

5장. ggplot2에 의한 자료 시각화

1

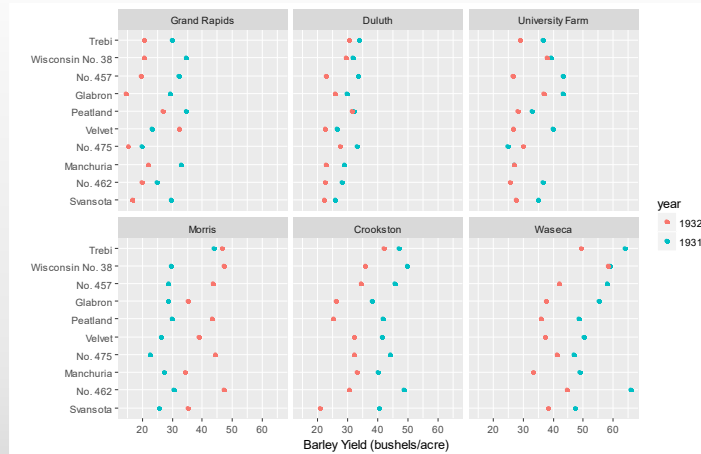
그래픽 기법의 활용 예제: 보리자료

- 자료분석과정에서 그래프의 이용이 필수적임을 보여주는 예제
- 1930년대 초 미네소타 주 농경학자들이 보리종류에 따른 수확량의 차이를 비교하기 위해 2년간 경작실험을 실시
- 요인: 6군데 경작지, 10종류의 보리, 2년간의 경작 년도
- 반응변수: 수확량
- 실험계획법에 대한 Fisher의 아이디어가 적용된 최초의 실험자료
- 저명한 학자들에 의해서 여러 번 분석된 자료
- Cleveland가 자료에 있는 문제 발견

2

2

- 보리자료의 문제점



- 모든경작지 에서 1931 > 1932
- Morris 지역만 1932 > 1931 (뒤바뀜 ?)

3

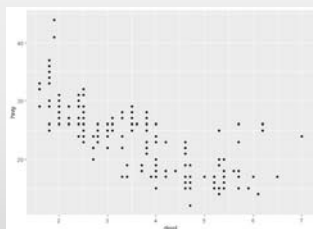
3

5.1 ggplot2 시작하기

- 패키지 ggplot2에 있는 데이터 프레임 mpg의 변수 displ과 hwy의 산점도 작성

```
> library(tidyverse)
> ggplot(data=mpg) +
  geom_point(mapping=aes(x=displ, y=hwy))
```

- "geom": geometry
- "aes": aesthetic



- 함수 ggplot(): 데이터 프레임 data 지정.
그래프가 작성될 비어있는 영역 확보
- 함수 geom_point(): 실질적인 그래프를 작성하는 geom 함수
- geom 함수 입력요소 mapping: geom 함수 내에서 항상 함수 aes()와 함께 시각적 요소를 데이터와 연결
- 함수 ggplot 안에서 mapping 도 가능 (추후설명)

4

4

- ggplot2에서 그래프 작성의 최소 요소

- 그래프 작성을 위한 법칙이 있음: 그래프의 문법
- 모든 그래프 작성에 일정하게 적용
- 익숙해지면 복잡한 형태의 그래프도 어렵지 않게 작성 가능

그래프 작성을 위한 3가지 최소 요소: <Data>, <Geom_function>, <Mappings>

```
ggplot(data=<Data>) +
  <Geom_function>(mapping=aes(<Mappings>)) +
  <Geom_function> + <Geom_function> + ...
```

<Data>: 그래프 작성에 사용될 데이터 지정. 반드시 데이터프레임 지정.

<Geom_function>: 레이어(layer) 작성을 위한 geom 함수.

- 여러 개의 geom 함수를 덧셈 기호로 연결하면 여러 레이어가 겹쳐진 그래프 작성
- 데이터를 지정할 수 있으나 지정하지 않으면 ggplot에서 지정된 데이터프레임 사용

<Mappings>: 시각적 요소(점의 크기, 모양, 색깔, ...)와 데이터 연결

- ggplot() 으로 시작
 - geom 함수내 입력요소 mapping, 항상 aes() 사용하여 지정
 - + 를 사용하여 layers, scales, coords and facets 추가
- Layers : geom_, stat_, position_, annotation_, ...

