일변량 자료 탐색

박영식(youngsik.park@bsl-lausanne.ch)

그래프에 의한 일변량 자료 탐색

◎ 범주형 자료를 위한 그래프
- 막대 그래프
- 파이 그래프
- Cleveland의 점 그래프

◎ 연속형 자료를 위한 그래프
- 줄기-잎 그림
- 상자그림
- Violin plot
- 히스토그램
- 확률밀도 함수 그래프
- 도수분포다각형
- 점 그래프(dot Plot)
- 경험적 누적분포함수 그래프

범주형 자료를 위한 그래프

◎ 막대 그래프

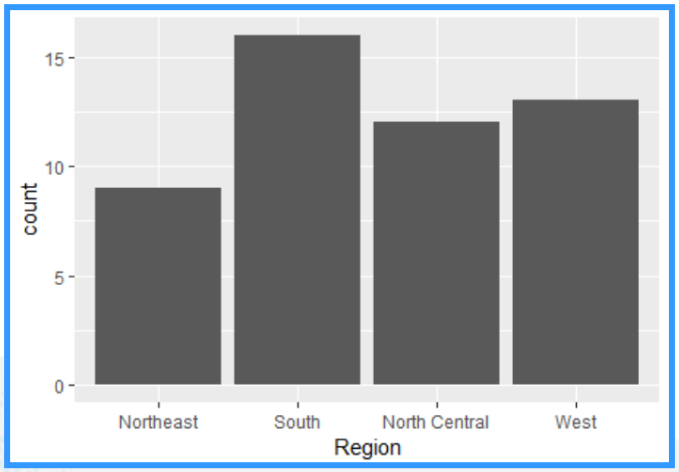
- 예: state.region

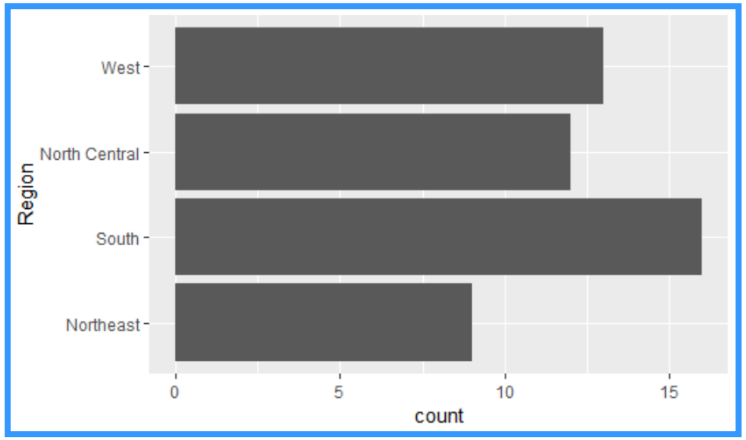
```
> str(state.region)
 Factor w/ 4 levels "Northeast", "South", . . :  2 4 4 2 4 4 1 2 2 2 . . .

> state.region[1:5]
[1] South West  West  South West
Levels: Northeast South North Central West
```

1) Input data가 요인인 경우

```
> ggplot(data.frame(state.region)) +
      geom_bar(aes(x=state.region)) +
      labs(x="Region")
```

```
> ggplot(data.frame(state.region)) +
      geom_bar(aes(x=state.region)) +
      labs(x="Region") +
      coord_flip( )
```





2) Input data가 도수분포표인 경우

```
> counts <-table(state.region)
> counts
state.region
    Northeast         South North Central          West
            9            16            12            13
```

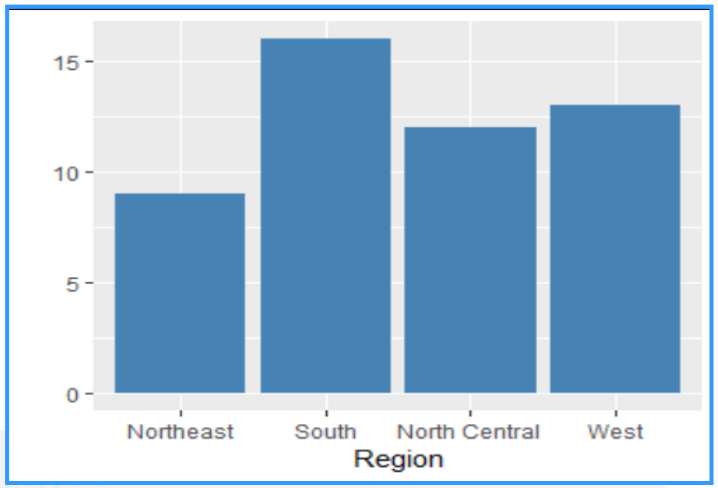
```
> df_1<-as.data.frame(counts)
> df_1
       state.region  Freq
1         Northeast     9
2             South    16
3   North Central    12
4              West    13
```

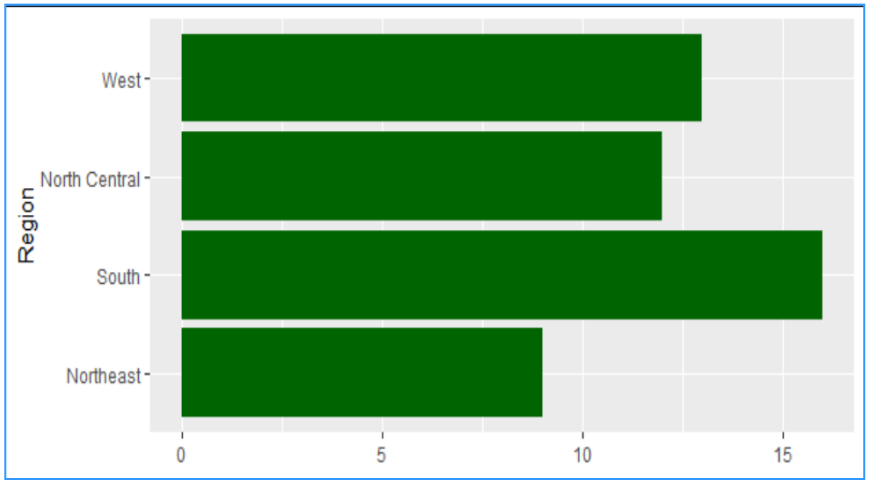
```
> ggplot(df_1, aes(x=state.region, y=Freq)) +
      geom_col(fill="steelblue") +
      labs(x="Region", y="")
```

함수 geom_col( ):

geom_bar(stat="identity")

```
> ggplot(df_1, aes(x=state.region, y=Freq)) +
      geom_col(fill="dark green") +
      labs(x="Region", y="") +
      coord_flip()
```





연속형 자료를 위한 그래프

◎ 상자그림(Boxplot)

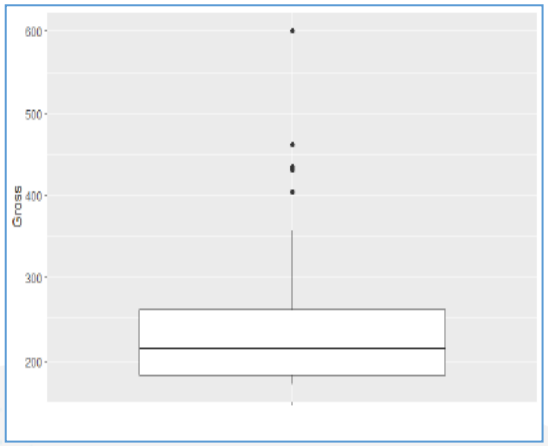
◎ 히스토그램

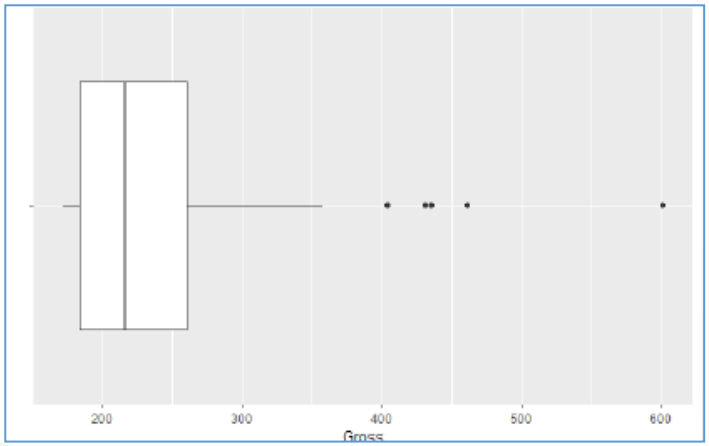
◎ 확률밀도함수 그래프

상자그림

◎ 예: UsingR::alltime.movies의 변수 Gross의 상자그림 작성

```
> library(UsingR)

> bp <- ggplot(alltime.movies, aes(x="", y=Gross)) +
    geom_boxplot() +
    labs(x="")
> bp
> bp + coord_flip()
```



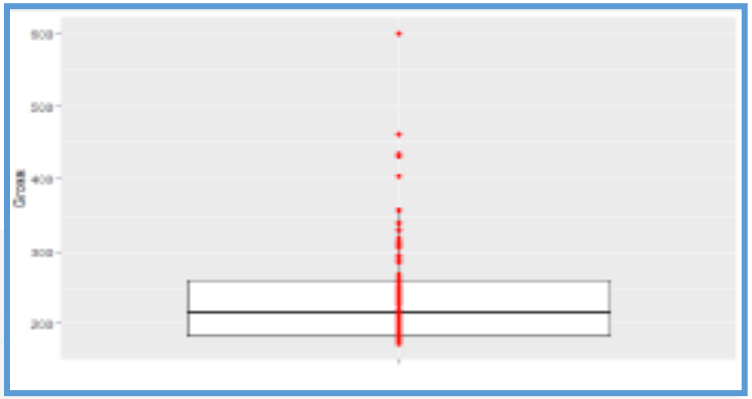


상자그림

◎ 상자 그림에 자료의 위치를 점으로 표시한다.

