

## 7. 좌표계: Coordinate system

---

- 좌표계: 시각적 요소  $x$ 와  $y$ 를 근거로 그래프의 각 요소의 2차원 위치를 결정하는 체계
- 좌표계의 종류
  - `coord_cartesian( )`: 디폴트
  - `coord_flip( )`
  - `coord_polar( )`

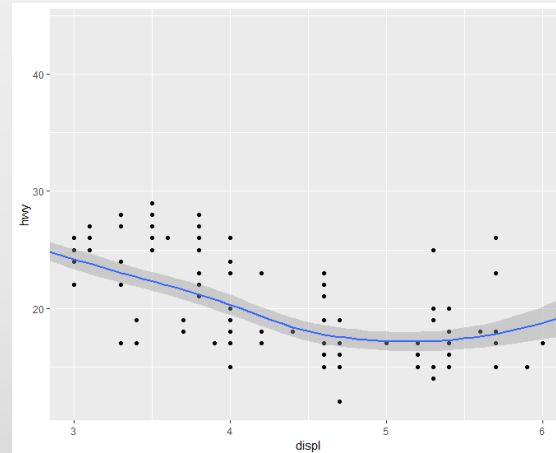
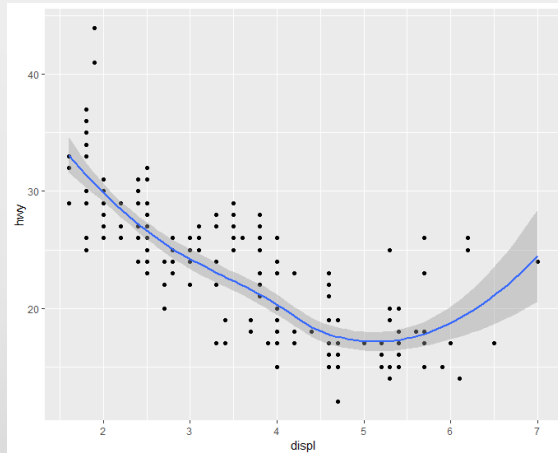
- coord\_cartesian( )의 활용: XY축 범위 조정

- 예: mpg에서 displ과 hwy의 산점도에 비모수 회귀곡선 추가한 그래프 작성.  
X축의 범위를 (3,6)으로 축소한 그래프 작성.

```
> p <- ggplot(data=mpg, mapping=aes(x=displ,y=hwy)) +  
  geom_point() + geom_smooth()
```

```
> p
```

```
> p + coord_cartesian(xlim=c(3,6))
```



- scale에 의한 XY축 범위 조정

- scale: 자료와 시각적 요소의 매핑 및 XY축과 범례 등의 내용 조정을 의미
- 대부분의 경우 디폴트 상태에서 그래프 작성
- XY축 범위 조정, XY축 라벨 변경이 필요한 경우에는 scale 함수를 사용하여 조정
- scale 함수의 일반적인 형태: `scale_*1*_*2*()`
  - \*1\*: 수정하고자 하는 시각적 요소; color, x, y, fill 등등
  - \*2\*: 적용되는 scale 지칭; discrete, continuous 등등
- 예: 연속형 X 변수의 범위 (3,6)으로 수정: `scale_x_continuous(limits=c(3,6))`  
연속형 X축의 라벨 변경: `scale_x_continuous(name="Engine")`
- 간편 함수:
  - XY 축 범위 조정: `xlim(3,6)`, `ylim(0,1)`
  - XY축 라벨 변경: `xlab("Engine")`, `ylab( )`, `labs(x="Engine")`