5

5

5.2 시각적 요소와 연결: Mapping

- 시각적 요소

- 그래프를 시각적으로 인식할 때 필요한 요소
- 산점도의 경우, 점의 위치, 크기, 모양 및 색깔 등이 시각적 요소

- 시각적 요소의 mapping과 setting

- mapping: 데이터의 값과 연결되어 결정. 함수 aes() 안에서 연결
- setting: 데이터와 관계없이 일정한 값 지정. 함수 aes() 밖에서 지정

- 시각적 요소의 mapping

- 기존의 그래프에 다른 변수의 정보 추가 가능

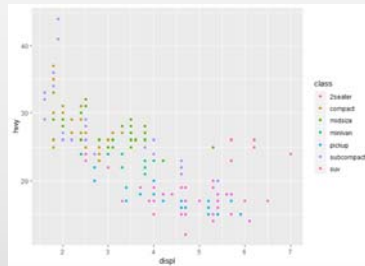
6

6

- 예제: 데이터 프레임 mpg의 변수 displ과 hwy의 산점도에 시각적 요소와의 mapping으로 다른 변수 정보 추가

- 변수 class를 시각적 요소 color와 mapping

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



- 변수 class: 문자형 벡터
- 변수 class의 값에 따라 다른 색 사용
- 사용된 색깔에 대한 범례는 자동으로 추가

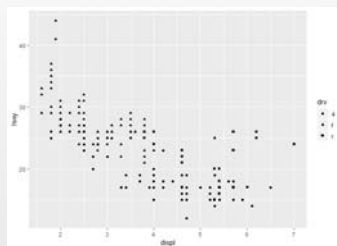
- 색의 종류(범주의 개수) 과다
- 좋은 그래프는 아님

7

7

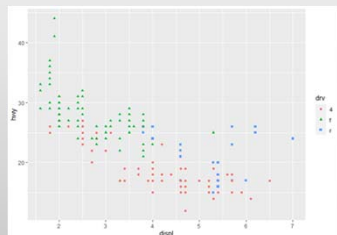
- 변수 drv를 시각적 요소 shape와 mapping

```
> ggplot(data=mpg) +  
  geom_point(mapping=aes(x=displ, y=hwy, shape=drv))
```



- shape에는 이산형 변수 mapping
- 변수 drv: 문자형 벡터
- 변수 drv의 값에 따라 다른 모양의 점 사용
- 범례 자동 추가
- 구분이 어려운 다른 모양의 점
- 좋은 그래프는 아님

- shape와 color를 drv와 mapping

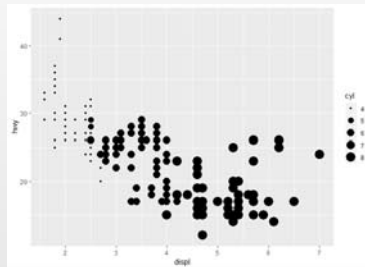


8

8

- 변수 cyl을 시각적 요소 size와 mapping

```
> ggplot(data=mpg) +  
  geom_point(mapping=aes(x=displ, y=hwy, size=cyl))
```



- size에 mapping되는 변수는 연속형이 좋음
- 변수 cyl: 정수형 변수
- cyl의 값에 따라 점의 크기 조절
- 범례 자동 추가

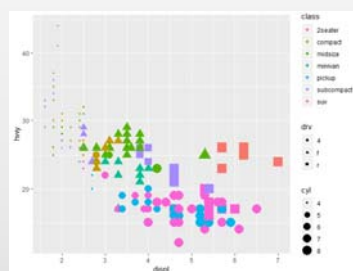
- 구분이 어려운 점 크기
- 범주의 개수 과다 → 좋은 그래프는 아님

9

9

- 여러 시각적 요소를 동시에 mapping
- 변수 class는 color와, drv는 shape와, cyl은 size와 mapping

```
> ggplot(data=mpg) +  
  geom_point(mapping=aes(x=displ, y=hwy,  
    color=class, shape=drv, size=cyl))
```



- 너무 많은 정보
- 그래프의 의미가 모호

10

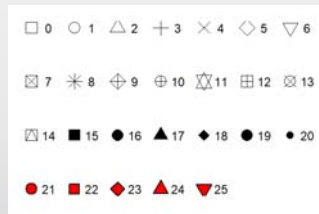
10

- 시각적 요소의 setting

- 함수 aes() 밖에서 사용자가 원하는 값으로 지정
- geom 함수의 입력 요소가 됨

- 시각적 요소 color, size, shape에 값 지정 법칙

- 1) color: 색깔을 나타내는 문자열 지정
- 2) size: 점 크기를 mm 단위로 지정
- 3) shape: 점의 형태를 나타내는 0~26 사이의 숫자



도형에 색깔 지정 방법

- 1) 0~14의 외곽선 및 15~20의 도형 색: color 사용
- 2) 21~25의 외곽선: color 사용
- 3) 21~25의 내부 색: fill 사용

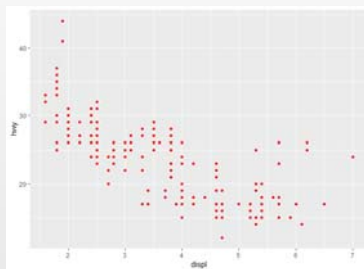
<https://www.r-graph-gallery.com/colors>

11

11

- 시각적 요소 color의 setting: 모든 점을 빨간 색으로

```
> ggplot(data=mpg) +  
  geom_point(mapping=aes(x=displ, y=hwy), color="red")
```



