8강

R 통계 그래픽스(2)

이화여대 통계학과 이은경교수

목 차

1 그래픽 문법

그래픽 문법

- 그래픽을 위한 요소들을 잘 이용할 수 있도록 정리해 놓은 도구
- 2005년 Wilkinson에 의해 이론이 확립
- 통계 그래픽을 기반이 되는 요소들 각각으로 나누어 이를 문법의 형태로 정리한 것으로 자료를 살펴보기 위한 도구로써의 통계 그래픽이 제 역할을 잘 할 수 있도록 해줌
- 여러 형태의 그림을 그래픽 문법으로 통합하여 하나의 큰 틀에서 다룰 수 있도록 함
- 층화 그래픽 문법(layered grammar of graphics)
  - ggplot2 라이브러리를 위하여 그래픽 문법을 좀 더 쉽게 접근할 수 있도록 바꾸어 놓은 것
  - qplot, ggplot의 두 함수를 이용하여 다양한 그래픽을 구현

2 자료 소개

abalone 자료

- 전복의 나이를 예측하기 위하여 성별, 길이, 무게 등을 측정한 자료
- UCI machine learning repository에 저장
- 변수 설명
  - sex : 수컷(M), 암컷(F), 유아기(I)
  - diameter : length에 수직인 길이
  - wholeW : 전체 무게
  - visceraW : 내장 무게
  - rings : 링의 수(나이를 나타냄)
  - length : 껍질 중 가장 긴 부분의 길이
  - height : 껍질 안의 몸통 길이
  - shuckedW : 껍질을 제외한 무게
  - shellW : 껍질 무게

abalone 자료

```
> abalone<-read.csv("abalone.csv", header=TRUE)
> dim(abalone)
[1] 4177    9
> head(abalone)
  sex length diameter height wholeW shuckedW visceraW shellW rings
1   M  0.455    0.365  0.095  0.5140   0.2245   0.1010  0.150    15
2   M  0.350    0.265  0.090  0.2255   0.0995   0.0485  0.070     7
3   F  0.530    0.420  0.135  0.6770   0.2565   0.1415  0.210     9
4   M  0.440    0.365  0.125  0.5160   0.2155   0.1140  0.155    10
5   I  0.330    0.255  0.080  0.2050   0.0895   0.0395  0.055     7
6   I  0.425    0.300  0.095  0.3515   0.1410   0.0775  0.120     8
```

Pconsump 자료

- 가정전력사용자료로 원 자료는 UCI machine learning repository에 저장
- 2006년 12월부터 2010년 11월까지 47개월간 프랑스의 한 지역에서 수집된 원 자료 중 2006년 12월 17일, 18일, 그리고 2007년 12월 17일, 18일을 추출한 자료
- 변수 설명
  - Date : 날짜
  - Time : 시간
  - X1 : 평균 유효전력
  - X2 : 평균 무효전력
  - X3 : 평균 전압
  - X4 : 전류 강도
  - X5 : 보조계량기1
  - X6 : 보조계량기2
  - X7 : 보조계량기3

Pconsump 자료

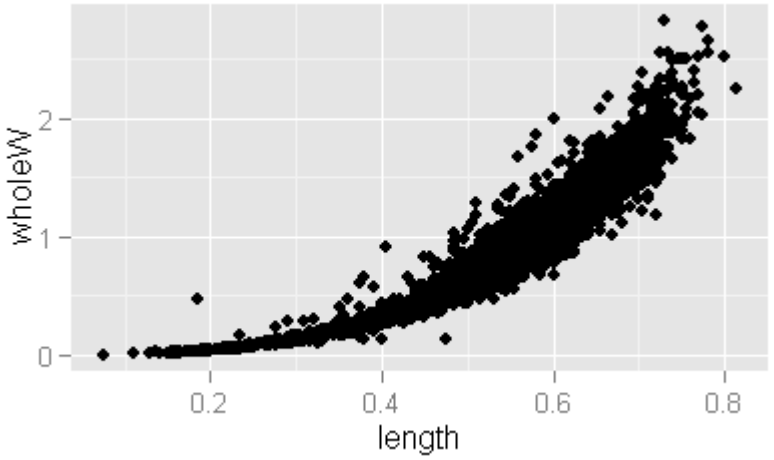
```
> Pconsump<-read.csv("power_consumption.csv")
> dim(Pconsump)
[1] 5760    9
> head(Pconsump)
        Date     Time    X1    X2     X3   X4 X5 X6 X7
1 17/12/2006 0:00:00 1.044 0.152 242.73 4.4  0  2  0
2 17/12/2006 0:01:00 1.520 0.220 242.20 7.4  0  1  0
3 17/12/2006 0:02:00 3.038 0.194 240.14 12.6 0  2  0
4 17/12/2006 0:03:00 2.974 0.194 239.97 12.4 0  1  0
5 17/12/2006 0:04:00 2.846 0.198 240.39 11.8 0  2  0
6 17/12/2006 0:05:00 2.848 0.198 240.59 11.8 0  1  0
```

3 qplot 함수를 이용한 그래프

산점도 (1)

- qplot함수는 plot이나 xyplot 함수와 비슷한 형태로 이용할 수 있으며 좀 더 편리한 기능을 제공
- 산점도를 위하여 x축 변수이름, y축 변수이름, 그리고 자료의 이름을 지정

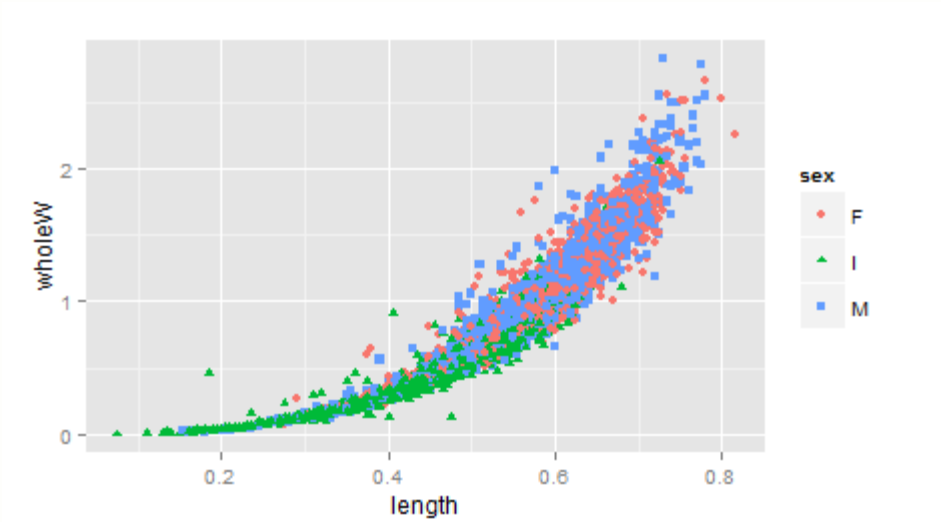
```
> library(ggplot2)
> qplot(length, wholeW, data = abalone)
```