- 예: XY축 조정 비교

- 1) 함수 `xlim()`에 의한 조정

범위를 벗어난 자료: 삭제

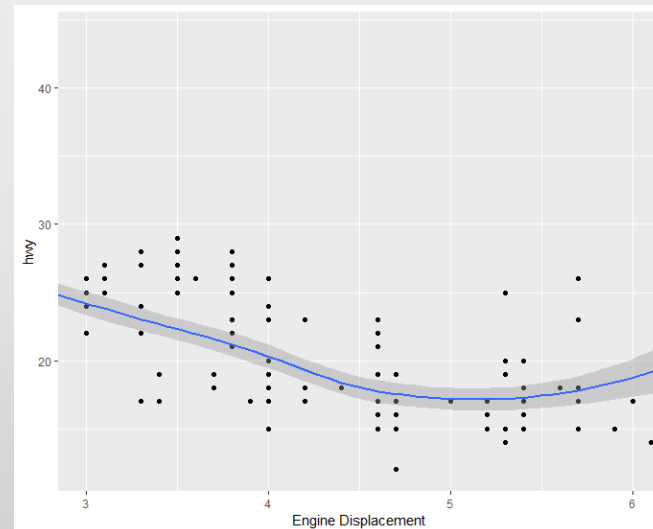
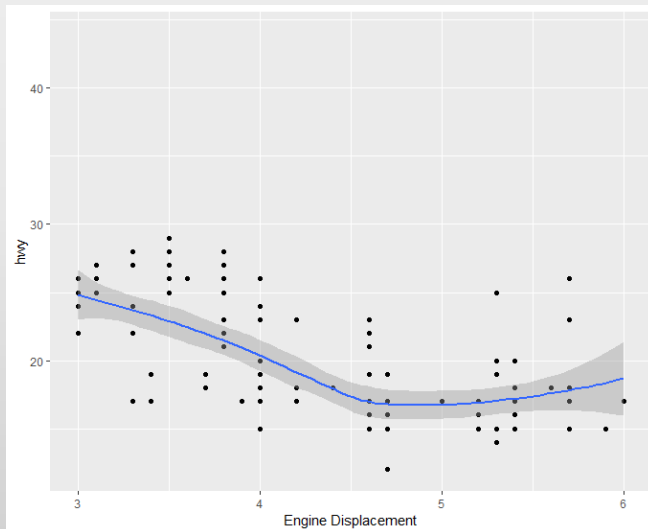
```
> p + xlim(3,6) + xlab("Engine Displacement")
```

Warning messages:

```
1: Removed 105 rows containing non-finite values (stat_smooth).  
2: Removed 105 rows containing missing values (geom_point).
```

- 2) 함수 `coord_cartesian()`에 의한 조정

```
> p + coord_cartesian(xlim=c(3,6)) +  
  xlab("Engine Displacement")
```



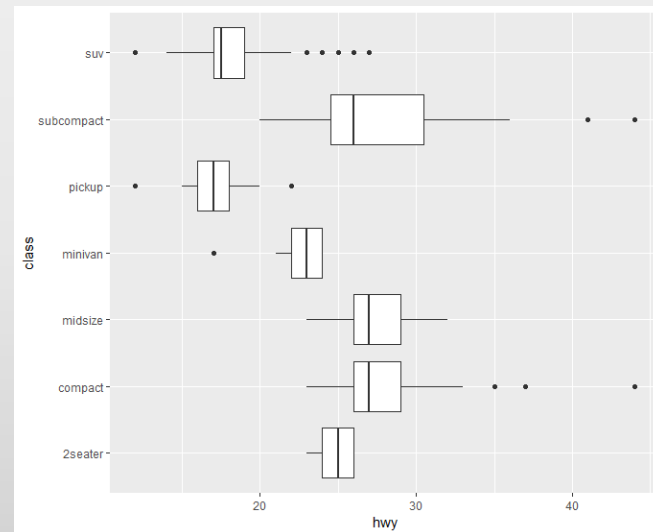
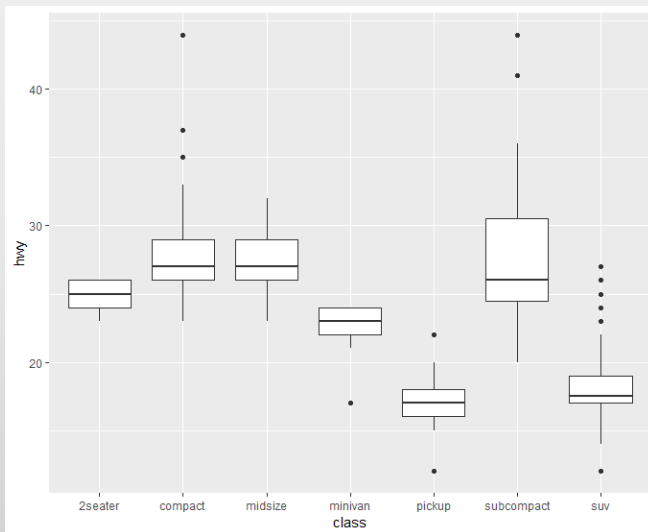
- 함수 `coord_flip()`의 활용: 평행한 상자그림 작성
  - 대부분의 geom 함수: 주어진 x 값에 대한 y의 분포 표현
  - 상자그림: 수직 방향의 작성되는 것이 디폴트
  - 수평 방향 상자그림: 디폴트 방향으로 작성하고, 그래프의 좌표를 90° 회전
  - 함수 `coord_flip()`: 작성된 그래프의 좌표 회전