- 함수 geom_point( ) 추가
- 상자그림에서 이상 값을 원으로 표시하는 것 중지: 자료의 점과 겹쳐져서 나타나므로 (함수 outlier.shape=NA 추가)

```
> bp1 <- ggplot(alltime.movies, aes(x="", y=Gross)) +
    geom_boxplot(outlier.shape=NA) +
    labs(x="")
> bp1 + geom_point(color="red")
```

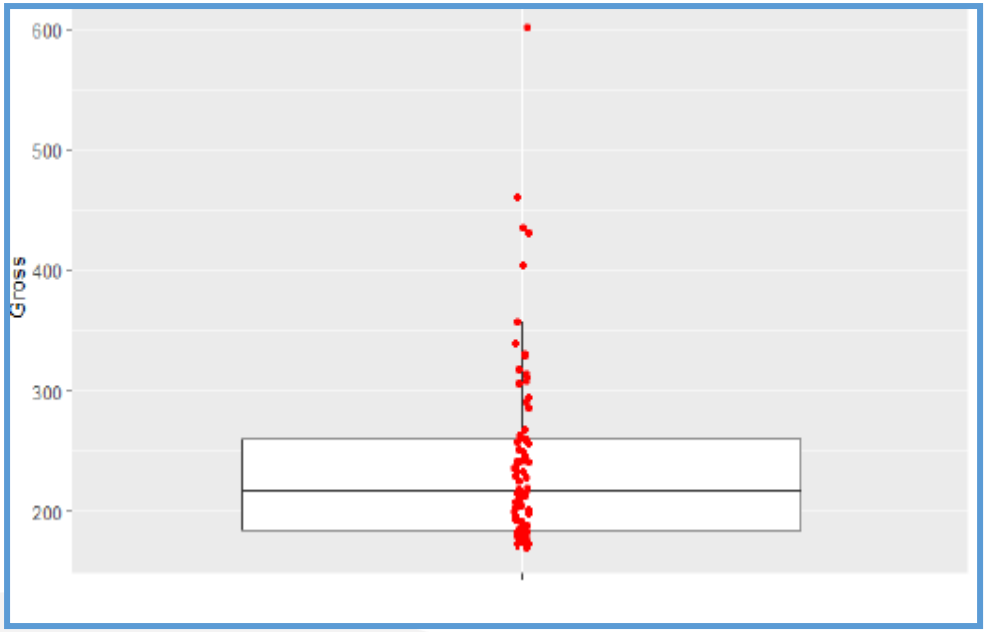


- 자료의 점이 겹쳐짐
- geom_jitter( )의 사용이 필요함

상자그림

- 함수 geom_jitter( )로 상자그림에 자료 위치 표시

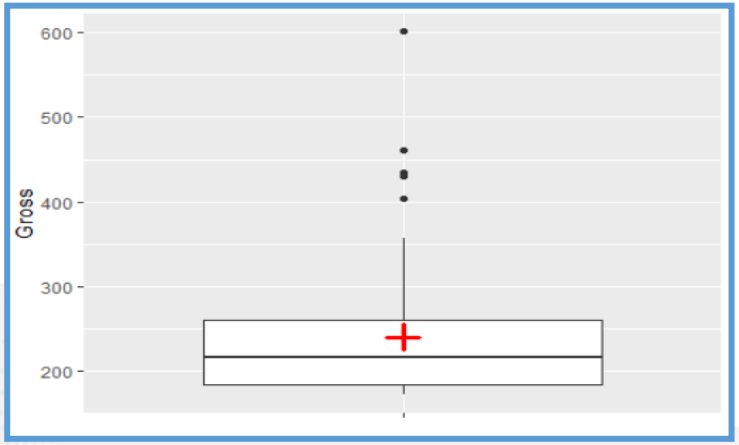
```
> bp1 + geom_jitter(color="red", width=0.01)
```



상자그림에 평균값 위치 표시

- 함수 stat_summary( ): 자료의 요약 통계량을 그래프에 표시
  하나의 x값에 대하여 주어진 y값의 통계량 값 계산
  원하는 요약 통계량: 변수 fun.y에 지정
  원하는 그래프 형태: 변수 geom에 지정

```
> ggplot(alltime.movies,aes(x="", y=Gross)) +
      geom_boxplot() +
      stat_summary(fun.y="mean", geom="point",
                        color="red", shape=3, size=4, stroke=2) +
      labs(x="")
```



이상값으로 표시된 자료 확인

- 함수 ggplot_build( )의 결과물 이용

```
> bp <- ggplot(alltime.movies,aes(x=“ ”,y=Gross))+
        geom_boxplot()
```

```
> ggplot_build(bp)$data #ggplot_build(bp)[[1]]
[[1]]
  ymin lower middle upper ymax                      outliers notchupper
1  172   184      216   260   357 601, 461, 435, 431, 404     229.5101
  notchlower   x PANEL group ymin_final ymax_final    xmin   xmax xid
1   202.4899   1     1     1         172         601 0.625    1.375   1
  newx new_width weight colour      fill size alpha   shape  linetype
1    1         0.75      1 grey20 white  0.5    NA      19       solid
```

```
> my_out <- ggplot_build(bp)[[1]][[1]]$outlier

> str(my_out)
List of 1
 $ : num [1:5] 601 461 435 431 404
```

이상값으로 표시된 자료 확인

- 해당 자료 출력

```
> (my_out <-my_out[[1]])
> my_out
[1] 601 461 435 431 404
```

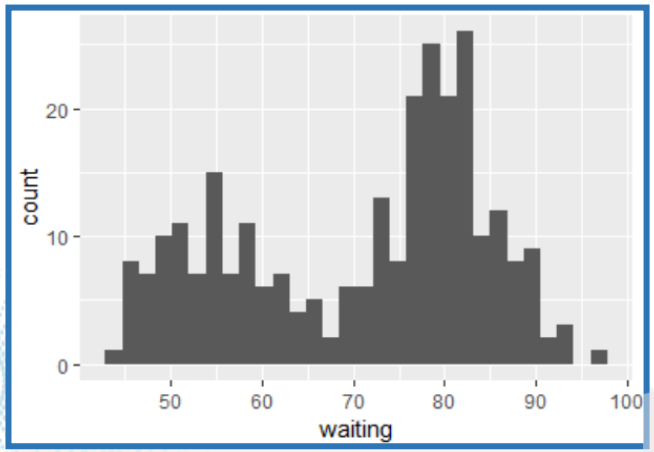
```
> alltime<-as_tibble(alltime.movies) %>%
+ rownames_to_column(var="Movie.Title")
>
> top_movies<-alltime%>%
+   filter(Gross %in% my_out)
> top_movies
# A tibble: 5 x 3
  Movie.Title                          Gross Release.Year
  <chr>                                <dbl>        <dbl>
1 "Titanic                              "   601         1997
2 "Star Wars                            "   461         1977
3 "E.T.                                 "   435         1982
4 "Star Wars: The Phantom Menace        "   431         1999
5 "Spider-Man                           "   404         2002
```

히스토그램

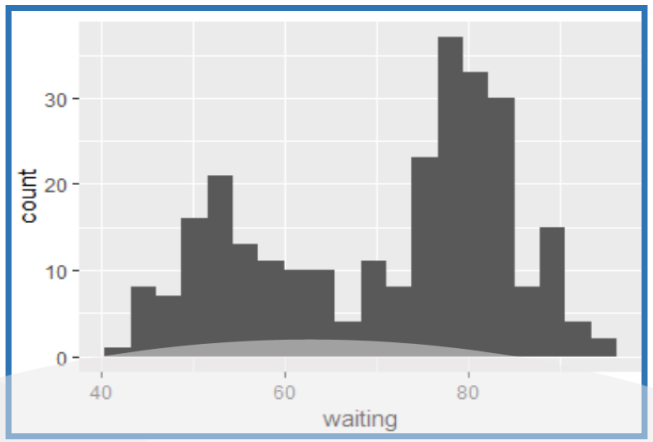
- 함수 geom_histogram ( )
- 히스토그램의 구간 조절: bins(구간의 개수) 혹은 binwidth(구간 폭)

▪ 예: faithful의 변수 waiting

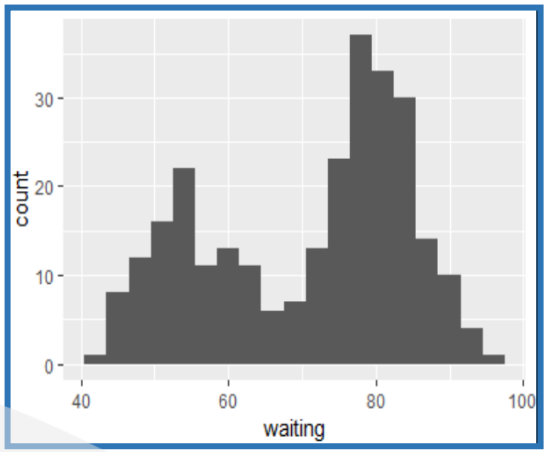
```
> h<-ggplot(faithful,aes(x=waiting))
> h+geom_histogram ( )
`stat_bin( )` using `bins = 30`. Pick better value with `binwidth`.
> h+geom_histogram (bins=20)
> h+geom_histogram (binwidth=3)
```



디폴트



bins=20



binwidth=3

확률밀도함수 그래프

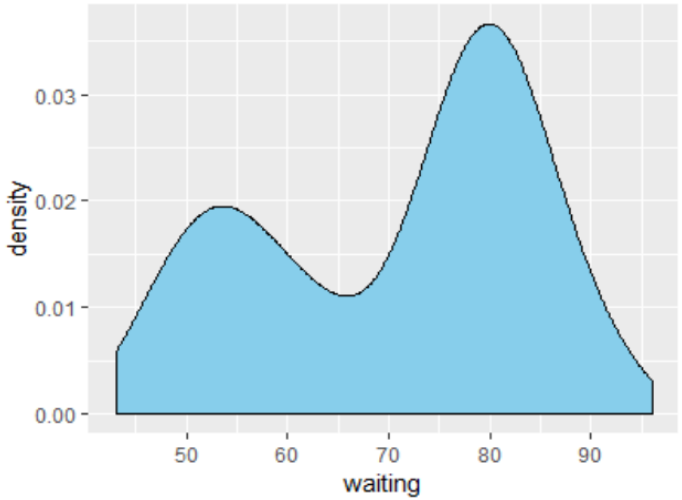
- 연속형 자료의 분포 표현에 가장 적합한 그래프

- 함수 geom_density( )로 작성

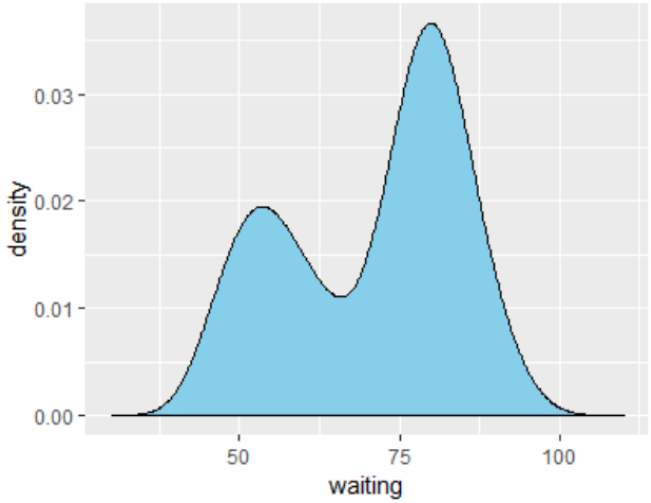
- 다른 그래프의 한계:
  상자그림: 분포의 세밀한 특징이 나타나지 않음
  히스토그램: 매끄럽지 않은 계단함수의 형태