산점도 (2)

- color = 범주형 변수 와 shape = 범주형 변수 옵션을 이용하여 범주형 변수의 그룹별로 다른 색과 다른 모양의 점을 이용할 수 있음

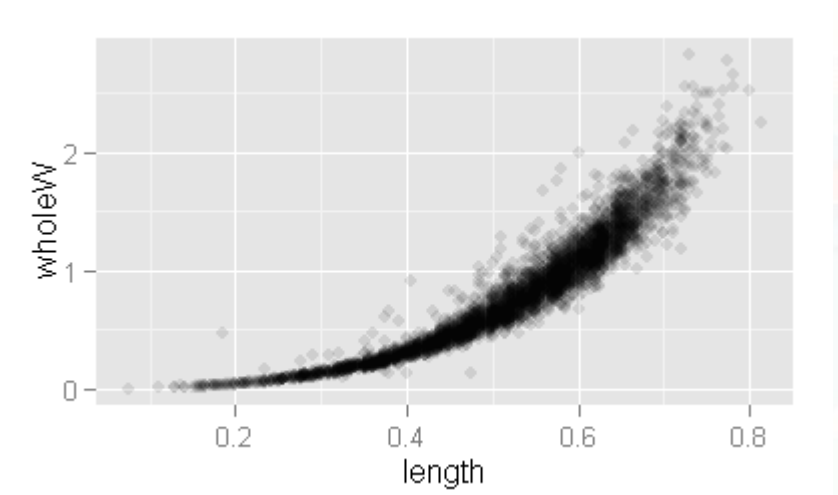
```
> qplot(length, wholeW, data = abalone, colour = sex, shape = sex)
```



산점도 (3)

- 관측의 수가 많은 경우 산점도에서 대부분의 점들이 겹쳐져서 그리는 문제를 해결하기 위하여 alpha = 불투명도 옵션을 이용
- 불투명도에서 1은 불투명, 0은 투명을 의미함

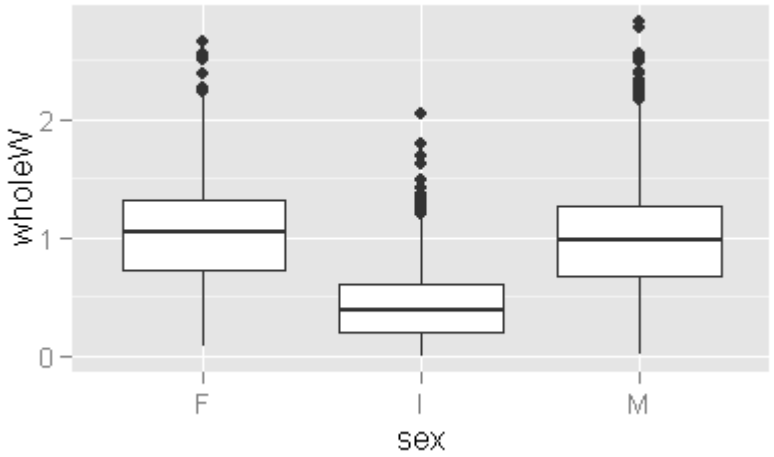
```
> qplot(length, wholeW, data = abalone, alpha=I(1/10))
```



상자그림 (1)

- geom = “boxplot” 옵션을 이용
- X축 변수에 범주형 변수를 이용하면 평행상자그림을 그려줌

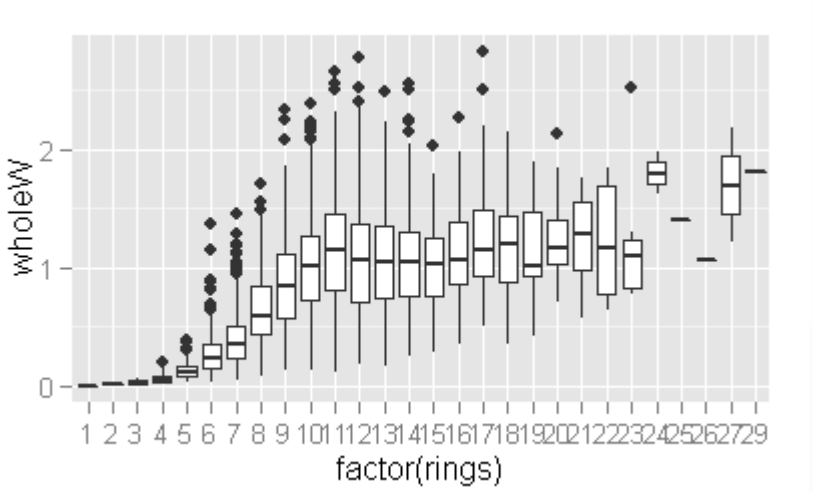
```
> qplot(sex, wholeW, data = abalone, geom="boxplot")
```



상자그림 (2)