- 예: mpg에서 class의 그룹별로 hwy의 상자그림 작성

- 상자그림: `geom_boxplot()`
- x 변수=class, y 변수=hwy

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

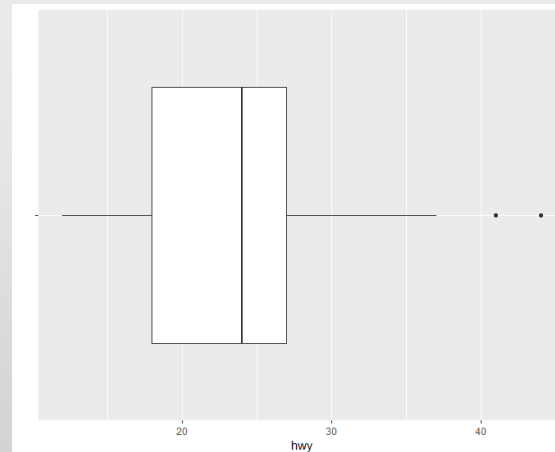
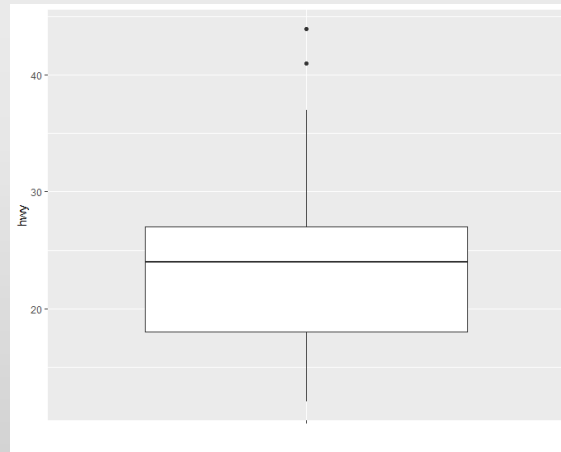
```
> ggplot(data=mpg, mapping=aes(x=class, y=hwy)) +  
  geom_boxplot() +  
  coord_flip()
```



- 예: 한 변수(hwy)의 상자그림 작성
  - 함수 `geom_boxplot()`에는 x와 y 모두 필요
  - x에는 하나의 값, y에는 연속형 변수 매핑

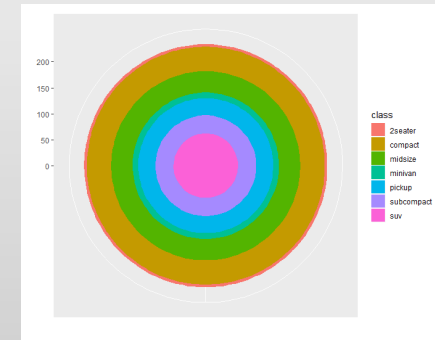
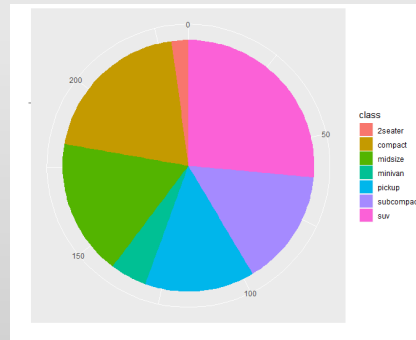
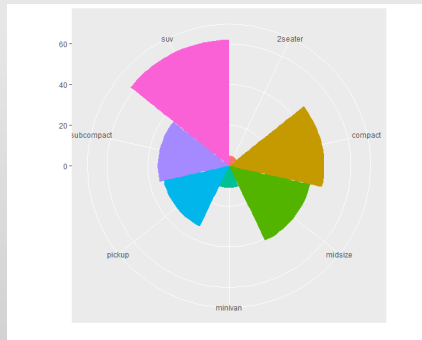
```
> ggplot(data=mpg, mapping=aes(x="", y=hwy)) +  
  geom_boxplot() +  
  xlab("")
```