예: faithful의 waiting 확률밀도함수

```
> p <- ggplot(faithful, aes(x=waiting)) +
        geom_density(fill="skyblue")
> p
> p + xlim(30,110)
```



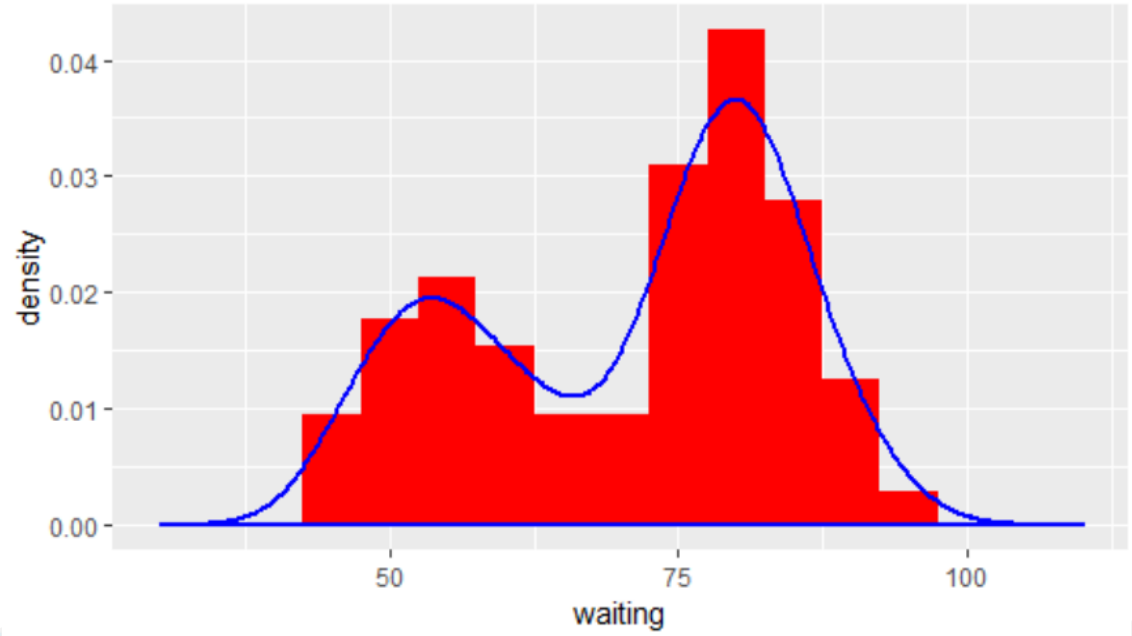
디폴트



X축 구간 확대
함수 xlim( ) 사용

히스토그램과 겹치게 작성

```
> ggplot(faithful, aes(x=waiting, y=stat(density))) +
    geom_histogram(fill="red", binwidth = 5) +
    geom_density(color="blue", size=1) +
    xlim(30,110)
```



함수 geom_density( )와
geom_histogram( )의
실행 순서를 바꾸면?

UsingR::cfb

- 2001년 미국 소비자 재정 상태에 대한 데이터
- 변수(Variable=Feature) INCOME: 가구당 소득
- 변수 INCOME의 분포 탐색

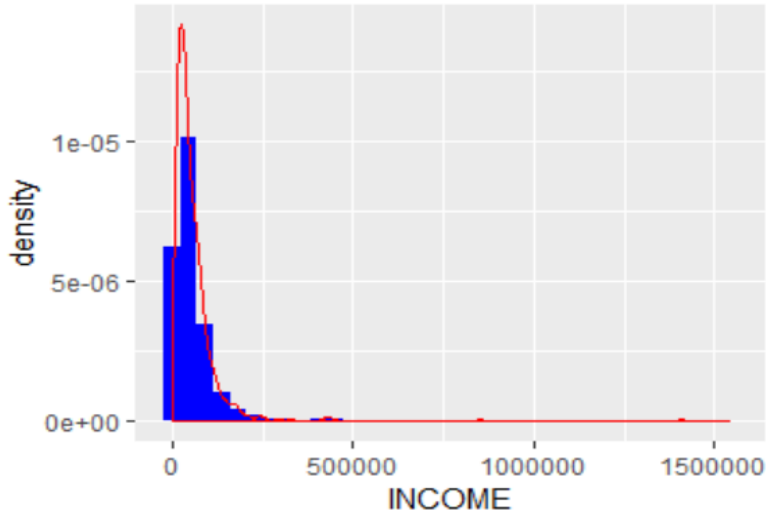
요약 통계량

```
> data(cfb, package = "UsingR")

> mean(cfb$INCOME)
[1] 63402.66

> median(cfb$INCOME)
[1] 38032.7
```

다른 방법을 상기해보자
(with, attach 등)

치우친 그래프

```
> ggplot(cfb, aes(x=INCOME,y=stat(density))) +
      geom_histogram(bins=35, fill="blue") +
      geom_density(color="red")
```



- 심하게 치우친 분포
- 로그변환으로 좌우대칭 분포로 변환 시도

로그 변환

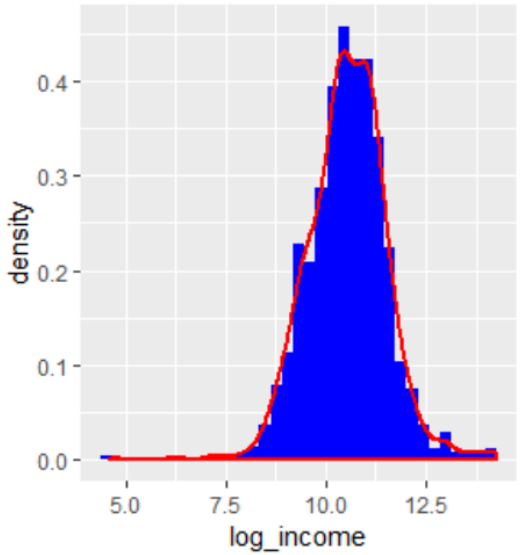
```
> log_income <- log(cfb$INCOME)

> range(log_income)
[1]       -Inf  14.2485
> range(cfb$INCOME)
[1]          0  1541866
```

- 로그 변환된 자료에 –inf 포함
- 변수 INCOME에 0이 있음
- 모든 데이터를 우측으로 1 이동

```
> log_income = log(cfb$INCOME+1)
> range(log_income)
[1]  0.0000  14.2485
```

로그 변환된 자료의 분포

```
> ggplot(data.frame(log_income), +
                aes(x=log_income, y=stat(density))) +
        geom_histogram(fill="blue", bins=35) +
        geom_density(color="red", size=1)
```




```
> mean(log_income); median(log_income)
[1] 10.49809
[1] 10.54623
```

이변량 자료 탐색

박영식(youngsik.park@bsl-lausanne.ch)

이변량 자료 탐색

◎ 이변량 자료의 분석
- 각 변수의 개별 분포 파악
- 두 변수의 분포 비교
- 두 변수의 관계 탐색

◎ 이변량 범주형 자료
- 막대 그래프: 쌓아 올린 형태, 옆으로 붙여 놓은 형태
- Mosaic plot

◎ 이변량 연속형 자료
- 분포 비교를 위한 그래프
- 관계 탐색을 위한 그래프

1. 연속형 변수의 분포를 비교하기 위한 그래프

◎ 예제: mpg의 변수 cyl에 따른 hwy의 분포 비교

- cyl로 구분되는 그룹에 속한 자료의 개수

```
> mpg %>% group_by(cyl) %>% summarise(n=n())
# A tibble: 4 x 2
   cyl     n
 <int> <int>
1    4    81
2    5     4
3    6    79
4    8    70
```

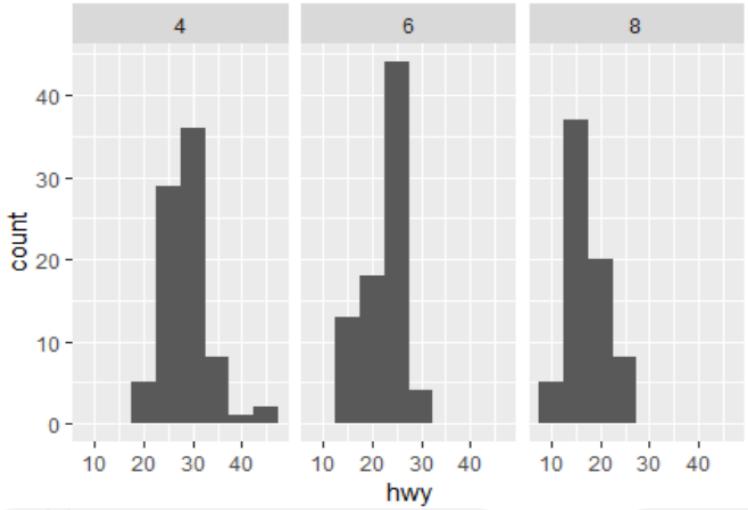
- Cyl이 5가 되는 자료의 개수가 너무 작음
- Cyl이 4,6,8인 그룹에 대해서만 hwy의 분포를 비교해보자

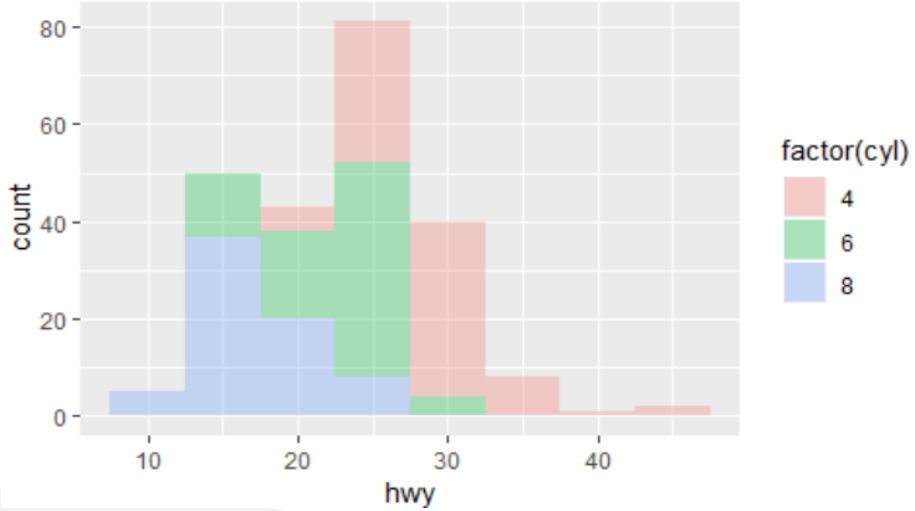
```
> mpg_1 < - mpg %>% filter(cyl!=5)
```