- 범주형 변수 : factor로 지정되어 있는 변수

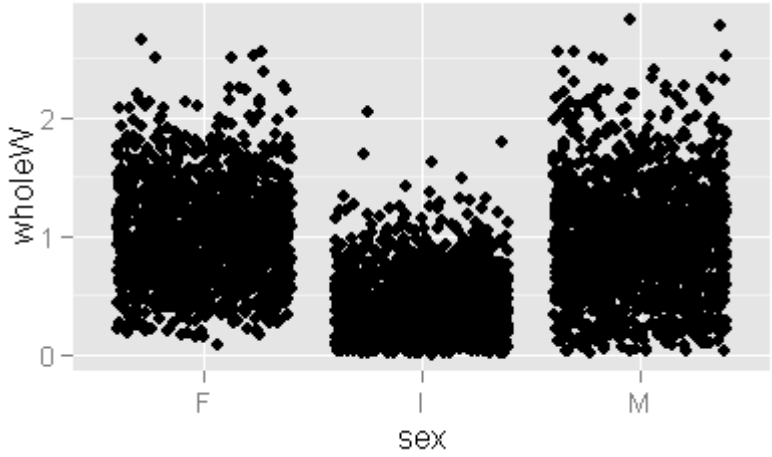
```
> qplot(factor(rings), wholeW, data = abalone, geom="boxplot")
```



흩뜨림(jittering) (1)

- 흩뜨림은 산점도에서 점이 겹쳐지는 문제를 해결하기 위하여 자료에 불규칙성 잡음을 더해서 그림을 그리는 것
- geom = “jitter” 옵션을 이용

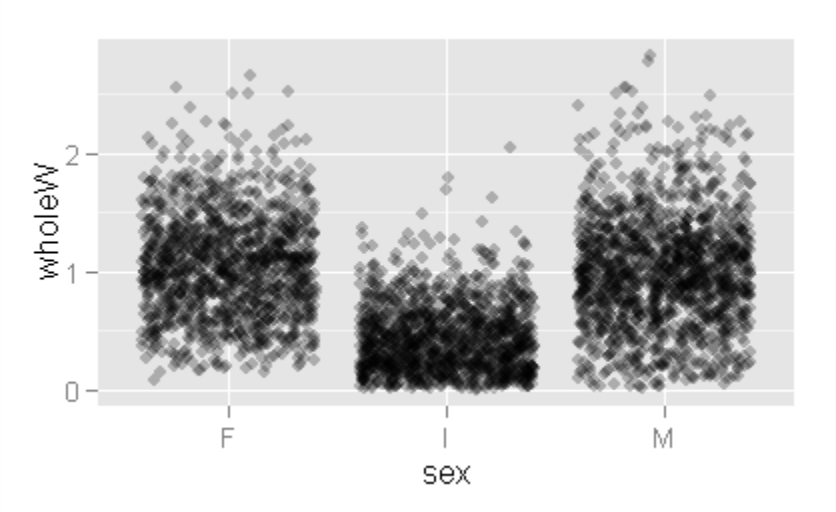
```
> qplot(sex ,wholeW, data = abalone, geom="jitter")
```



흩뜨림(jittering) (2)

- 흩뜨림을 alpha 옵션과 함께 이용하면 좀 더 정확한 분포를 파악할 수 있음

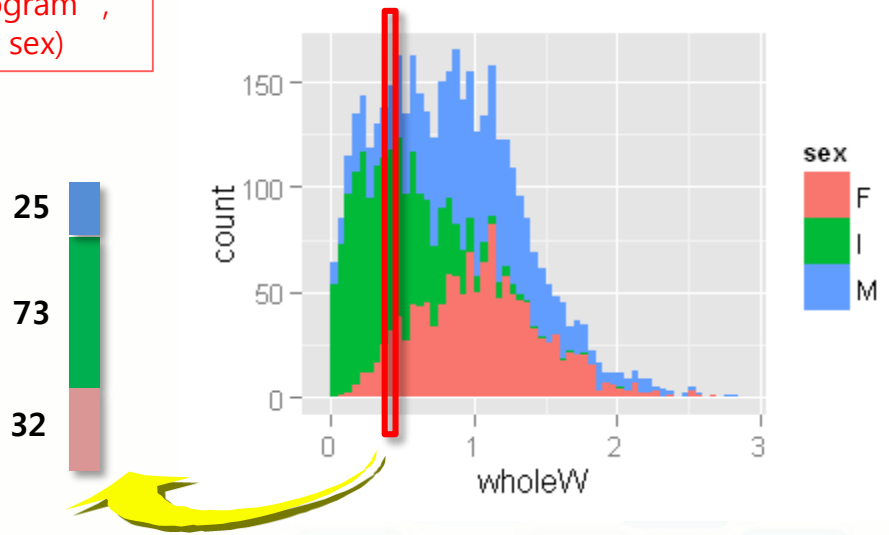
```
> qplot(sex,wholeW,data = abalone,geom="jitter",alpha=I(1/3))
```



히스토그램 (1)

- geom = “histogram” 옵션을 이용
- fill = 범주형변수 옵션을 이용하여 하나의 히스토그램 내에서 여러 범주를 비교할 수 있으나 해석에 유의해야 함

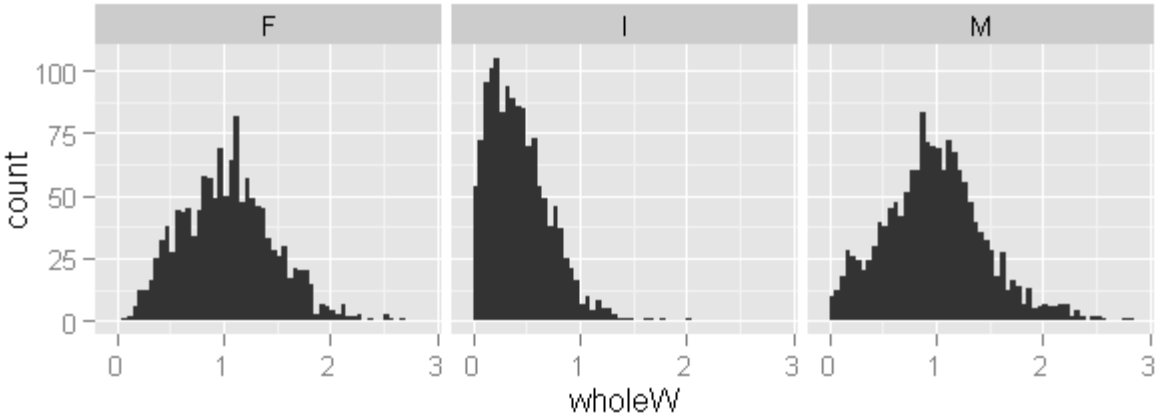
```
> qplot(wholeW, data = abalone, geom = "histogram" ,
+                              binwidth = 0.05, fill = sex)
```