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



- 함수 `coord_polar( )`의 활용: 파이 그래프 작성

- 극좌표(polar coordinate): 2차원 공간의 어느 한 점의 위치를 원점에서의 거리와 각도로 표현
- 함수 `coord_polar( )`: 데카르트 좌표를 극좌표로 변환
- 변수 `theta`: 시각적 요소 `x`와 `y` 중 각도로 변환할 요소 지정(디폴트는 `theta="x"`)
- 함수 `coord_polar( )`를 활용하여 막대 그래프에서 변형된 그래프
  - 1) Coxcomb 또는 Wind rose 그래프
  - 2) 파이 그래프
  - 3) Bullseye 그래프



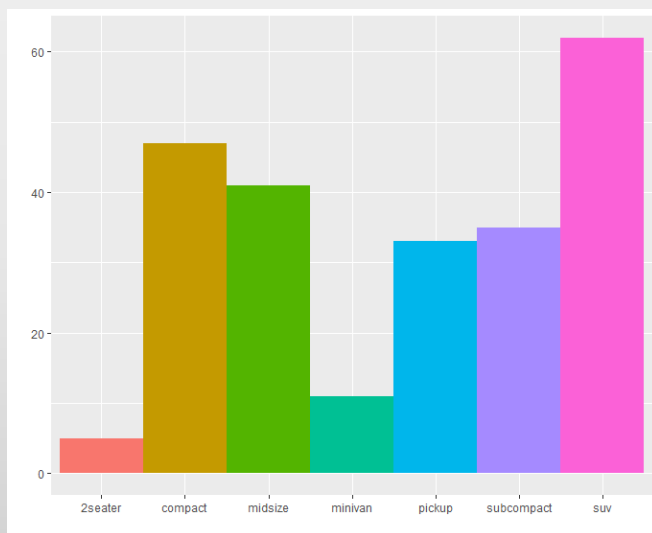


- 예제: mpg의 변수 class의 Coxcomb 그래프 작성

- 1) 막대 그래프 작성

- 각 막대마다 다른 색 사용
- 막대 사이 간격 제거
- 범례 제거

```
> b <- ggplot(data=mpg,mapping=aes(x=class, fill=class)) +  
  geom_bar(show.legend=FALSE, width=1) +  
  labs(x="",y="")  
> b
```

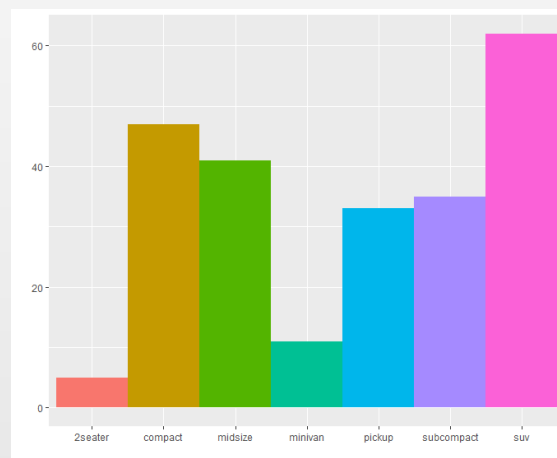
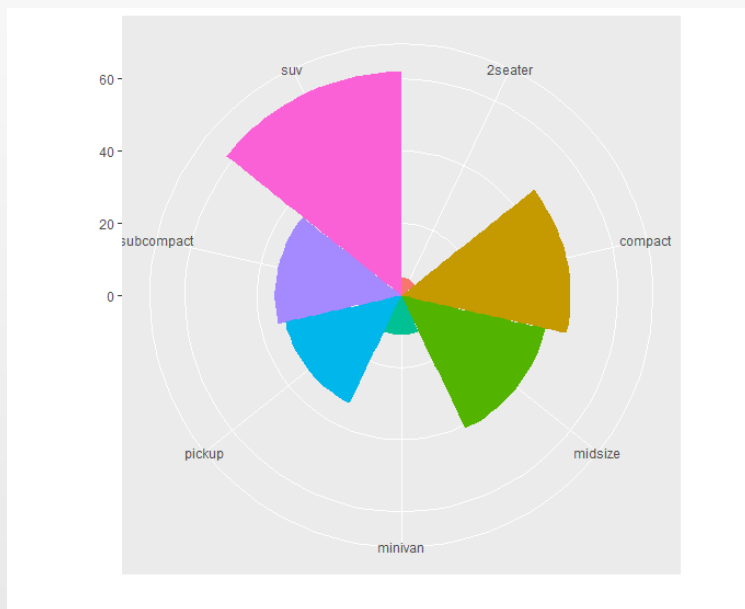