히스토그램에 의한 그룹 자료의 분포 비교

```
> ggplot(mpg_1, aes(x=hwy) ) +
        geom_histogram(binwidth=5) +
        facet_wrap(~ cyl)
```

```
> ggplot(mpg_1, aes(x=hwy, fill=factor(cyl))) +
        geom_histogram(binwidth=5, alpha=0.3)
```

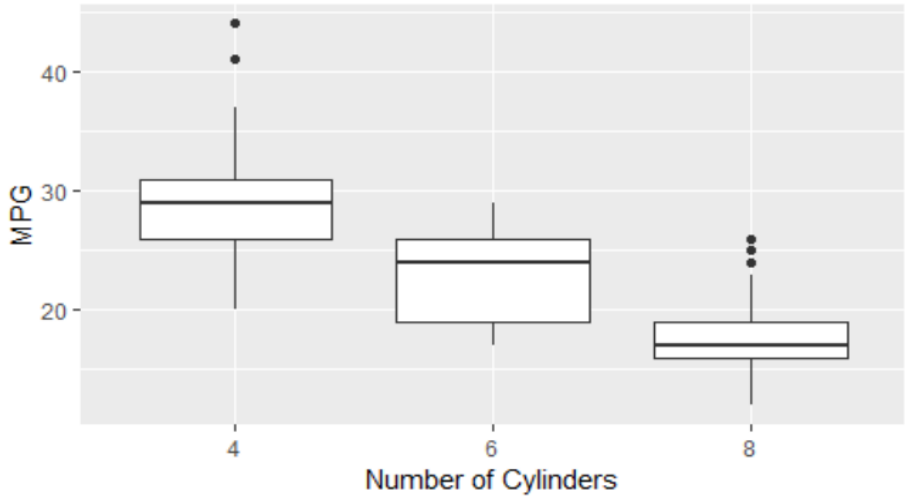




- 그룹간 분포 비교가 용이하지 않음.

상자 그림에 의한 그룹 자료의 분포 비교

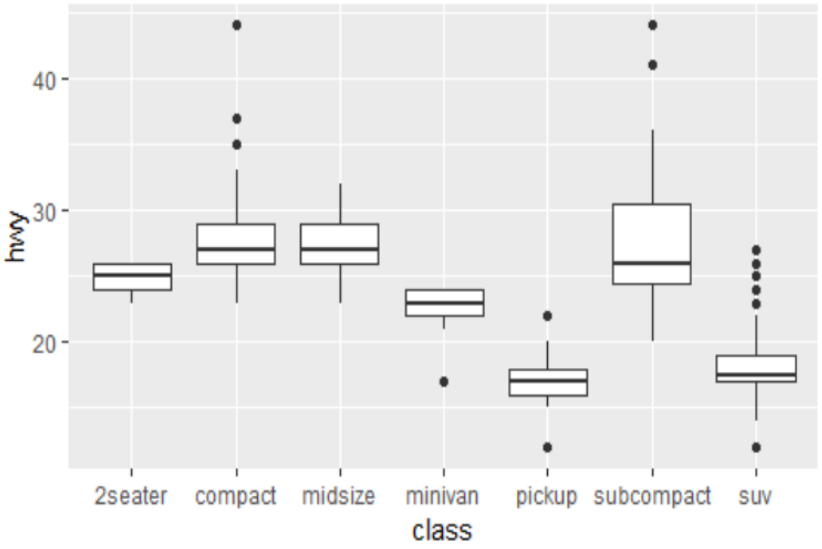
```
> ggplot(mpg_1, aes(x=factor(cyl), y=hwy)) +
      geom_boxplot() +
      labs(x="Number of Cylinders", y="MPG")
```



상자 그림에 의한 그룹 자료의 분포 비교

- mpg의 변수 hwy의 상자그림을 class의 수준별로 작성

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



- class의 범주 순서에 따라 상자그림 배열

- hwy의 중앙값에 따라 배열하는 것이 분포 비교에 더 좋음

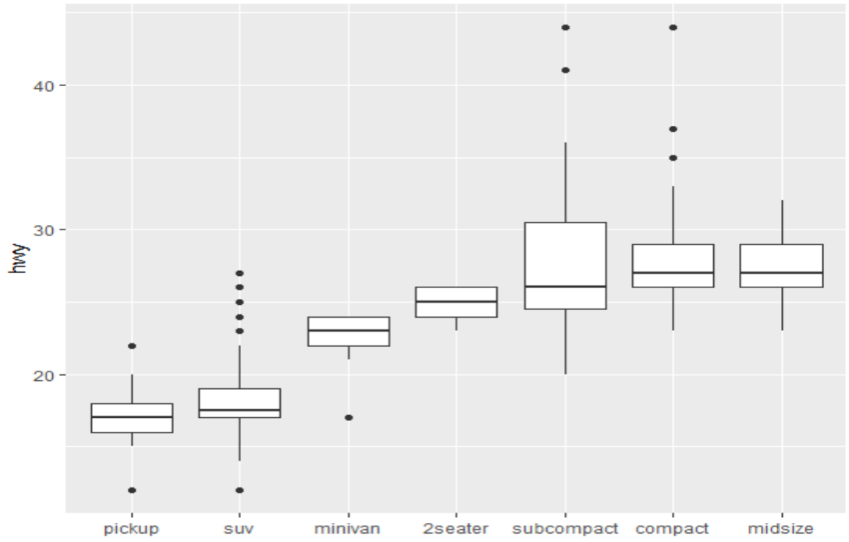
- class의 범주 수준을 hwy의 중앙값을 기준으로 다시 배열

reorder(class, hwy, FUN=median)

상자 그림에 의한 그룹 자료의 분포 비교

- mpg의 변수 hwy의 상자그림을 class의 수준별로 작성

```
> ggplot(mpg, aes(x=reorder(class, hwy, FUN=median), y=hwy)) +
    geom_boxplot() +
    labs(x="")
```

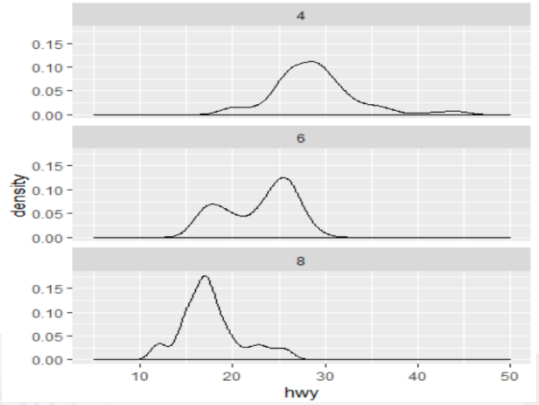


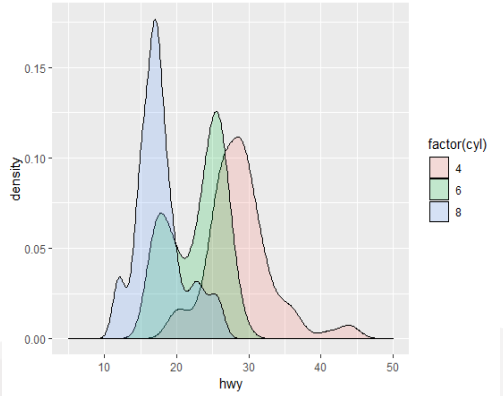
확률밀도함수 그래프에 의한 그룹 자료의 분포 비교

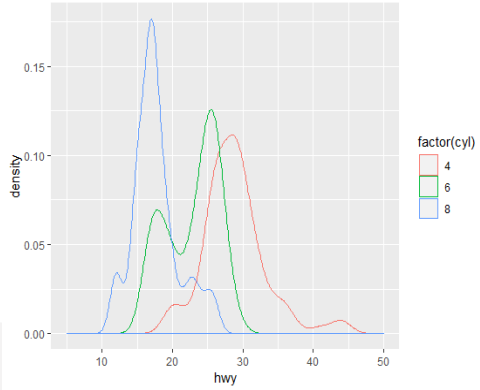
```
> ggplot(mpg_1, aes(x=hwy)) +
        geom_density() +
        xlim(5,50) +
        facet_wrap(~cyl, ncol=1)
```

```
> ggplot(mpg_1, aes(x=hwy, fill=factor(cyl))) +
        geom_density(alpha=0.2) +
        xlim(5,50)
```

```
> ggplot(mpg_1, aes(x=hwy, color=factor(cyl))) +
        geom_density() +
        xlim(5,50)
```





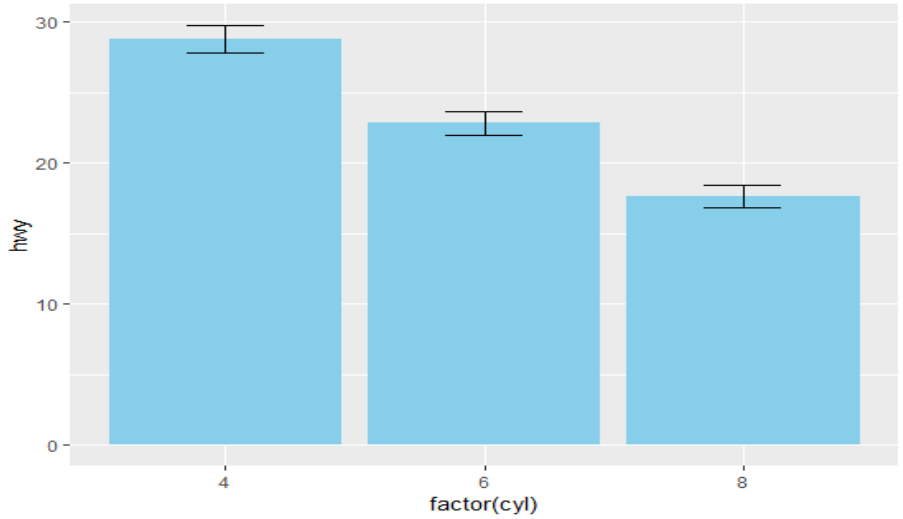


평균 막대 그래프와 error bar에 의한 그룹 자료의 평균값 비교

- 그룹별 자료의 평균 비교에 대해 막대 그래프를 이용

- Error bar: 분포의 변동 혹은 신뢰구간을 표시하는 그래프

  - mpg의 변수 cyl에 따른 hwy의 평균 및 신뢰구간



-막대 그래프: 변수 cly에 따른 hwy의 평균

-Error bar: 각 그룹별 hwy의 95% 신뢰구간

작성 방법

1)그룹별 자료의 평균 비교에 대해 막대 그래프를 이용

```
> hwy_stat
# A tibble: 3 x 6
    cyl mean_hwy sd_hwy n_hwy ci_low ci_up
  <int>    <dbl>  <dbl> <int>  <dbl> <dbl>
1     4     28.8   4.52    81   27.8  29.8
2     6     22.8   3.69    79   22.0  23.6
3     8     17.6   3.26    70   16.9  18.4
```