히스토그램 (2)

- facets = .~B의 옵션을 이용하여 범주별 히스토그램을 그릴 수 있음
- B 변수의 범주를 열로 하는 행렬 형태로 히스토그램을 나열

```
> qplot(wholeW, data = abalone, geom="histogram", binwidth=0.05 , facets = .~sex)
```



날짜, 시간 변환

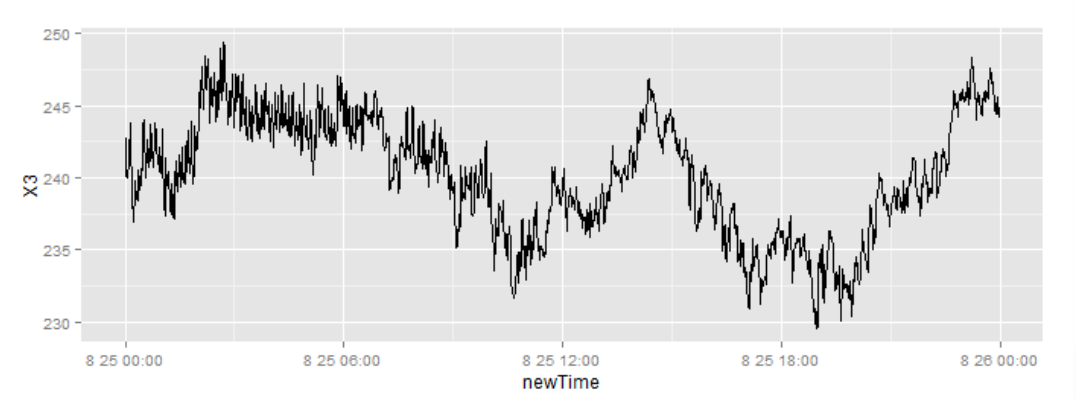
- as.POSIXlt함수를 이용하여 factor로 저장된 Date와 Time 변수를 날짜와 시간으로 변환
- %d (날짜, 01~31), %m (달, 01~12), %Y (4자리로 표시된 연도)
- %H (시간, 00~23), %M (분, 00~59), %S (초, 00~59)

```
> class(Pconsump$Date) # 17/12/2006
[1] "factor"
> class(Pconsump$Time) # 0:00:00
[1] "factor"
> Pconsump$newDate <- as.POSIXlt(Pconsump$Date, format="%d/%m/%Y")
> Pconsump$newTime <- as.POSIXlt(Pconsump$Time,format="%H:%M:%S")
> Pconsump$year <-  format(Pconsump$newDate,"%Y")
```

시계열 그림 (1)

- geom = “line”을 이용

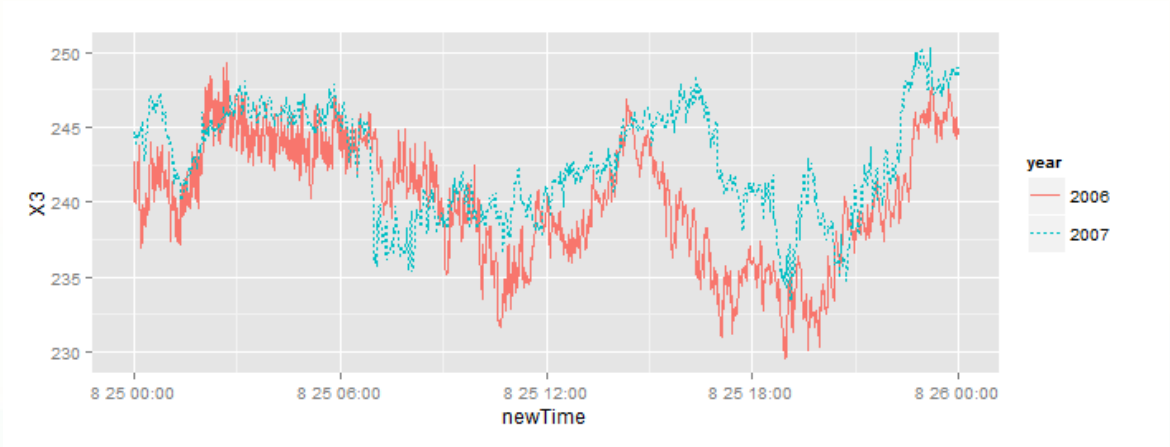
```
> Pconsump.2006.12.17 <- Pconsump[Pconsump$Date == "17/12/2006",]
> qplot(newTime, X3, data = Pconsump.2006.12.17, geom = "line")
```



시계열 그림 (2)

- color와 linetype 옵션을 이용하여 범주별로 다른 선으로 표시

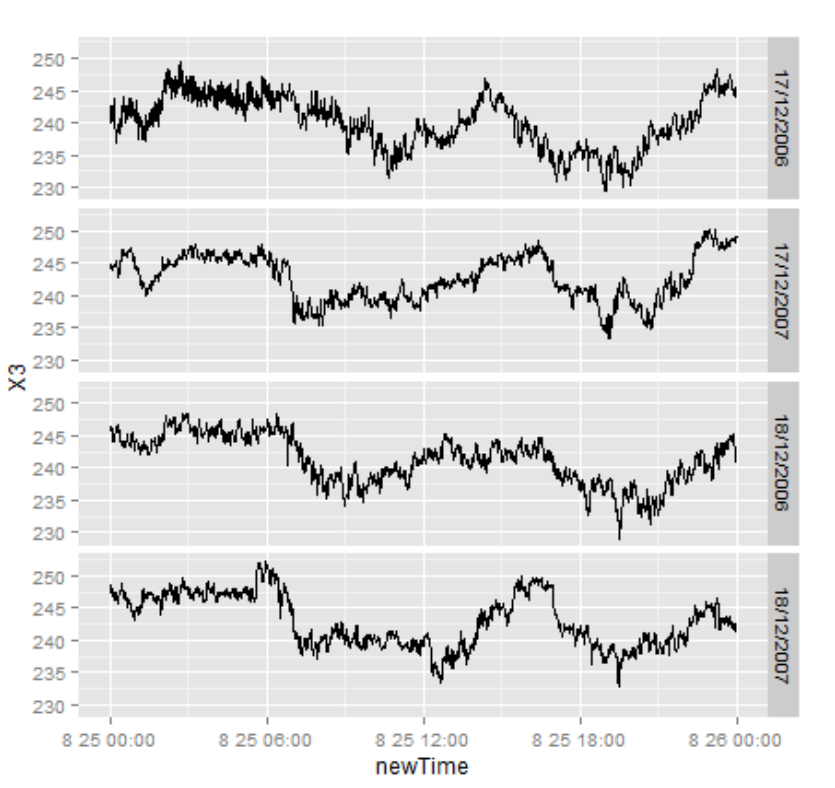
```
> Pconsump.12.17 <- Pconsump[(Pconsump$Date == "17/12/2006" |
+                             Pconsump$Date == "17/12/2007"), ]
> qplot(newTime, X3, data = Pconsump.12.17, color = year,
+                    geom = "line", linetype = year)
```