- show.legend=FALSE: 범례 제거
- width: 막대 폭 지정. 디폴트는 각 막대 구간의 90%의 차지하는 0.9.

## 2) theta="x"로 함수 coord\_polar( ) 실행

```
> b + coord_polar()
```

디폴트 theta="x"



Coxcomb 그래프(극좌표)

- 각 조각의 각도: theta="x"로 지정 → 동일한 각도
- 각 조각의 반지름: 각 막대의 높이에 비례

막대 그래프(데카르트 좌표)

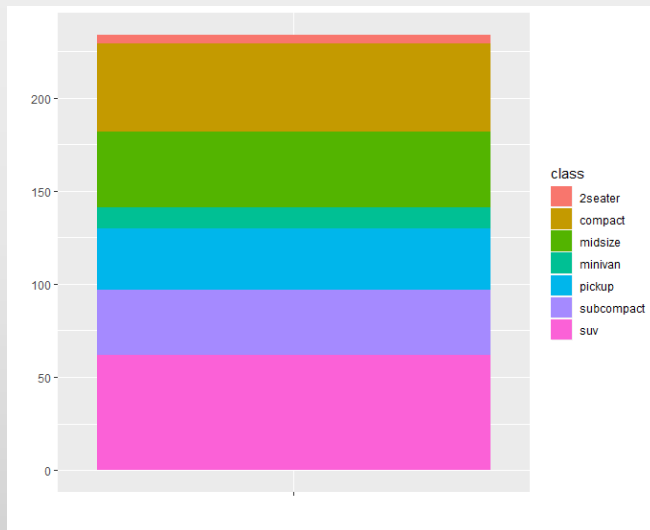
- 막대의 폭은 동일 / 높이는 다름
- x 변수: 막대의 폭(같은 값)
- y 변수: 막대의 높이(다른 값)

- 예제: mpg의 변수 class의 파이 그래프 작성

- 1) 막대 그래프 작성

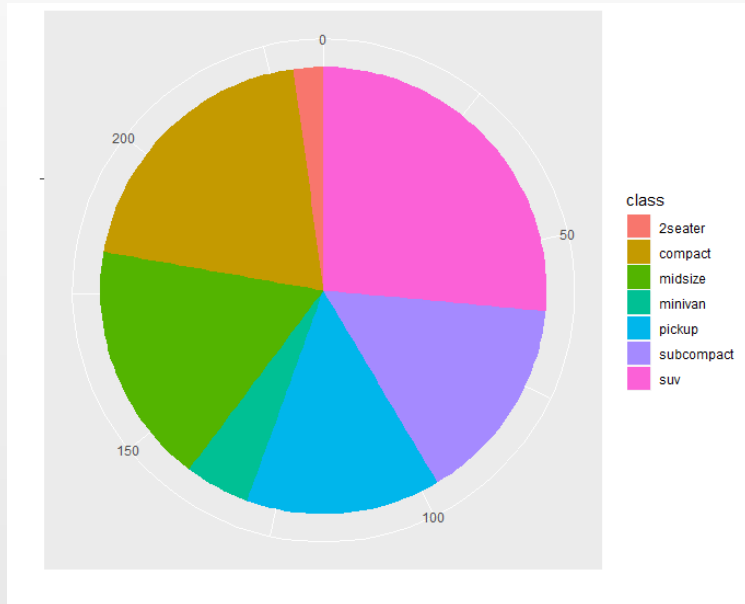
- 각 막대마다 다른 색 사용
- 쌓아 올린 형태로 작성
- 막대의 폭을 X축 전체 구간으로 확장

```
> b2 <- ggplot(data=mpg, mapping=aes(x="", fill=class)) +  
  geom_bar(width=1) +  
  labs(x="", y="")  
> b2
```

