
economics      # tibble
```

```
## # A tibble: 574 x 6
##    date         pce    pop psavert uempmed unemploy
##    <date>     <dbl>  <dbl>   <dbl>   <dbl>    <dbl>
##  1 1967-07-01  507. 198712    12.6     4.5     2944
##  2 1967-08-01  510. 198911    12.6     4.7     2945
##  3 1967-09-01  516. 199113    11.9     4.6     2958
##  4 1967-10-01  512. 199311    12.9     4.9     3143
##  5 1967-11-01  517. 199498    12.8     4.7     3066
##  6 1967-12-01  525. 199657    11.8     4.8     3018
##  7 1968-01-01  531. 199808    11.7     5.1     2878
##  8 1968-02-01  534. 199920    12.3     4.5     3001
##  9 1968-03-01  544. 200056    11.7     4.1     2877
## 10 1968-04-01  544  200208    12.3     4.6     2709
## # ... with 564 more rows
```

```
str(economics)
```

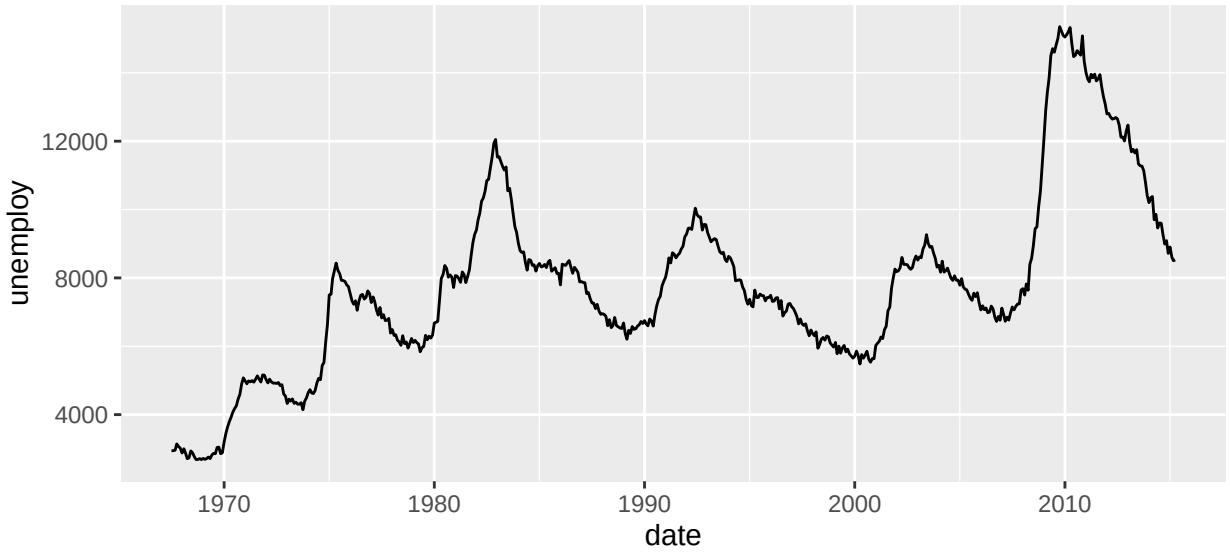
```
## Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 574 obs. of  6 variables:
##  $ date    : Date, format: "1967-07-01" "1967-08-01" ...
##  $ pce     : num  507 510 516 512 517 ...
##  $ pop     : num  198712 198911 199113 199311 199498 ...
##  $ psavert : num  12.6 12.6 11.9 12.9 12.8 11.8 11.7 12.3 11.7 12.3 ...
##  $ uempmed : num  4.5 4.7 4.6 4.9 4.7 4.8 5.1 4.5 4.1 4.6 ...
##  $ unemploy: num  2944 2945 2958 3143 3066 ...
```