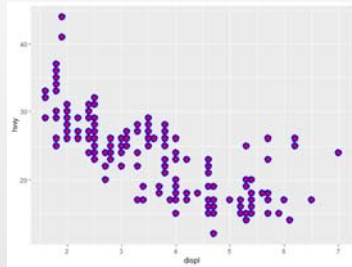
- color를 함수 aes() 밖에서 지정
- 함수 geom_point()의 입력 요소

12

12

- 여러 시각적 요소를 동시에 setting

```
> ggplot(data=mpg) +  
  geom_point(mapping=aes(x=displ, y=hwy, color="blue", size=3,  
    shape=21, fill="red", stroke=2))
```



- 점의 모양: shape=21
- 점의 내부 색: 빨간색
- 점의 외곽선 색: 파란색
- 점의 크기 확대: size=3
- 점의 외곽선 두께 조절: stroke=2

13

13

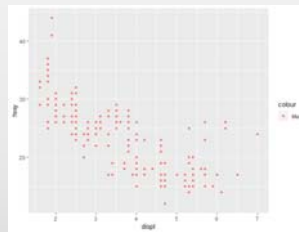
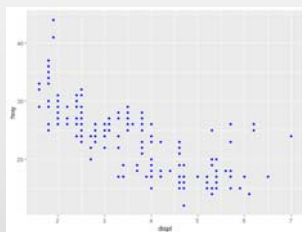
- 함수 aes() 안에서 시각적 요소에 특정 값을 setting한 경우의 결과

Setting

```
> ggplot(data=mpg)+  
  geom_point(mapping=aes(x=displ, y=hwy, color="blue"))
```

Mapping

```
> ggplot(data=mpg) +  
  geom_point(mapping=aes(x=displ, y=hwy, color="blue"))
```



- mapping은 변수와의 연결을 의미
- "blue"라는 값을 갖는 변수 스스로 생성, 매핑
- 하나의 값만 있는 변수에 color 매핑, 디폴트 색

14

14

5.3 그룹별 그래프 작성: Facet

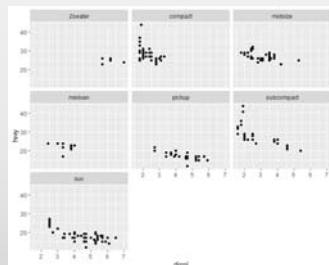
- 범주형 변수가 다른 변수에 미치는 영향력을 그래프로 확인하는 방법
 - 1) 시각적 요소에 범주형 변수를 지정하여 구별(mapping)
 - 2) 범주형 변수로 그룹 구성하고, 각 그룹별 그래프 작성: faceting
- facet을 적용하기 위한 함수
 - ① `facet_wrap()`: 한 변수에 의한 facet
 - ② `facet_grid()`: 한 변수 또는 두 변수에 의한 facet

15

15

- 함수 `facet_wrap()`에 의한 faceting
 - 데이터를 구분하는 변수가 하나인 경우: `facet_wrap(~ x)` (~ :R formula 표시)
- 데이터 프레임 mpg의 변수 displ과 hwy의 산점도를 class의 범주별로 작성

```
> ggplot(data=mpg) +  
  geom_point(mapping=aes(x=displ, y=hwy)) +  
  facet_wrap(~ class)
```



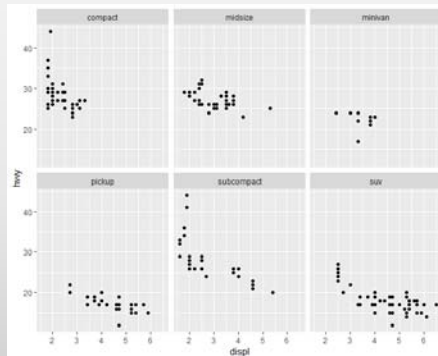
- 패널 '2seater'에는 적은 수의 데이터 존재
- class가 '2seater'인 케이스 제거 후 다시 작성 (next slide, exercise)

16

16

- 데이터 프레임 mpg의 변수 displ과 hwy의 산점도를 class의 범주별로 작성 (2seater 케이스 제외)

```
> mpg %>%
  filter(class != "2seater") %>%
  ggplot() +
  geom_point(mapping=aes(x=displ, y=hwy)) +
  facet_wrap(~ class)
```



17

17

- 패널 배치 조절

- 2x3 패널 패치를 3x2 배치로 수정: ncol=2
- 패널에 그래프 배치 순서를 열 단위로 수정: dir="v"

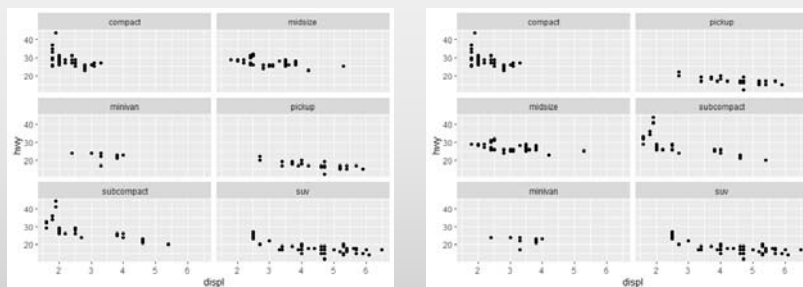
direction="vertical"

```
> pp <- mpg %>%
  filter(class != "2seater") %>%
  ggplot() +
  geom_point(mapping=aes(x=displ, y=hwy))
```