시계열 그림 (3)

- facets = A~. 를 이용하여 A의 범주를 행으로 하는 행렬형태로 시계열 그림을 나타냄

```
> qplot(newTime,X3,data = Pconsump,
+     facets=Date~.,geom="line")
```



4 ggplot을 이용한 층화 그래프

ggplot – 그래프 정의와 aesthetic mapping

- 자료와 그림의 기본사항들을 이용하여 그래프를 정의
- ggplot에서는 data frame으로 정의된 자료를 이용
- aes( )를 이용하여 그림의 기본사항들을 정의
  - x = x축 변수
  - y = y축 변수
  - color = 색을 다르게 표현할 그룹변수
  - shape = 모양을 다르게 표현할 그룹변수
  - size = 크기를 나타내는 연속변수

```
> plot.basic<-ggplot(tipping,
+     aes(x=totbill,
+         y=tip,
+         color=sex,
+         shape=sex,
+         size=tiprate))
```

ggplot으로 산점도 그리기

- 층화그래픽 문법은 정의된 그래프에 layer를 이용하는 형태로 이용
- 산점도는 layer(geom = “point”)의 층화를 이용
- 또는 geom_point()를 이용

```
> plot.basic + layer(geom="point")
```

```
> plot.basic + geom_point()
```

geom 종류

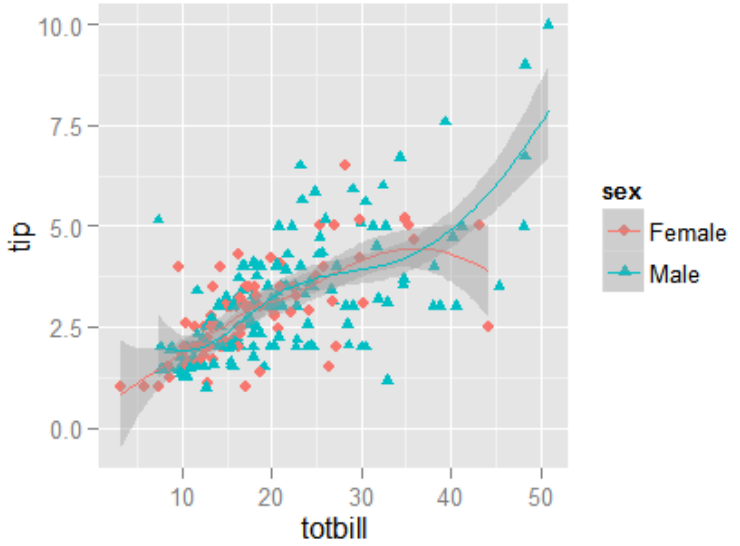
- geom_이름( )의 형태로 이용

abline	절편과 기울기를 이용하여 그리는 직선
area	영역 그림
bar	막대그림 (Y축에 지정된 변수를 이용)
boxplot	상자그림
density	smooth density
histogram	히스토그램
hline/vline	수평직선/수직직선
jitter	점 흐트려 그리기
line	x값 순서로 점 연결하여 그리는 직선
smooth	Smooth line

다양한 geom을 이용한 그림그리기

- 여러 geom을 하나의 그림에 그릴 수 있으며 geom을 정의할 때 마다 필요한 aesthetic mapping을 추가로 지정

```
> plot.basic+geom_point()+
+              geom_smooth(aes(group=sex))
```



stat 종류

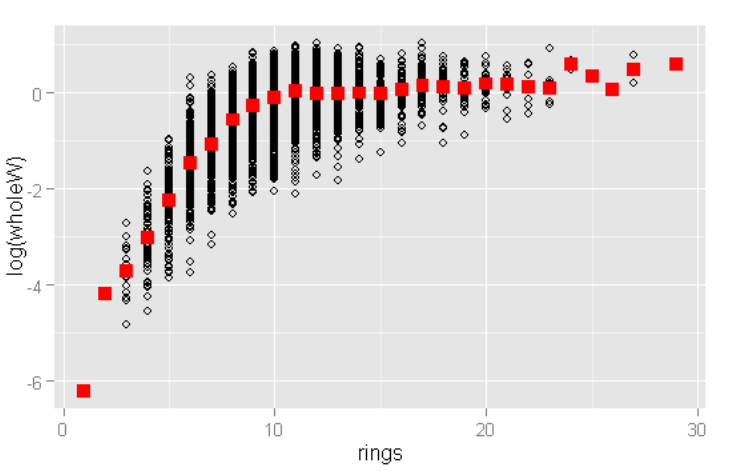
- stat_이름( )의 형태로 이용

bin	범주화 자료로 만들기
boxplot	상자그림을 위한 통계량 계산
contour	3차원 등고선도를 위한 자료 생성
density	1차원 밀도함수 계산
density_2d	2차원 밀도함수 계산
identity	통계량을 계산하지 않고 원자료를 그대로 이용
qq	QQ-plot을 위한 통계량 계산
quantile	분위수 계산
step	계단 함수 형태의 그림을 위한 값 계산
summary	같은 x값에 대하여 y의 통계량을 계산

다양한 stat을 이용한 그래프 (1)