```
eco <- as.data.frame(economics)  #     eco
str(eco)
```

```
## 'data.frame':    574 obs. of  6 variables:
##  $ date    : Date, format: "1967-07-01" "1967-08-01" ...
##  $ pce     : num  507 510 516 512 517 ...
##  $ pop     : num  198712 198911 199113 199311 199498 ...
##  $ psavert : num  12.6 12.6 11.9 12.9 12.8 11.8 11.7 12.3 11.7 12.3 ...
##  $ uempmed : num  4.5 4.7 4.6 4.9 4.7 4.8 5.1 4.5 4.1 4.6 ...
##  $ unemploy: num  2944 2945 2958 3143 3066 ...
```

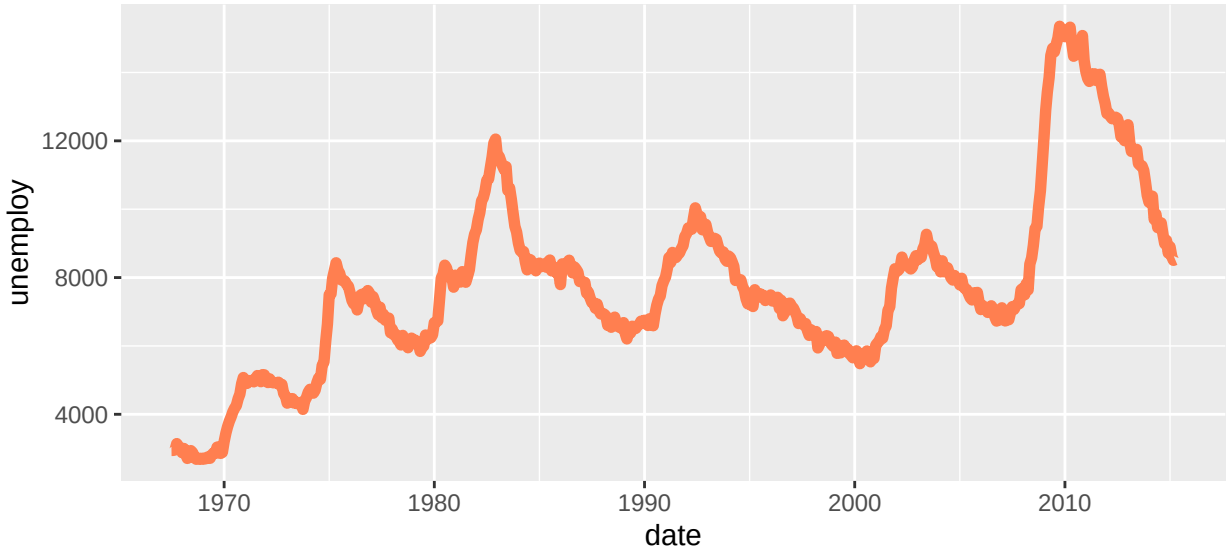
다음과 같이 eco 데이터와 geom_line()을 이용하여 선 그래프를 작성한다.

```
ggplot(eco, aes(x = date, y = unemploy)) + geom_line()
```



위의 그래프에서 선의 색과 두께를 변경해 보자.

```
ggplot(eco, aes(x = date, y = unemploy)) + geom_line(color='coral', lwd=2)
```



상자 그림(Box Plot)

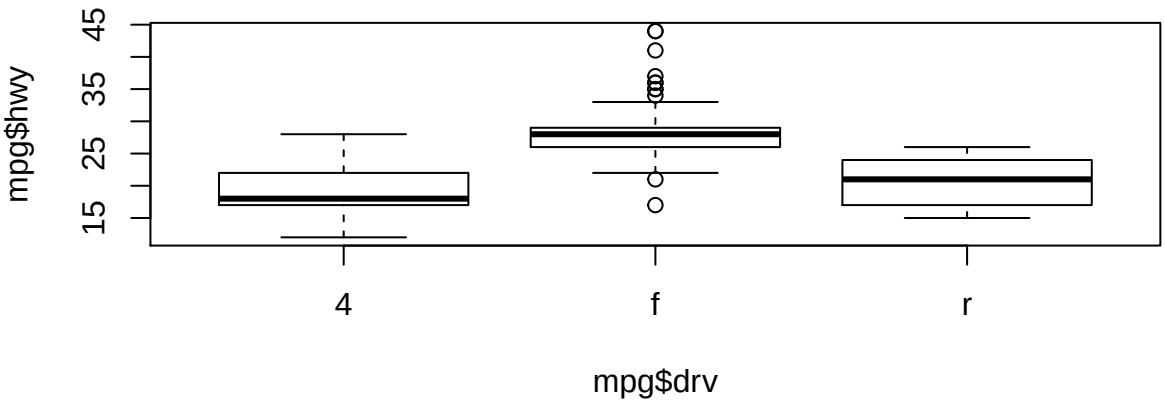
상자그림은 데이터가 퍼져있는 형태 즉, 분포를 직사각형 모양으로 표현한 그래프이다. 평균만 볼 때보다 데이터의 특징을 더 잘 이해할 수 있다.

구동방식별 고속도로연비의 데이터분포 확인

상자그림을 통해, mpg 데이터의 고속도로연비(hwy) 변수값이 어떤 분포를 가지고 있는지 구동방식(drv)별로 살펴보자.

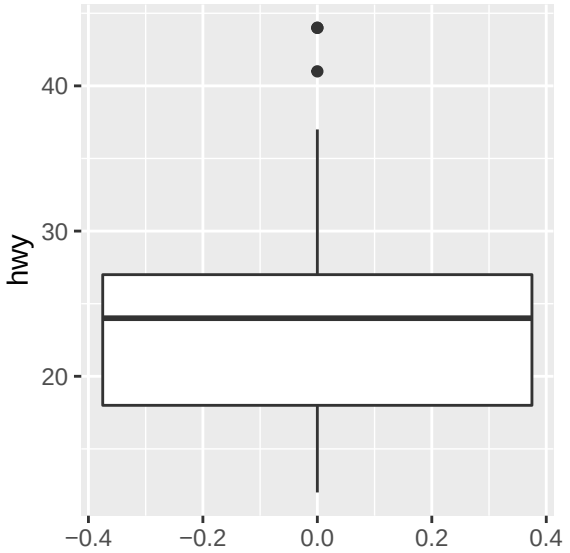
앞서 데이터를 전체적으로 살펴볼 때 **boxplot()**이용하여 이상값 여부와 데이터의 분포를 살펴보았다. 그룹별로 상자그림을 그리려면 ~ 을 사용하여 그룹 변수를 지정한다.