두 그래프에 공통적으로 적용되는 내용을 개체에 할당

```
> pp + facet_wrap(~ class, ncol=2)
> pp + facet_wrap(~ class, ncol=2, dir="v")
```

객체(ggplot) + 레이아웃



18

18

- 함수 `facet_grid()`에 의한 faceting

- 한 변수에 의한 faceting:
하나의 행으로 패널 배치: `facet_grid(. ~ x)`
하나의 열로 패널 배치: `facet_grid(x ~ .)`
- 두 변수에 의한 faceting: `facet_grid(y ~ x)`
행 범주: 변수 y의 범주
열 범주: 변수 x의 범주

(y ~ x): 행 ~ 열
(. ~ x): 1행 ~ 열
(x ~ .): 행 ~ 1열

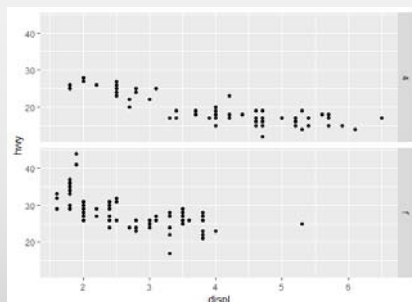
19

19

- 데이터 프레임 `mpg`에서 변수 `drv`와 `cyl`의 범주별로 `displ`과 `hwy`의 산점도 작성. 단, `drv`가 "r"인 자료와 `cyl`이 5인 자료는 제외

```
> my_plot <- mpg %>%  
  filter(cyl != 5, drv != "r") %>%  
  ggplot() +  
  geom_point(mapping=aes(x=displ, y=hwy))
```

```
> my_plot + facet_grid(drv ~ .)
```



drv='4'

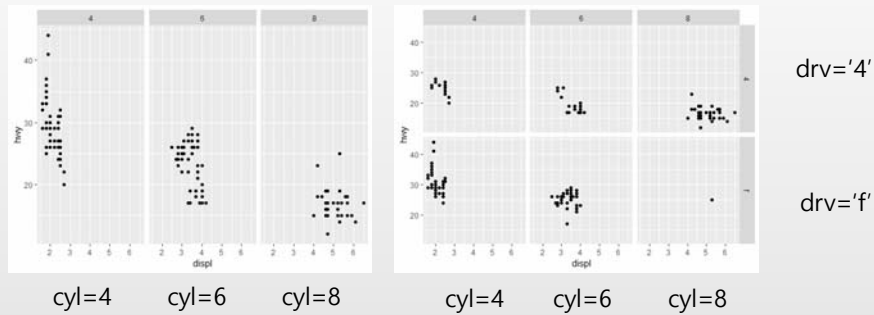
drv='f'

20

20

```
> my_plot + facet_grid(. ~ cyl)
```

```
> my_plot + facet_grid(drv ~ cyl)
```



21

21

5.4 기하 객체: Geometric object

- Base graphics에서 그래프 작성 방식: pen on paper
 - 높은-수준의 그래프 함수: 좌표축과 주요 그래프 작성
 - 낮은-수준의 그래프 함수: 점, 선, 문자 등을 추가하여 원하는 그래프 작성
- ggplot2에서 그래프 작성 방식
 - 필요한 유형의 그래프(점 또는 선 그래프 등) 작성
 - ▶ geom 함수 실행 → 해당 유형의 그래프가 작성된 layer 생성
 - 점 그래프: `geom_point()`
 - 선 그래프: `geom_line()`
 - 작성된 그래프를 겹쳐 놓음으로써 원하는 그래프 작성
 - ▶ geom 함수로 작성된 layer를 차례로 겹쳐 놓음

22

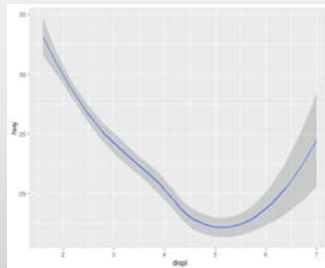
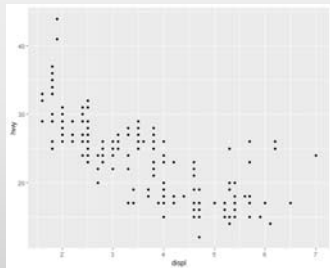
22

- 동일 자료에 다른 geom 적용

- mpg의 변수 displ과 hwy를 대상으로 point geom과 smooth geom 적용
 - point geom: 점 그래프 작성
 - smooth geom: 비모수 회귀곡선 작성

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

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



23

23

- geom 함수

- 현재 대략 30개 이상의 geom 함수가 있음
- 한 변수에 대한 함수: geom_bar(), geom_histogram(), geom_density(), geom_dotplot() 등등
- 두 변수에 대한 함수: geom_point(), geom_smooth(), geom_text(), geom_line(), geom_boxplot() 등등
- 세 변수에 대한 함수: geom_contour(), geom_tile() 등등
- geom 함수의 리스트: R studio의 메뉴에서 'Help > Cheatsheets > Data Visualization with ggplot2' 에서 확인 가능