- 산점도에 통계량을 이용한 점 찍기

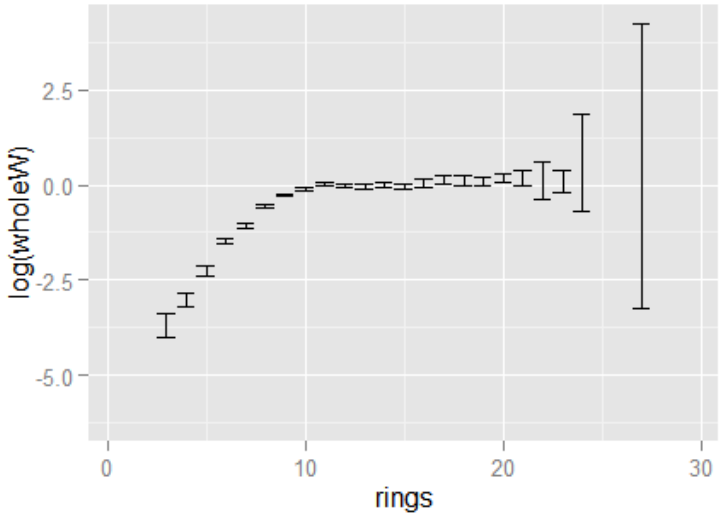
```
> plot.stat<-ggplot(abalone,aes(x=rings,y=log(wholeW)))
> plot.stat + geom_point(shape=1) +
+             stat_summary(size=3,shape=15,color="red",
+                          fun.y = "mean", geom = "point")
```



다양한 stat을 이용한 그래프 (2)

- fun.data = “mean_cl_normal” 옵션을 이용하여 x의 각 값에 대하여 y의 신뢰구간을 표현

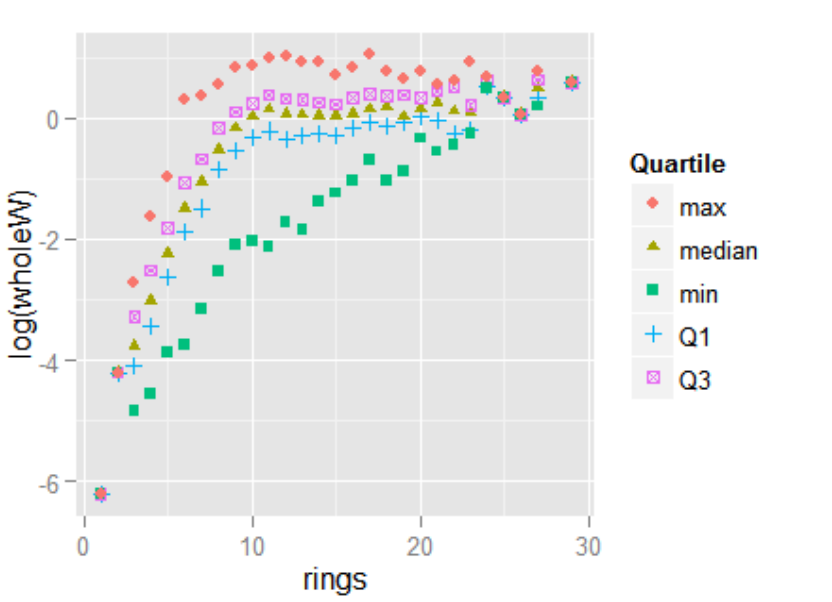
```
> plot.stat + stat_summary(fun.data = "mean_cl_normal", geom = "errorbar")
```



다양한 stat을 이용한 그래프 (3)

- 각 x값에 대한 최소값, Q1, 중앙값, Q3, 최대값을 나타낸 그림

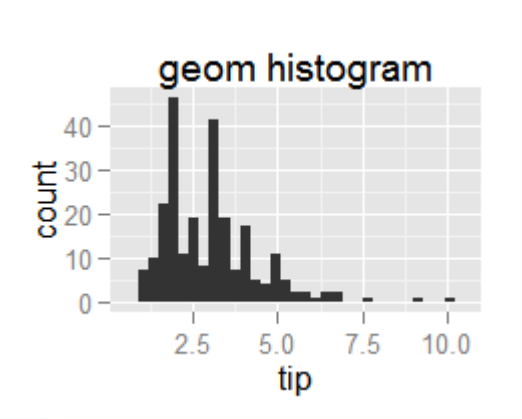
```
> q1<-function(x) quantile(x,p=0.25)
> q3<-function(x) quantile(x,p=0.75)
> plot.stat + stat_summary(aes(color="Q1",shape="Q1"),
+                                    fun.y=q1,geom="point") +
+   stat_summary(aes(color="median",shape="median"),
+                fun.y=median,geom="point")
+   stat_summary(aes(color="Q3",shape="Q3"),
+                fun.y=q3,geom="point") +
+   stat_summary(aes(color="min",shape="min"),
+                fun.y=min,geom="point") +
+   stat_summary(aes(color="max",shape="max"),
+                fun.y=max,geom="point") +
+   scale_color_hue("Quartile")+scale_shape("Quartile")
```

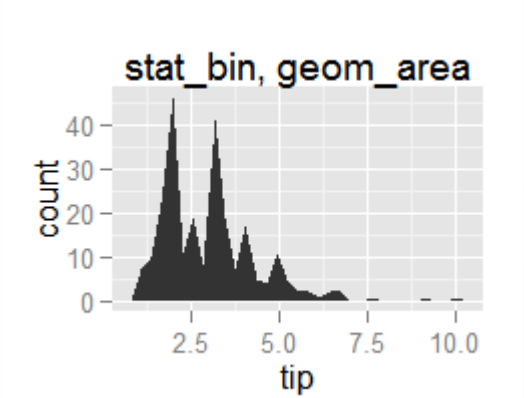


히스토그램의 변형 (1)

- stat_bin과 geom을 결합하여 히스토그램을 다양한 형태로 변형할 수 있음

```
> plot.1D <- ggplot(tipping, aes(x = tip))
> plot.1D + geom_histogram() + ggtitle("geom histogram")
> plot.1D + stat_bin(geom = "area") + ggtitle("stat_bin, geom_area")
```