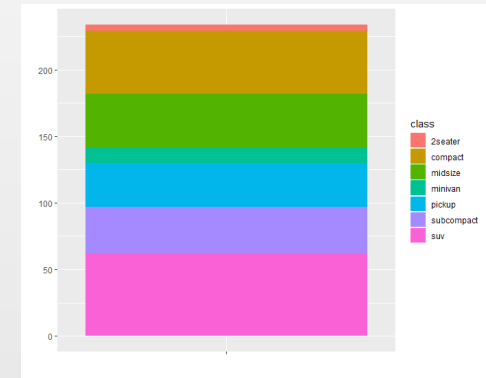


2) theta="y"로 함수 coord\_polar( ) 실행: 파이 그래프 작성

```
> b2 + coord_polar(theta="y")
```

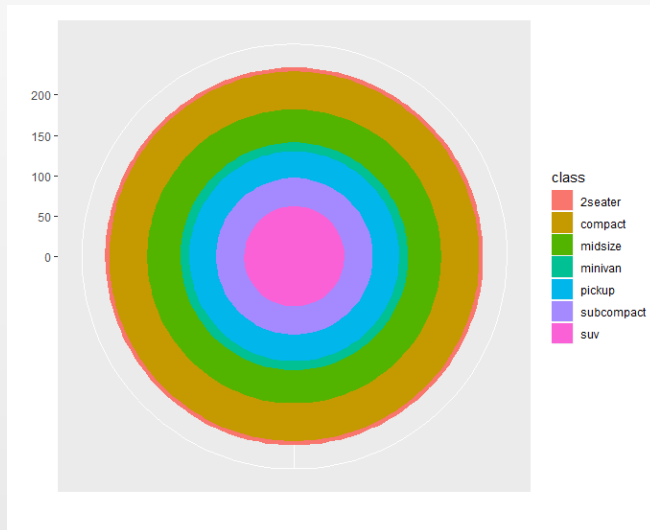


- 각 조각의 각도: 막대의 높이에 비례
- 각 조각의 반지름: 막대의 폭에 비례



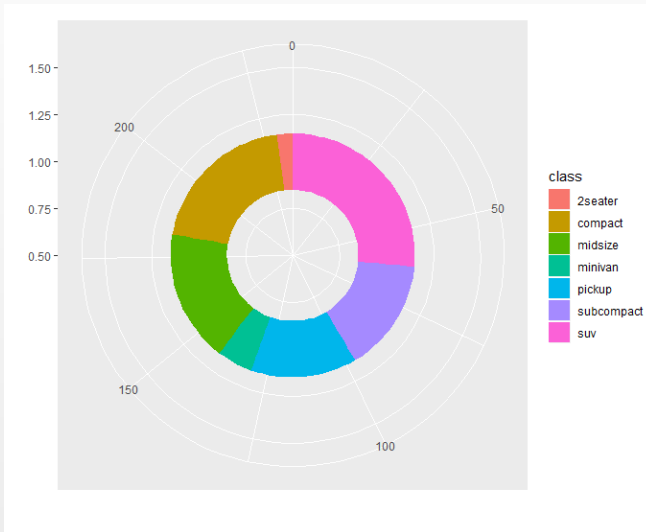
### 3) theta="x"로 함수 coord\_polar( ) 실행: Bullseye 그래프 작성

```
> b2 + coord_polar()
```



- 각 조각의 각도: 전체 구간에 해당하는 360° 적용
- 각 조각의 반지름: 막대의 높이에 비례

- 예: 다음의 그래프 작성



- 파이 그래프의 가운데 부분이 없는 그래프
- Donut 그래프 혹은 ring 그래프

```
> b3 <- ggplot(data=mpg, mapping=aes(x=1, fill=class)) +  
  geom_bar(width=0.3) + labs(x="", y="") + xlim(0.5, 1.5)  
> b3 + coord_polar(theta="y")
```