24

24

- 글로벌 매핑과 로컬 매핑

- 글로벌 매핑: 함수 `ggplot()`에서의 매핑. 해당 그래프 작성에 참여한 모든 `geom` 함수에 적용
- 로컬 매핑: `geom` 함수에서의 매핑. 해당 `geom` 함수로 작성되는 layer에만 적용. 해당 layer에서는 글로벌 매핑보다 우선해서 적용됨.

```
ggplot(data, mapping=aes( ) ) +  
  geom_*(mapping=aes( ) ) +  
  geom_*(mapping=aes( ) )
```

글로벌 매핑

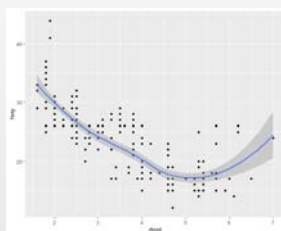
로컬 매핑

25

25

- 예: mpg의 변수 `displ`과 `hwy`의 산점도에 비모수 회귀곡선 추가

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



두 `geom` 함수에 동일한 내용의 매핑이 중복 입력되는 상황

- 글로벌 매핑으로 중복 입력 문제 해결

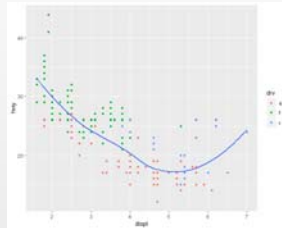
```
> ggplot(data=mpg, mapping=aes(x=displ, y=hwy)) +  
  geom_point() +  
  geom_smooth()
```

26

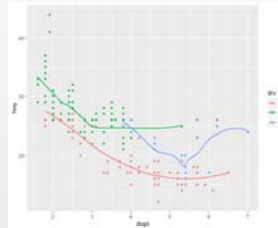
26

- 예: mpg의 변수 displ과 hwy의 비모수 회귀곡선 작성. 그 위에 산점도 추가하되 drv의 값에 따라 점의 색을 구분.

비모수 회귀곡선: 전체 자료 대상



각 그룹별로 추정된 비모수 회귀



```
> ggplot(data=mpg, mapping=aes(x=displ, y=hwy)) +  
  geom_point(mapping=aes(color=drv)) +  
  geom_smooth(se=FALSE)
```

x, y: 글로벌 매핑
color: 로컬 매핑
(점만 해당)

```
> ggplot(data=mpg, mapping=aes(x=displ, y=hwy, color=drv)) +  
  geom_point() +  
  geom_smooth(se=FALSE) # se: confidence interval
```

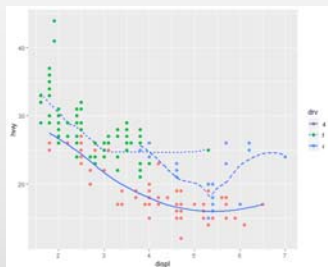
x, y, color: 글로벌 매핑
(생성되는 모든 것 해당)

27

27

- 예: mpg의 변수 displ과 hwy의 비모수 회귀곡선 작성하되 drv에 의해 구분되는 그룹별 각각 추정하여 선의 종류를 다르게 표시. 그 위에 산점도 추가하되 drv의 값에 따라 점의 색을 구분, 점의 크기 확대.

```
> ggplot(data=mpg, mapping=aes(x=displ, y=hwy)) +  
  geom_point(mapping=aes(color=drv), size=2) +  
  geom_smooth(mapping=aes(linetype=drv), se=FALSE)
```



- `linetype`: 선의 종류를 나타내는 시각적 요소

- `linetype` 을 지정하면

지정된 변수의 그룹별로 함수수행
+
그룹별 결과의 선 종류 다르게 표시

- 지정된 변수의 그룹별 함수 수행만 하려면 ?

`group` 사용

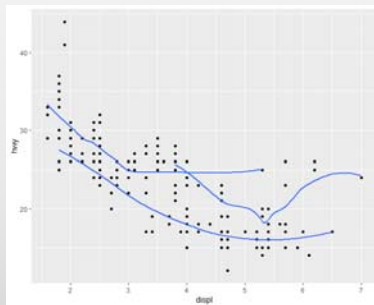
28

28

- 예: 다음의 그래프 작성

- 변수 drv의 그룹별로 따로 비모수 회귀곡선 작성하되, 선의 색과 종류는 같은 것을 사용

```
> ggplot(data=mpg, mapping=aes(x=displ, y=hwy)) +  
  geom_point() +  
  geom_smooth(mapping=aes(group=drv), se=FALSE)
```



group: 그룹을 구성하는 시각적 요소

29

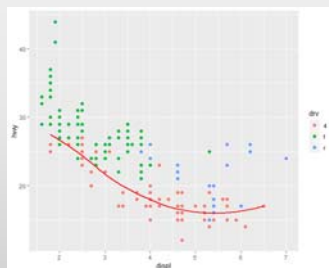
29

- 각 geom 함수에서 다른 데이터 사용

- 각 geom 함수로 작성되는 layer마다 다른 데이터로 그래프 작성 가능

- 예: mpg의 변수 displ과 hwy의 산점도. drv에 따라 점의 색 구분. 비모수 회귀곡선 추가하되 drv가 4인 데이터만을 대상으로 추정.

```
> ggplot(data=mpg, mapping=aes(x=displ, y=hwy)) +  
  geom_point(mapping=aes(color=drv), size=2) +  
  geom_smooth(data=filter(mpg, drv=="4"),  
    se=FALSE, color="red")
```