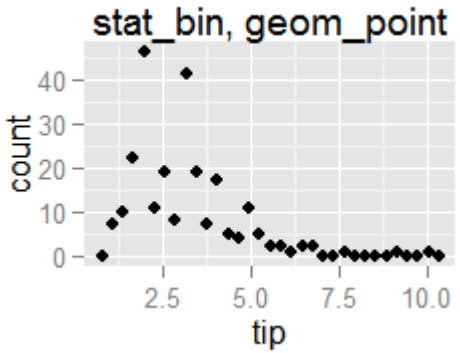


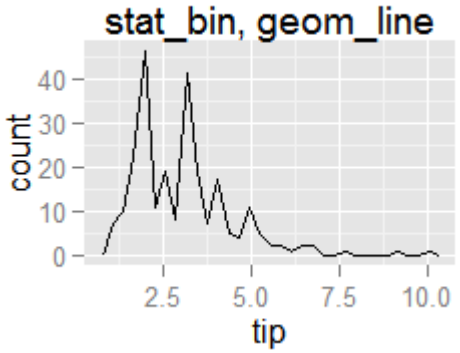


히스토그램의 변형 (2)

- geom = “point”와 geom=“line”을 이용

```
> plot.1D + stat_bin(geom = "point") + ggtitle("stat_bin, geom_point")
> plot.1D + stat_bin(geom = "line") + ggtitle("stat_bin, geom_line")
```





Position adjustment (1)

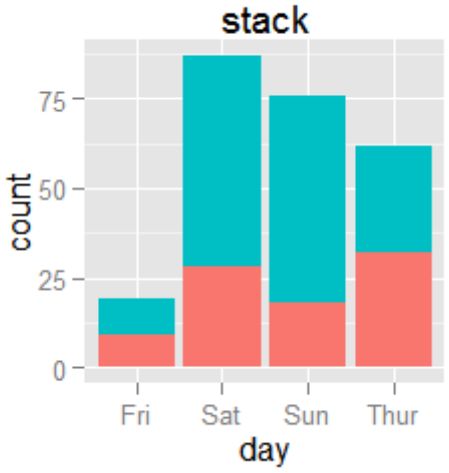
- 그림에서 겹쳐서 나타나는 부분을 처리하는 방법을 지정

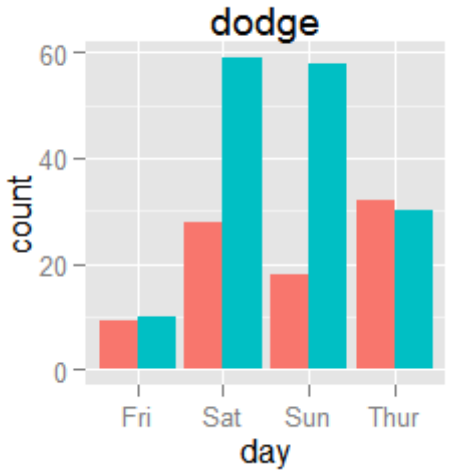
dodge	겹쳐지는 부분을 옆으로 나란히 정렬하는 방법
fill	전체영역을 칠하기
identity	그대로 나타내기
jitter	겹쳐진 점을 흐트려서 나타내기
stack	겹쳐진 부분을 위에 쌓는 형태로 그리기

Position adjustment (2)

- position = “stack”
- position = “dodge”
- theme(legend.position =“none”) : 그림에서 범례를 제외시키기 위한 옵션

```
> plot.pos <- ggplot(tipping,aes(x = day, fill = sex, shape = sex))
> plot.pos + geom_bar(position = "stack") + ggtitle("stack") +
+             theme(legend.position="none")
> plot.pos + geom_bar(position = "dodge") + ggtitle("dodge") +
+             theme(legend.position="none")
```

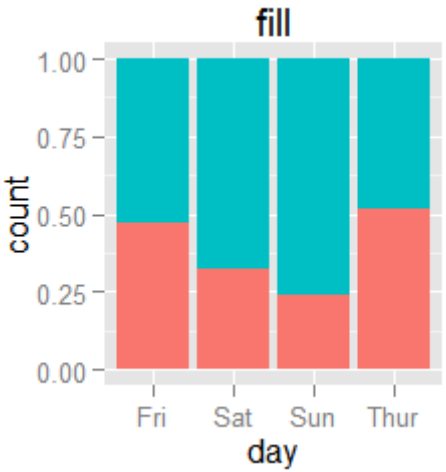


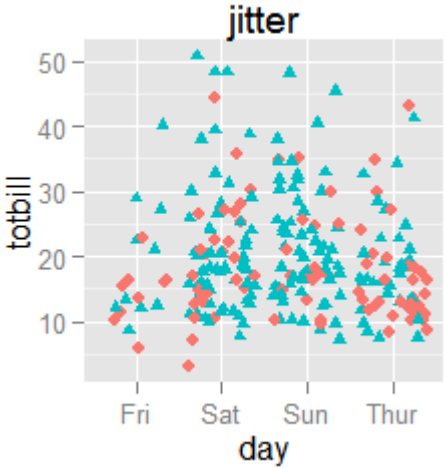


Position adjustment (3)

- position = “fill”
- position = “jitter”

```
> plot.pos + geom_bar(position = "fill") + ggtitle("fill") +
+              theme(legend.position="none")
> plot.pos +
geom_point(aes(y=totbill,color=sex,shape=sex),position="jitter") +
+              ggtitle("jitter") + theme(legend.position="none")
```

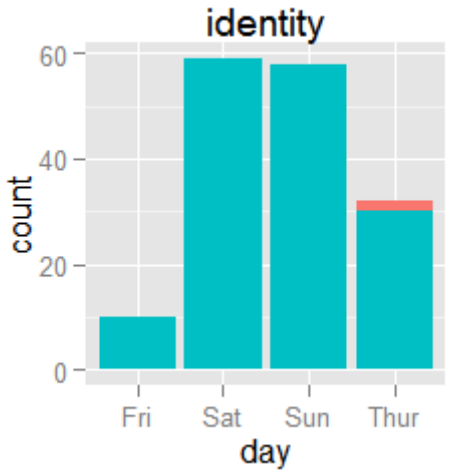


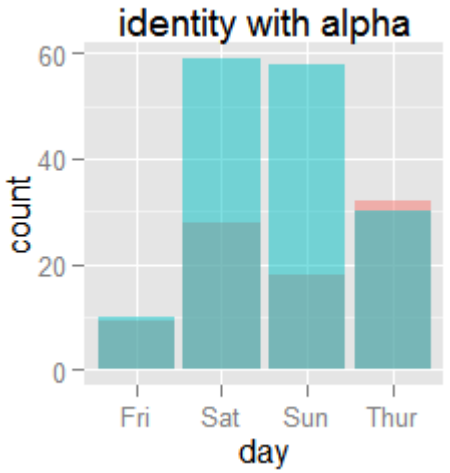


Position adjustment (4)