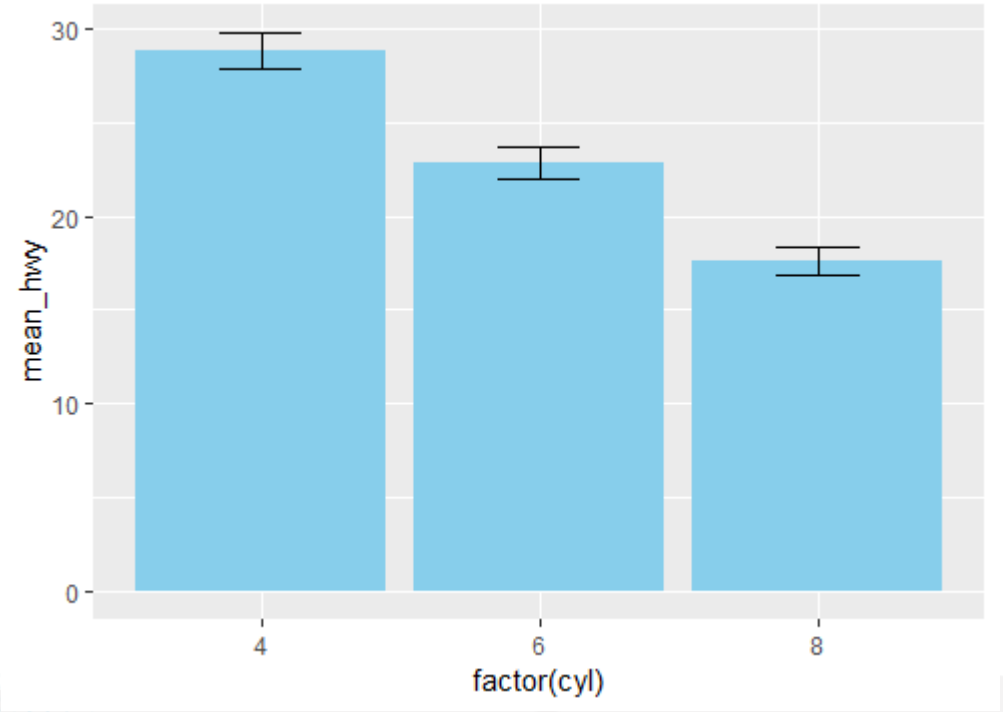
-평균: mean_hwy
-신뢰구간 상한: ci_up
-신뢰구간 하한: ci_low

- hwy_stat의 계산

```
> hwy_stat <- mpg %>%
        filter(cyl!=5) %>%
        group_by(cyl) %>%
        summarize(mean_hwy=mean(hwy), sd_hwy=sd(hwy),
                  n_hwy=n(),
                  ci_low=mean_hwy-qt(0.975,df=n_hwy-1)*sd_hwy/sqrt(n_hwy),
                  ci_up=mean_hwy+qt(0.975,df=n_hwy-1)*sd_hwy/sqrt(n_hwy))
```

hwy 자료를 통한 막대 그래프 및 error bar 작성

```
> ggplot(hwy_stat, aes(x=factor(cyl), y=mean_hwy)) +
        geom_col(fill="skyblue") +
        geom_errorbar(aes(ymin=ci_low, ymax=ci_up), width=0.3
```



작성 방법

2) 원 자료만 주어진 경우

```
> mpg %>% filter(cyl!=5) %>%
        ggplot(aes(x=factor(cyl), y=hwy)) +
        stat_summary(fun.y="mean", geom="bar", fill="skyblue") +
        stat_summary(fun.data="mean_cl_normal", geom="errorbar", width=0.3)
```

- fun.data: ymin과 y, ymax를 계산할 수 있는 함수 지정

- mean_cl_normal( ): 정규분포를 가정하며 모평균의 신뢰구간 계산

2. 연속형 변수의 관계 탐색을 위한 그래프: 산점도

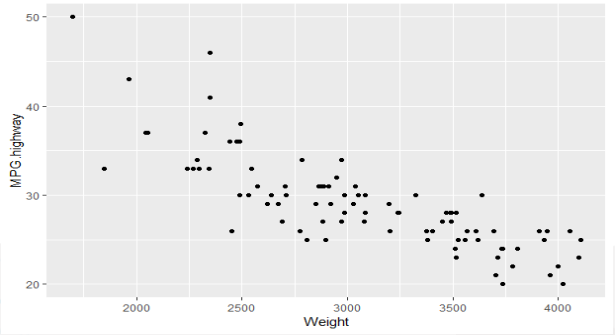
1) 다양한 유형의 산점도 작성

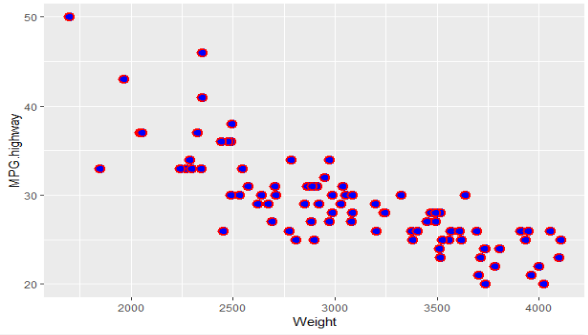
◎ 기본적인 형태의 산점도: Cars93의 weight와 MPG . highway

```
> data(Cars93, package="MASS")
```

```
> ggplot(Cars93, aes(x=weight, y=MPG.highway)) +
        geom_point()
```

```
> ggplot(Cars93, aes(x=Weight, y=MPG.highway)) +
        geom_point(shape=21, color="red", fill="blue",
                   stroke=1.5, size=3)
```





2. 연속형 변수의 관계 탐색을 위한 그래프: 산점도

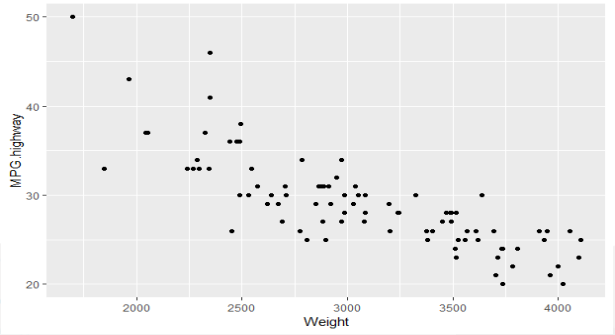
1) 다양한 유형의 산점도 작성

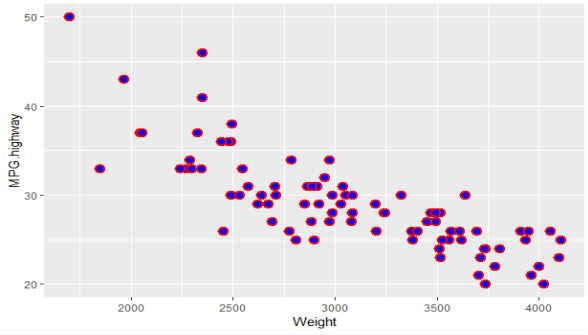
◎ 기본적인 형태의 산점도: Cars93의 weight와 MPG . highway

```
> data(Cars93, package="MASS")
```

```
> ggplot(Cars93, aes(x=weight, y=MPG.highway)) +
        geom_point()
```

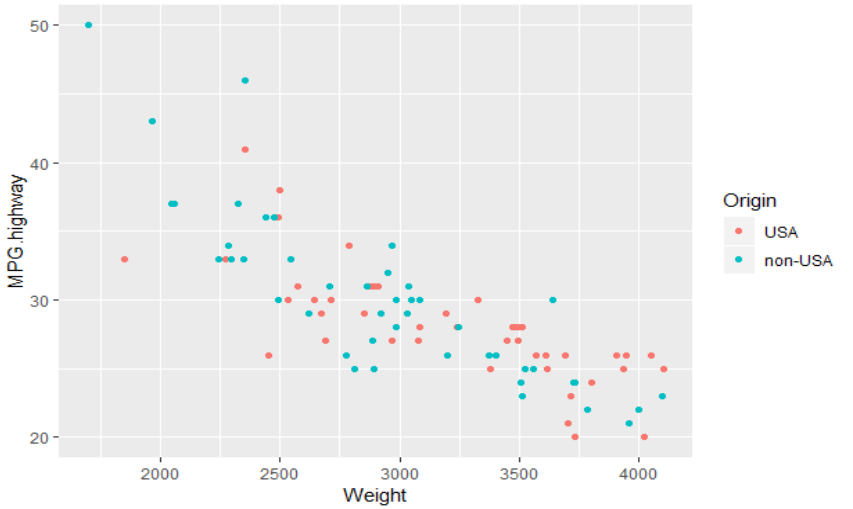
```
> ggplot(Cars93, aes(x=Weight, y=MPG.highway)) +
        geom_point(shape=21, color="red", fill="blue",
                   stroke=1.5, size=3)
```