30

30

5.5 통계적 변환: Statistical transformation (skip)

- ggplot() 그래프 작성식 사용되는 자료
 - 1) 입력된 자료 그대로 : 산점도
 - 2) 입력된 자료를 대상으로 통계적 변환 과정을 거쳐 생성된 자료: 비모수 회귀 곡선 그래프
- 각 그래프에서 필요한 통계적 변환을 지정해야 함.
- 통계적 변환(stat)
 - 입력된 데이터 프레임 자료의 변환을 의미
 - 각 그래프 유형별 대응되는 stat 존재
 - ▶ 산점도: stat="identity"
 - ▶ 비모수 회귀곡선: stat="smooth"
 - ▶ 막대 그래프: stat="count"
 - 각 geom 함수마다 대응되는 디폴트 stat 존재
 - ▶ geom_point() → geom_point(stat="identity")
 - ▶ geom_smooth() → geom_smooth(stat="smooth")
 - ▶ geom_bar() → geom_bar(stat="count")

31

31

5.6 위치 조정: Position adjustments

- 그래프 요소들의 위치 조정
 - 연속형 자료: 산점도의 점이 겹쳐지는 경우
 - 범주형 자료: 이변량 막대 그래프 작성
- 산점도의 점이 겹치는 문제
 - 산점도 작성의 가장 큰 문제
 - 해결 방안
 - ▶ 반올림 처리 등으로 같은 값이 많은 자료의 경우: 자료에 약간의 난수 추가로 점의 위치 조정(jittering)
- 이변량 막대 그래프
 - 쌓아 올린 막대 그래프 / 옆으로 붙여 놓은 막대 그래프

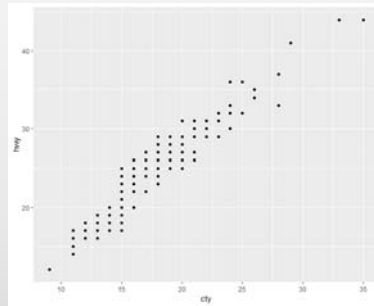
32

32

- 산점도에서 점이 겹쳐지는 문제 해결

- 예: mpg에서 변수 cty와 hwy의 산점도 작성

```
> ggplot(data=mpg, mapping=aes(x=cty, y=hwy)) +  
  geom_point()
```



- 산점도에 나타난 점의 개수가 전체 데이터 개수인 234개에 훨씬 못 미쳐 보임
- 두 변수의 값이 반올림 처리되어 같은 값이 많아짐

jittering

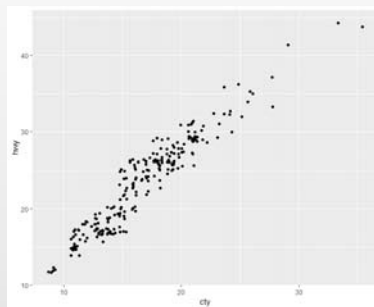
- 자료에 약간의 난수 추가
- $(x, y) \rightarrow (x + \varepsilon, y + \varepsilon)$
 $\varepsilon \sim Unif(-\alpha, \alpha)$

33

33

- jittering 실시

```
> ggplot(data=mpg, mapping=aes(x=cty, y=hwy)) +  
  geom_point(position="jitter")
```



- 작성되는 그래프마다 미세한 차이 발생
- 추가되는 난수의 크기를 조절하고자 하는 경우에는 함수 geom_jitter() 사용

34

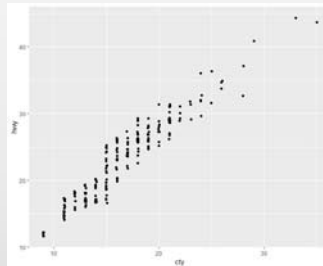
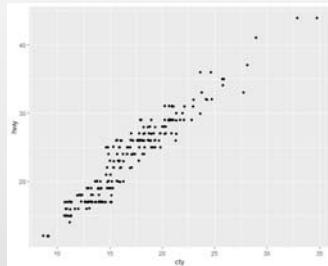
34

- 함수 geom_jitter()

```
> ggplot(data=mpg, mapping=aes(x=cty, y=hwy))+  
  geom_jitter(width=0.4, height=0.05)
```

width : 좌우로 흔들는 정도
height : 상하로 흔들는 정도

```
> ggplot(data=mpg, mapping=aes(x=cty, y=hwy))+  
  geom_jitter(width=0.05, height=0.4)
```