```
boxplot(mpg$hwy ~ mpg$drv)
```



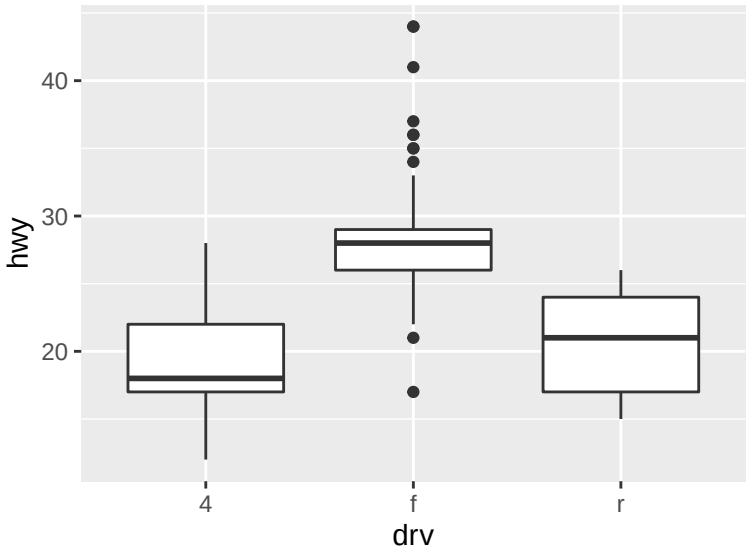
ggplot2패키지를 사용하여 상자그림을 그리려면, ggplote()에 **geom_boxplot()**을 추가한다. y축에 상자그림의 대상 변수를 지정해야 한다.

```
ggplot(mpg, aes(y=hwy)) + geom_boxplot()
```



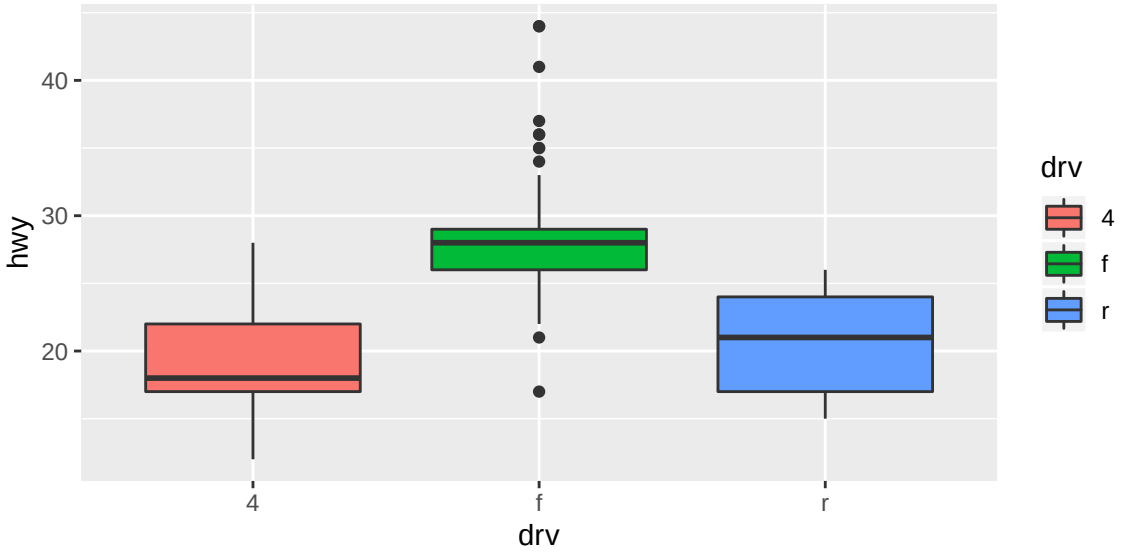
그룹별로 상자그림을 그리기 위해 x축에 그룹 변수를 지정한다.

```
ggplot(mpg, aes(x=drv, y=hwy)) + geom_boxplot()
```



위의 상자그림에서 색을 지정하면 다음과 같다.

```
ggplot(mpg, aes(x=drv, y=hwy, fill=drv)) + geom_boxplot()
```



자동차 종류(class)별로 도시연비(cty)의 차이를 상자그림으로 비교해 보면 다음과 같다. 위의 R코드와 동일한 기능을 한다.

```
ggplot(mpg, aes(x=class, y=cty)) + geom_boxplot(aes(fill=class))
```