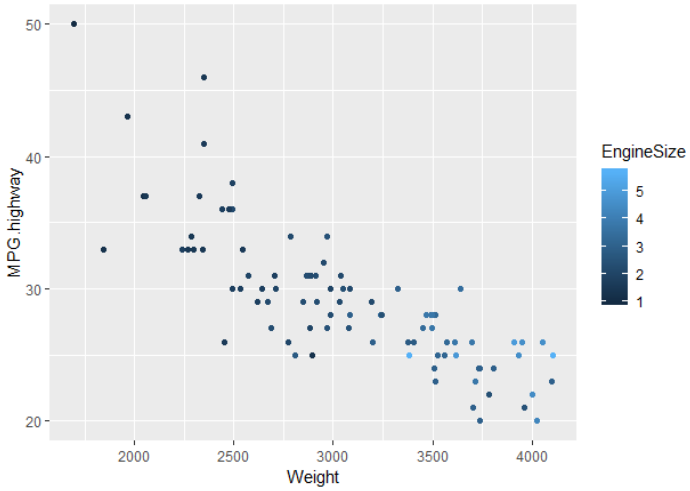


시각적 요소에 또 다른 변수를 매핑(mapping)

```
> ggplot(Cars93, aes(x=Weight, y=MPG.highway, color=Origin)) +
        geom_point()

> ggplot(Cars93, aes(x=Weight, y=MPG.highway, color=EngineSize)) +
        geom_point()
```



- Color에 요인 매핑

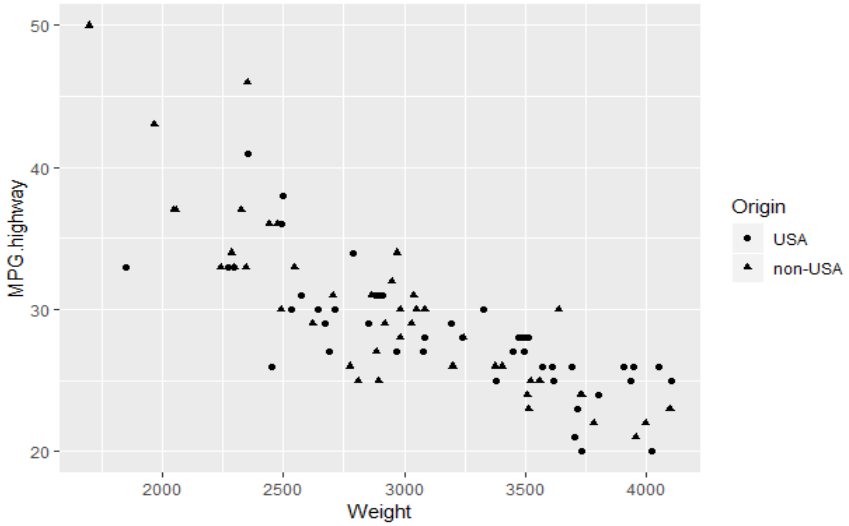


- Color에 숫자형 변수 매핑

– shape에 요인 및 숫자형 변수 매핑

```
> ggplot(Cars93, aes(x=Weight, y=MPG.highway, shape=Origin)) +
      geom_point()
```

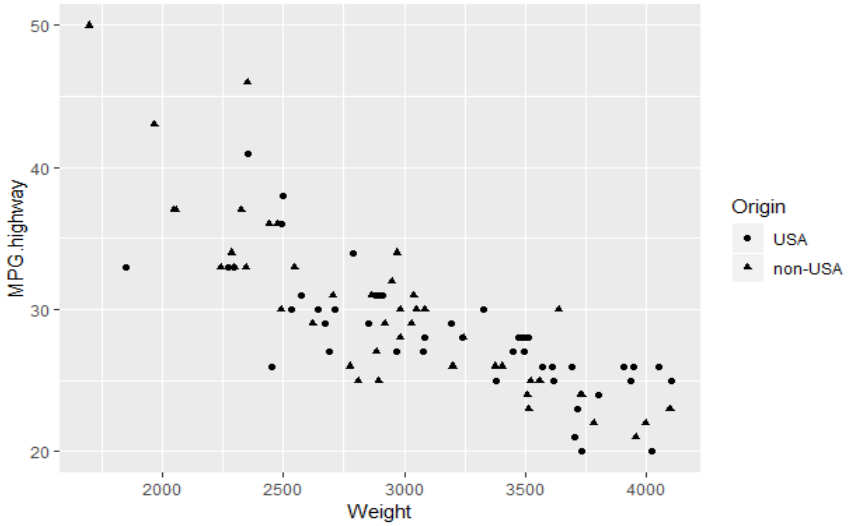
```
> ggplot(Cars93, aes(x=Weight, y=MPG.highway, shape=EngineSize)) +
      geom_point()
```



– shape에 요인 및 숫자형 변수 매핑

```
> ggplot(Cars93, aes(x=Weight, y=MPG.highway, shape=Origin)) +
      geom_point()
```

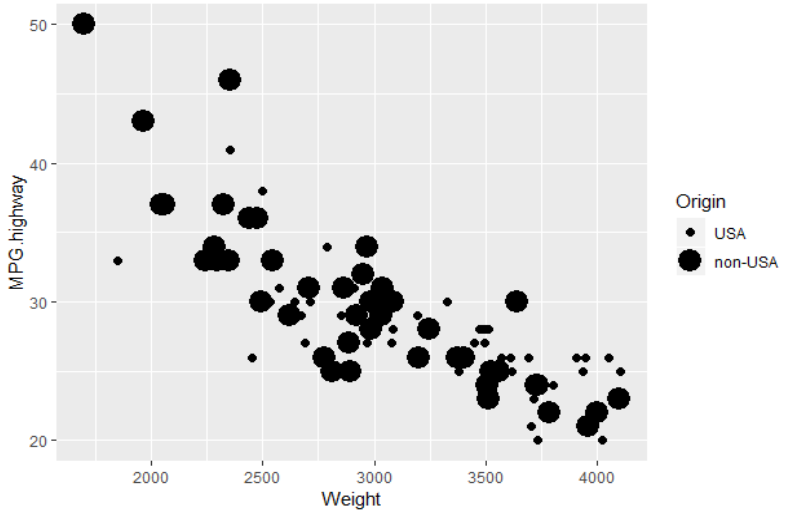
```
> ggplot(Cars93, aes(x=Weight, y=MPG.highway, shape=EngineSize)) +
      geom_point()
```

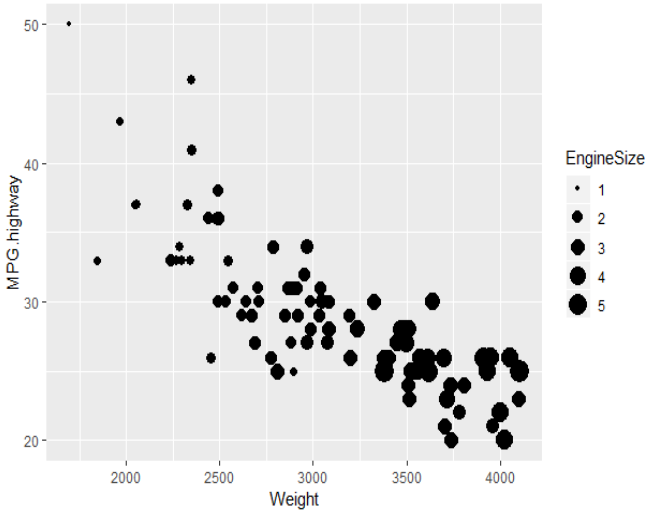


– size에 요인 및 숫자형 변수 매핑

```
> ggplot(Cars93, aes(x=Weight, y=MPG.highway, size=Origin)) +
      geom_point()
```

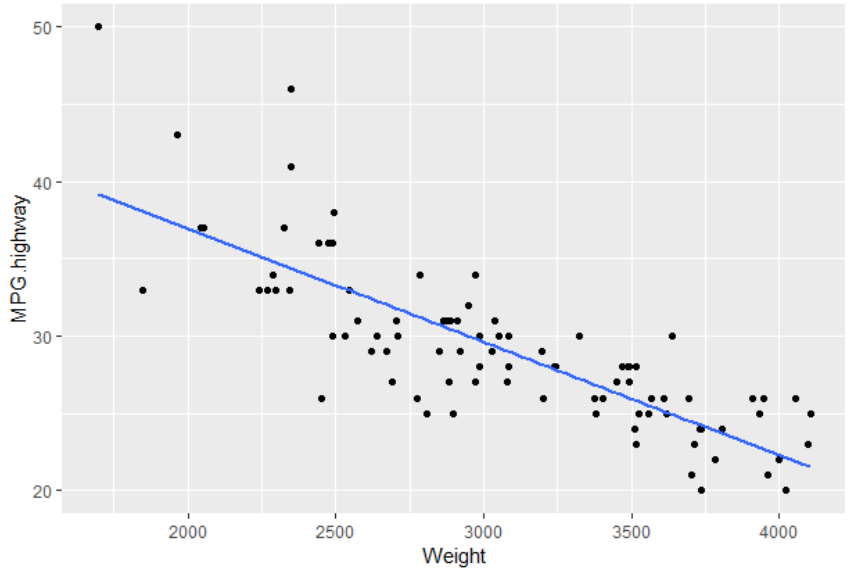
```
> ggplot(Cars93, aes(x=Weight, y=MPG.highway, size=EngineSize)) +
      geom_point()
```