35

35

● 이변량 막대 그래프 작성

- 막대 그래프 작성: geom_bar()
- 이변량 막대 그래프: 함수 geom_bar()에 시각적 요소 x와 fill, position 사용

- 예제: mpg에서 trans의 범주를 auto와 manual로 통합한 변수 am 생성
변수 cyl이 5인 케이스 제거 후 am과 cyl의 이변량 막대 그래프 작성

- 자료 준비

```
> mpg_1 <- mpg %>%  
  mutate(am=substr(trans, 1, nchar(trans)-4)) %>%  
  filter(cyl != 5)
```

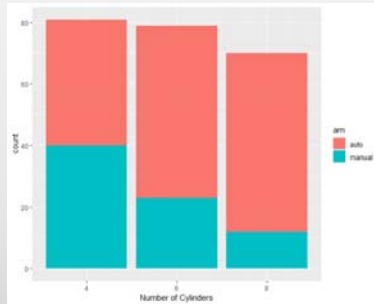
36

36

- 쌓아 올린 막대 그래프와 옆으로 붙여 놓은 막대 그래프 작성
- geom_bar 에서 시각적 요소 x 와 fill 에 변수 지정
- position에 의해 이변량막대 형태 결정

```
> p_1 <- ggplot(data=mpg_1,
  mapping=aes(x=as.factor(cyl), fill=am)) +
  xlab("Number of Cylinders")
```

```
> p_1 + geom_bar()
```



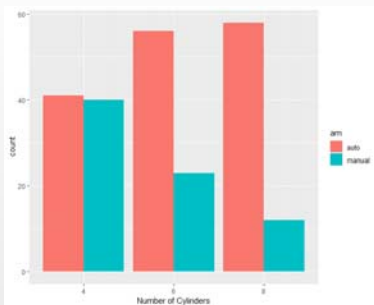
position="stack", "dodge", "dodge2", "fill"
빈도 비율
디폴트 : stack, 수직누적

ggplot(mpg_1, aes(x=cyl, fill=am))을
실행하면 어떤 문제가 있는가?
- cyl 실수 취급. X 축 막대 좌표 표시 변화.

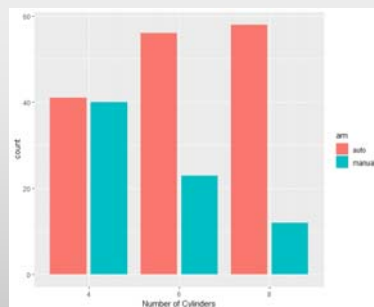
37

37

- 옆으로 붙여 놓은 막대 그래프



```
> p_1 + geom_bar(position="dodge")
```



```
> p_1 + geom_bar(position="dodge2")
```

- 차이 ; 막대간 간격

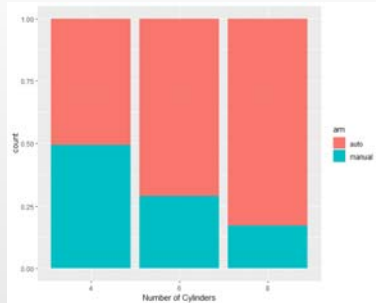
38

38

- 조건부 확률로 쌓아 올린 막대 그래프

```
> p_1 + geom_bar(position="fill")
```

position="fill":
수직누적, 확률표시



cyl을 조건으로 하는 cyl과 am의 조건부 확률

	am	
cyl	auto	manual
4	0.5061728	0.4938272
6	0.7088608	0.2911392
8	0.8285714	0.1714286

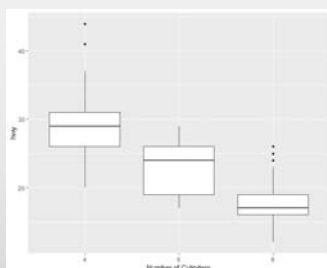
39

39

● 나란히 서 있는 상자그림

- geom_boxplot()
- 필요한 시각적 요소: x=그룹을 구성하는 변수(요인)
y=연속형 변수

```
> ggplot(data=mpg_1, mapping=aes(x=as.factor(cyl), y=hwy)) +  
  geom_boxplot() +  
  xlab("Number of Cylinders")
```



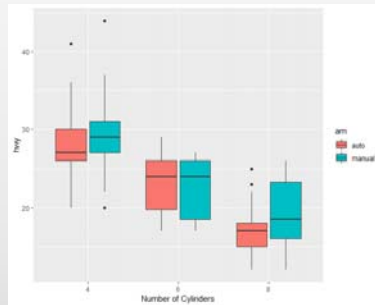
- 시각적 요소 x에 요인이 아닌 숫자형 변수인 cyl을 매핑하면 어떤 그래프가 작성되는가?
(단일 상자, 전체 데이터 사용)

40

40

- 그룹을 구성하는 변수가 두 개인 경우의 상자그림

```
> ggplot(data=mpg_1, mapping=aes(x=as.factor(cyl), y=hwy)) +  
  geom_boxplot(mapping=aes(fill=am)) +  
  xlab("Number of Cylinders")
```



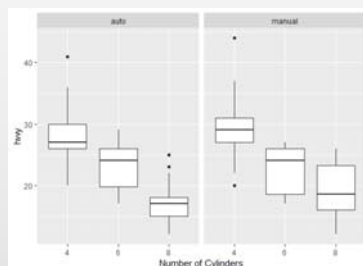
- 변수 am에 따라 다른 색이 채워져 있고 두 상자그림이 옆에 붙어 있다
- 필요한 시각적 요소: x, y, fill, position
- position="dodge"가 디폴트로 적용됨

41

41

- 그룹을 구성하는 변수가 두 개인 경우의 상자그림

```
> ggplot(data=mpg_1, mapping=aes(x=as.factor(cyl), y=hwy)) +  
  geom_boxplot() +  
  xlab("Number of Cylinders") +  
  facet_wrap(~am)
```



42

42

5.7 좌표계: Coordinate system

- 좌표계: 시각적 요소 x와 y를 근거로 그래프의 각 요소의 2차원 위치를 결정하는 체계
- 좌표계의 종류
 - `coord_cartesian()`: 디폴트
 - `coord_flip()`
 - `coord_polar()`

43

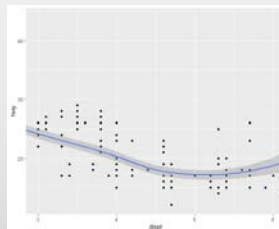
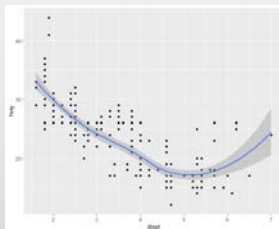
43

- `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))
```



44

44

- scale에 의한 XY축 범위 조정 (추후 참고)

- 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")`

45

45

- 예: XY축 조정 비교

1) 함수 `xlim()`에 의한 조정

범위를 벗어난 자료: 삭제

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

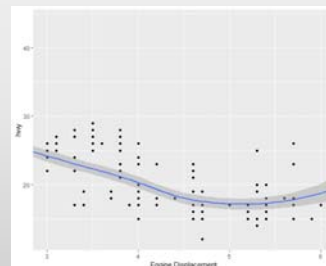
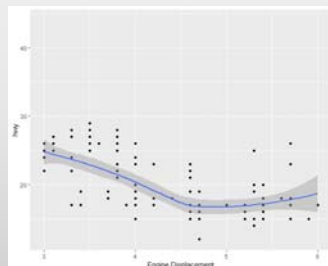
Warning messages:

- 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

범위 + 여분

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



46

46

- 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

47

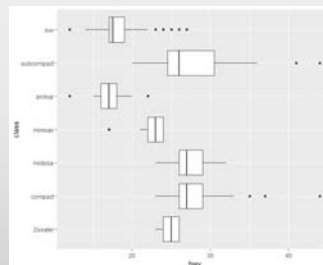
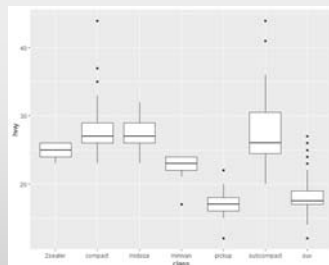
47

- 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()
```



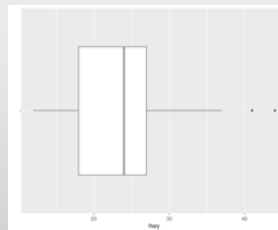
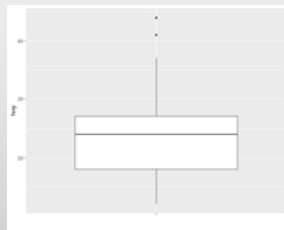
48

48

- 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("")
```

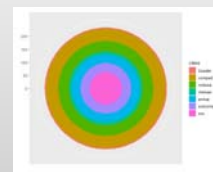
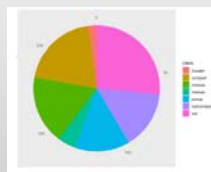
```
> ggplot(data=mpg, mapping=aes(x="", y=hwy)) +  
  geom_boxplot() +  
  xlab("") +  
  coord_flip()
```



49

49

- 함수 `coord_polar()`의 활용: 파이 그래프 작성 (skip)
 - 극좌표(polar coordinate): 2차원 공간의 어느 한 점의 위치를 원점에서 거리와 각도로 표현
 - 함수 `coord_polar()`: 데카르트 좌표를 극좌표로 변환
 - 변수 `theta`: 시각적 요소 x와 y 중 각도로 전환할 요소 지정(디폴트는 `theta="x"`)
- 함수 `coord_polar()`를 활용하여 막대 그래프에서 변형된 그래프
 - Coxcomb 또는 Wind rose 그래프
 - 파이 그래프
 - Bullseye 그래프



50

50