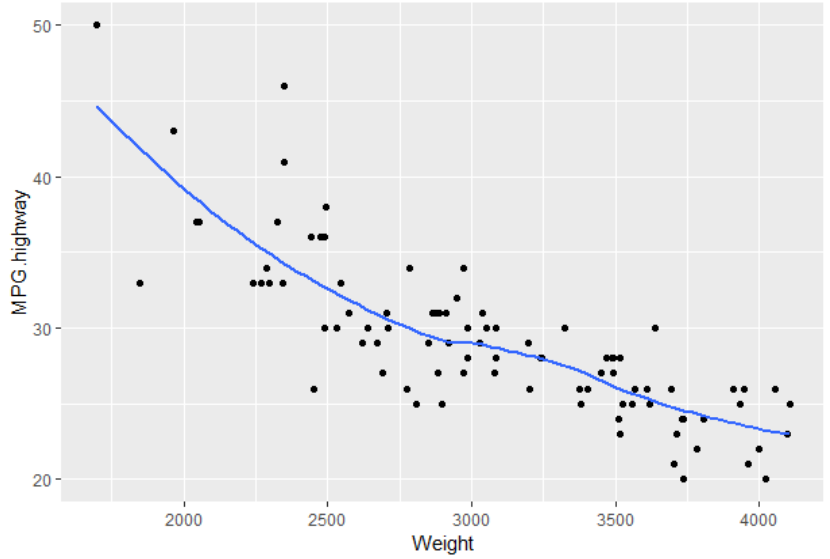




산점도에 회귀직선 추가

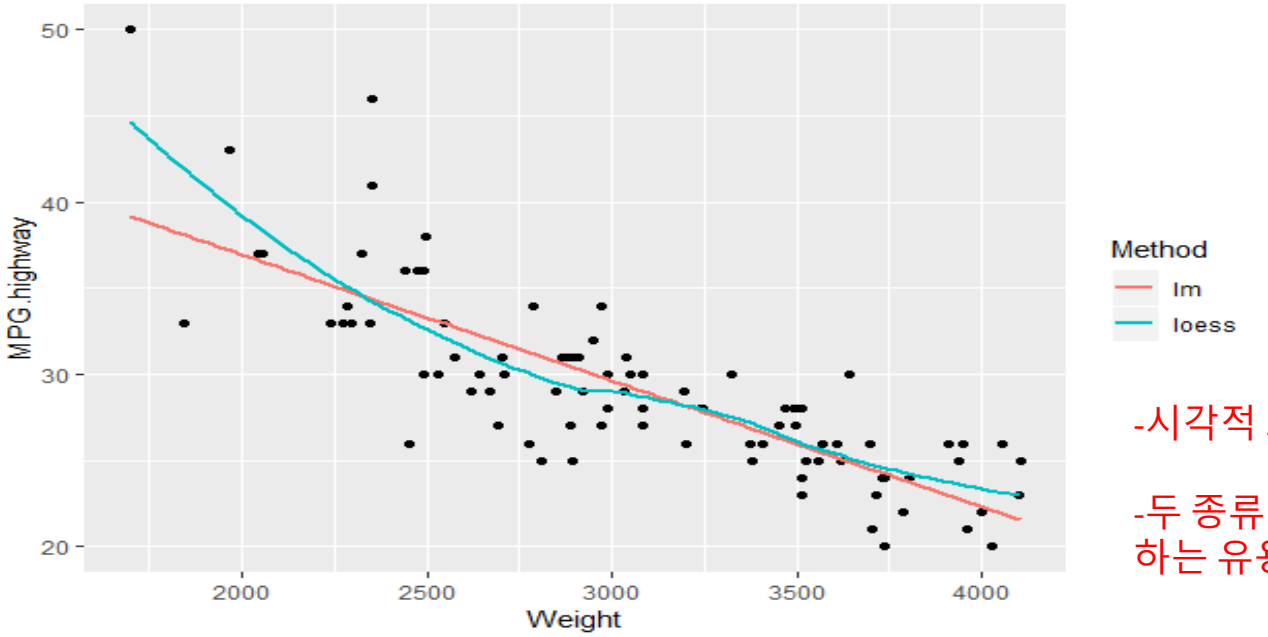
```
> ggplot(Cars93, aes(x=Weight, y=MPG.highway)) +
        geom_point() +
        geom_smooth(method="lm", se=FALSE)

> ggplot(Cars93, aes(x=Weight, y=MPG.highway)) +
        geom_point() +
        geom_smooth(se=FALSE)
```





– 회귀직선과 비모수 회귀곡선을 함께 산점도에 추가

```
> ggplot(Cars93, aes(x=Weight, y=MPG.highway)) +
        geom_point() +
        geom_smooth(aes(color="lm"), method="lm", se=FALSE) +
        geom_smooth(aes(color="loess"), se=FALSE) +
        labs(color="Method")
```



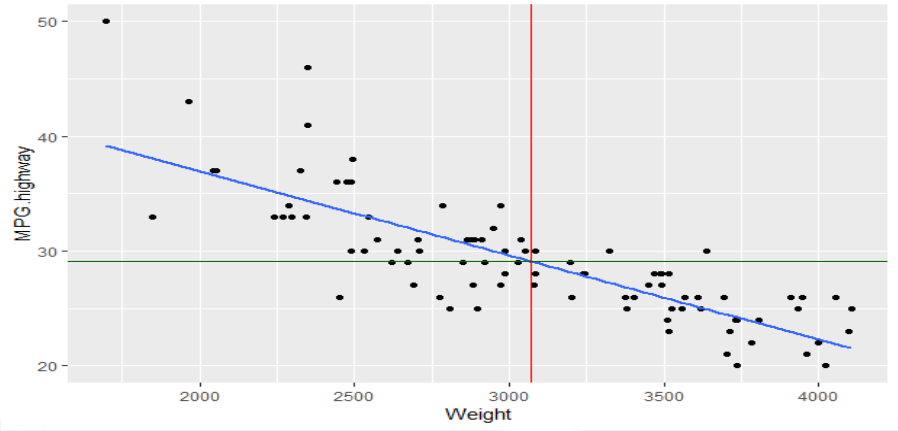
-시각적 요소에 문자열 매핑

-두 종류의 선에 legend를 추가하는 유용한 방법

산점도에 수평선, 수직선 추가

- 직선 추가 함수: geom_abline(slope, intercept)
- 수직선 추가 함수: geom_vline(xintercept)
- 수평선 추가 함수: geom_hline(yintercept)

```
> ggplot(Cars93, aes(x=Weight, y=MPG.highway)) +
        geom_point() +
        geom_smooth(method="lm", se=FALSE) +
        geom_vline(aes(xintercept=mean(Weight)), color= " red") +
        geom_hline(aes(yintercept=mean(MPG.highway)), color= " dark green")
```

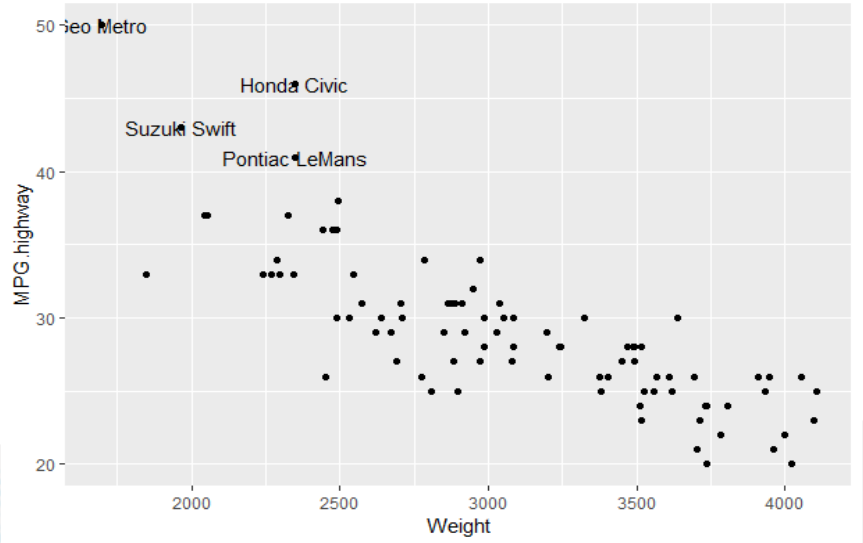


-산점도
-산점도
-두 종류의 선에 legend를 추가하는 유용한 방법

산점도의 점에 라벨 추가

- Weight와 MPG.highway의 산점도
- MPG.highway >40인 점에 라벨 추가
- 라벨 내용: Manufacturer와 Model의 값을 결합

```
> p <- ggplot(Cars93, aes(x=Weight, y=MPG.highway)) +
        geom_point()
> p + geom_text(data=filter(Cars93, MPG.highway>40),
                    aes(label=paste(Manufacturer, Model)))
```



-라벨의 위치 조정이 필요

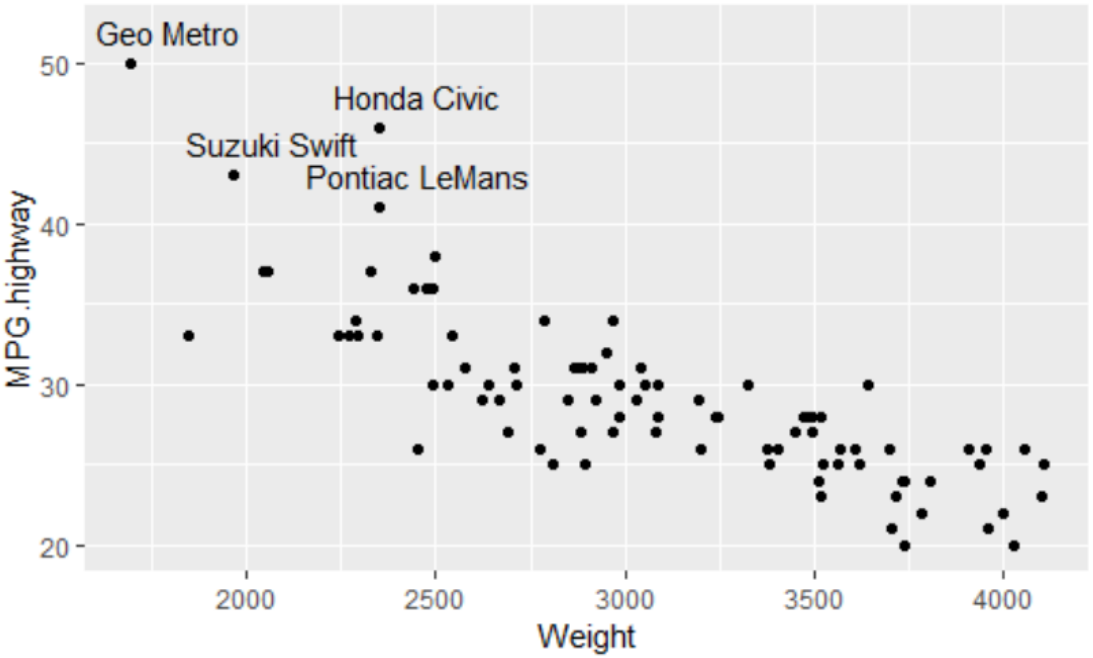
-라벨 위치 조정

① vjust, hjust

② nudge_x, nudge_y

– 라벨 위치 조정: nudge_x & nudge_y 이용

```
> p + geom_text(data=filter(Cars93, MPG.highway>40),
                aes(label=paste(Manufacturer, Model)),
                nudge_y=2, nudge_x=100)
```



-nudge_x:

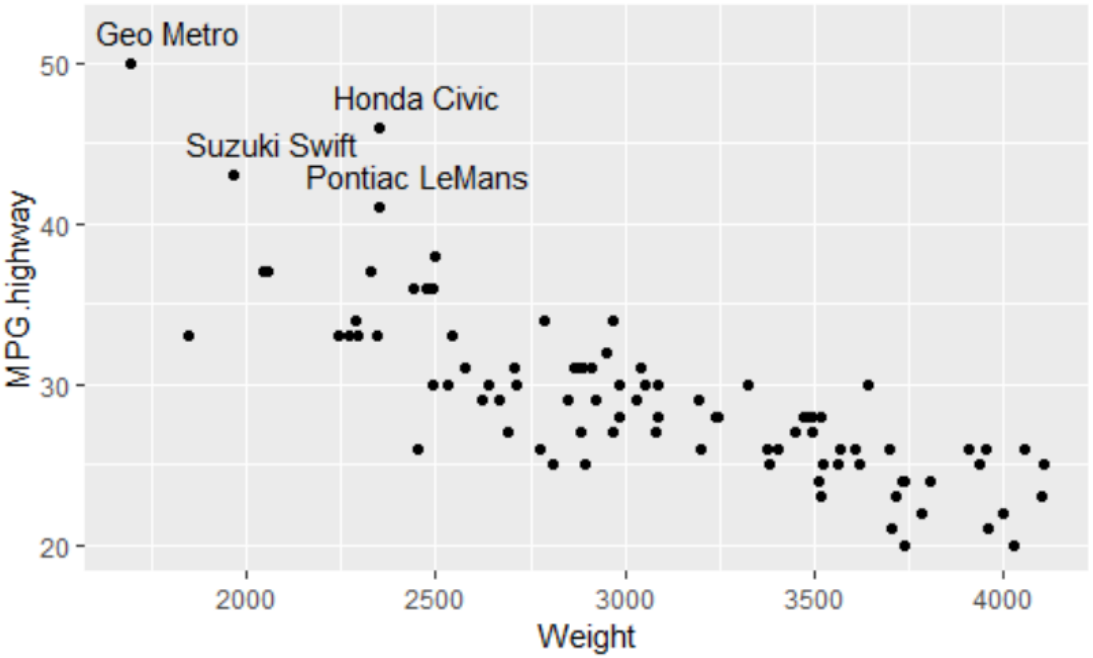
양의 값: 우측으로 이동
음의 값: 좌측으로 이동

-nudge_y:

양의 값: 위로 이동
음의 값: 아래로 이동

– 라벨 위치 조정: nudge_x & nudge_y 이용

```
> p + geom_text(data=filter(Cars93, MPG.highway>40),
                aes(label=paste(Manufacturer, Model)),
                nudge_y=2, nudge_x=100)
```



-nudge_x:

양의 값: 우측으로 이동
음의 값: 좌측으로 이동

-nudge_y:

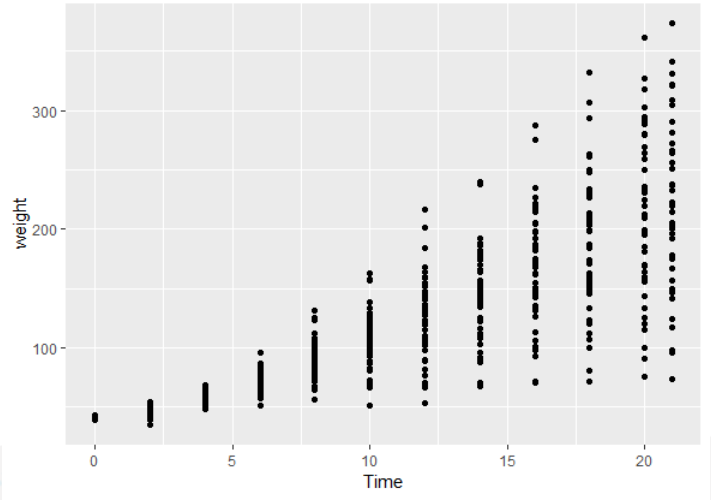
양의 값: 위로 이동
음의 값: 아래로 이동

2) 산점도에서 점이 겹쳐질 경우

- 대규모 자료일 경우
- 두 변수 중 한 변수가 이산형인 경우
- 자료가 반올림될 경우

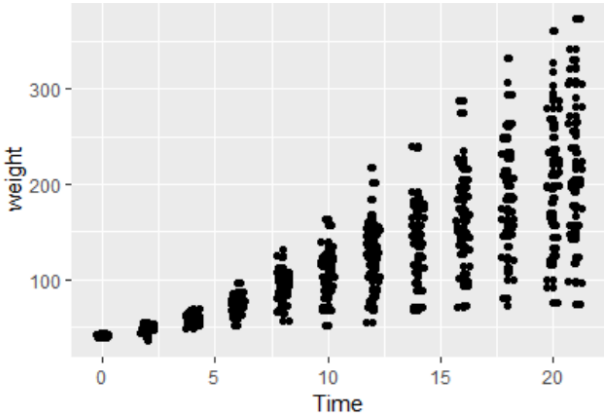
① 한 변수가 이산형인 경우의 예: Chickweight의 변수 Time과 weight

```
> p1 <- ggplot(ChickWeight, aes(x=Time, y=weight))
> p1 + geom_point()
```



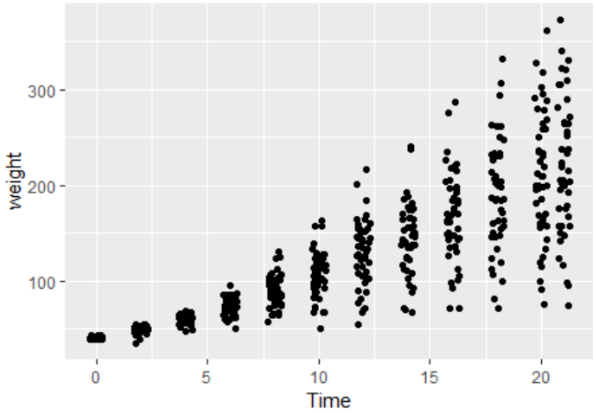
대안 1: jittering

```
> p1 + geom_point()+geom_jitter(width=0.3, height=0)
```



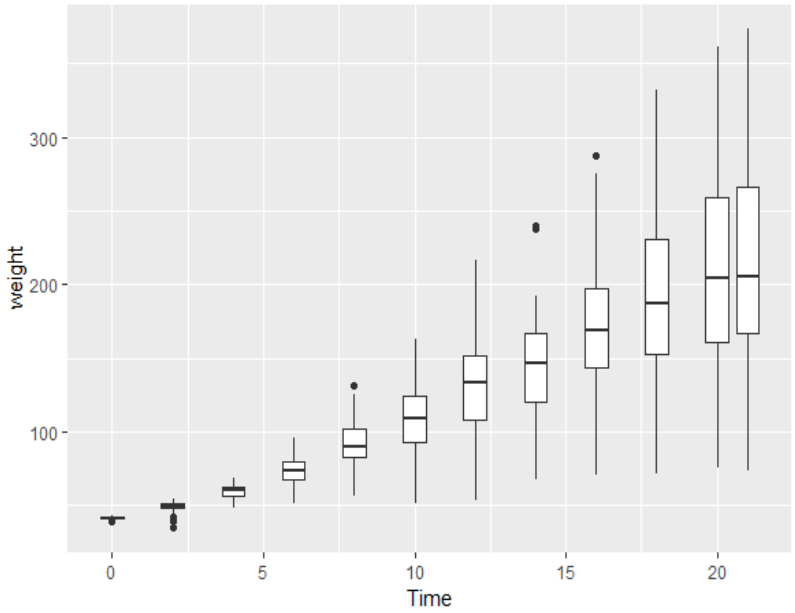
VS

```
> p1 + geom_jitter(width=0.3, height=0)
```



대안 2: 상자그림

```
> p1 + geom_boxplot(aes(group=Time))
```



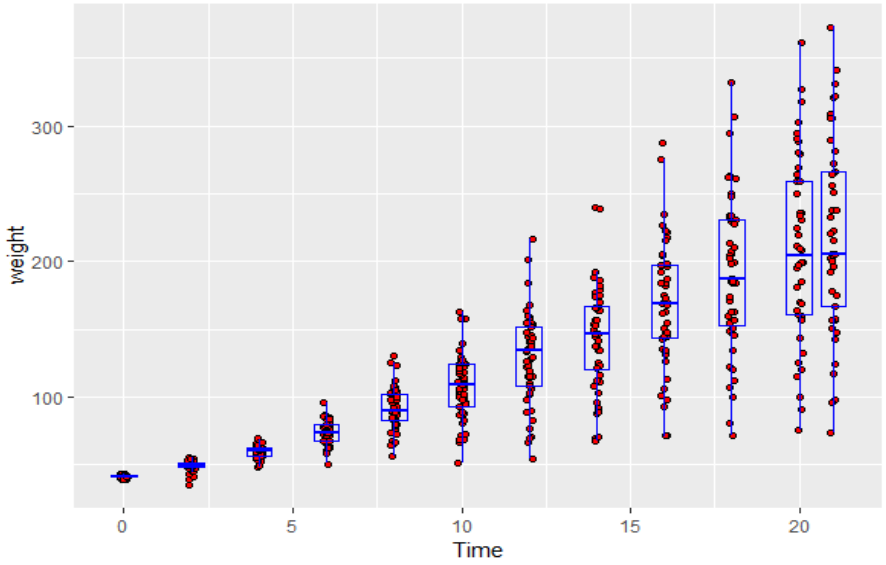

```
> class(ChickWeight$Time)
 [1] "numeric"
> typeof(ChickWeight$Time)
[1] "double"
```

- x 변수인 Time이 숫자형 변수

- 시각적 요소 group에 x를 매핑

대안 3: 상자그림과 jittering

```
> p1 + geom_jitter(width=0.1, fill="red", shape=21) +
        geom_boxplot(aes(group=Time), outlier.shape=NA,
                              fill=NA, color="blue")
```



fill = NA
- 상자 내부의 흰 배경 제거
- geom_boxplot을 먼저 실행 후 그 위에 점 jittering시에는 소용이 없다! 왜 그럴까??

주요 이력
現) ㈜RTMC 전략기획실장
前) ㈜B사 웹로그분석 및 DP사업 完
前) ㈜H금속사 회계팀
前) ㈜B건설사 회계팀
前) K문고 CRM VIP 군집전략 CRM프로젝트 보조연구원
前) L백화점 CRM Alert 전략 CRM프로젝트 보조연구원

BSL(스위스 로잔 비즈니스 스쿨) MBA
ASSIST 빅데이터경영통계 MBA

국가공인 ADSP(빅데이터 준전문가)
現 **코리아IT아카데미 빅데이터 R 강사**
現 **코리아IT아카데미 빅데이터 기초 파이썬 강사**
現 **코리아IT아카데미 빅데이터 기초통계 전담강사**

"자료는 대가이신 박동련 교수님께 도움을 받았음을 밝힙니다."

[박영식] 완성에 이르기까지