- position = “identity”
- position = “identity” 와 alpha 옵션

```
> plot.pos <- ggplot(tipping,aes(x = day, fill = sex, shape = sex))
> plot.pos + geom_bar(position = "identity") + ggtitle("identity") +
+           theme(legend.position="none")
> plot.pos + geom_bar(position = "identity", alpha = I(0.5)) +
+           ggtitle("identity with alpha") +
+           theme(legend.position="none")
```

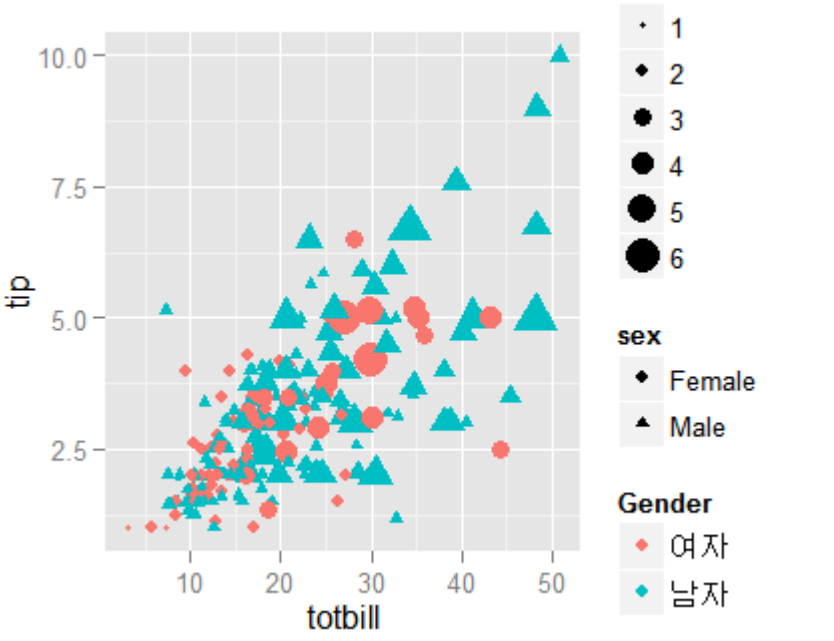




Scale (1)

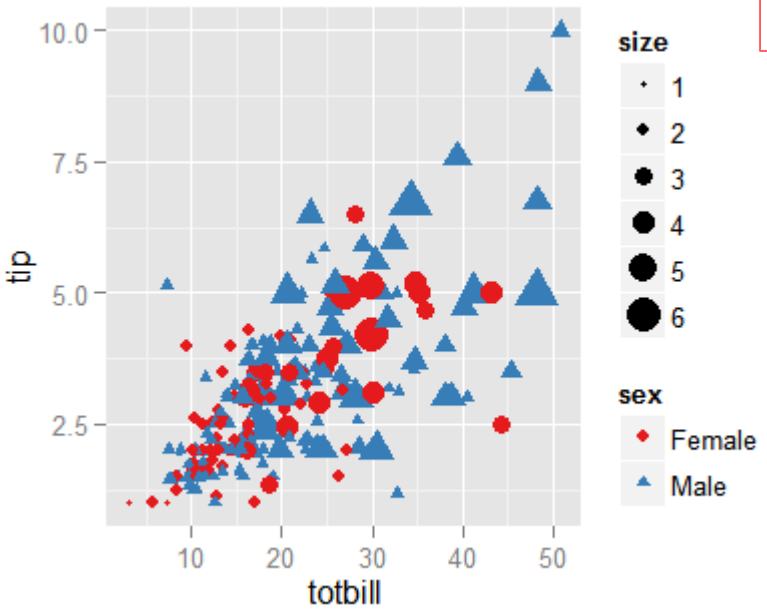
- 점의 모양(shape), 크기(size), 색(color) 등을 지정하는 것을 의미함
- scale_color_hue : 색에 대한 범례의 제목과 범주의 이름을 지정하는 함수

```
> plot.scale1 <- ggplot(tipping, aes(x=totbill, y = tip,
+                                    color = sex, shape = sex,
+                                    size = size)) +
+                       geom_point()
> plot.scale1 +
+   scale_color_hue("Gender", labels = c("여자", "남자"))
```



Scale (2)

- Scale_color_brewer : 범주에 따라 자동으로 지정되는 색을 바꾸기 위한 함수

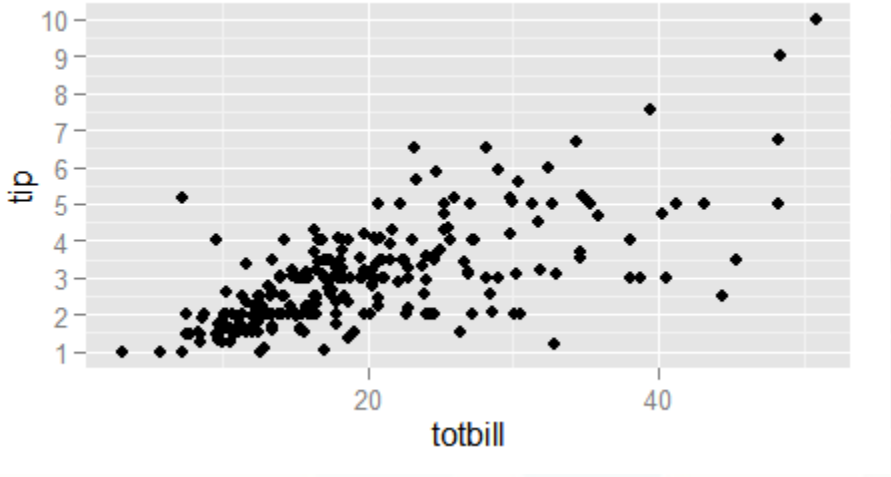



```
> plot.scale1 +
+       scale_color_brewer(palette="Set1")
```

Scale (3)

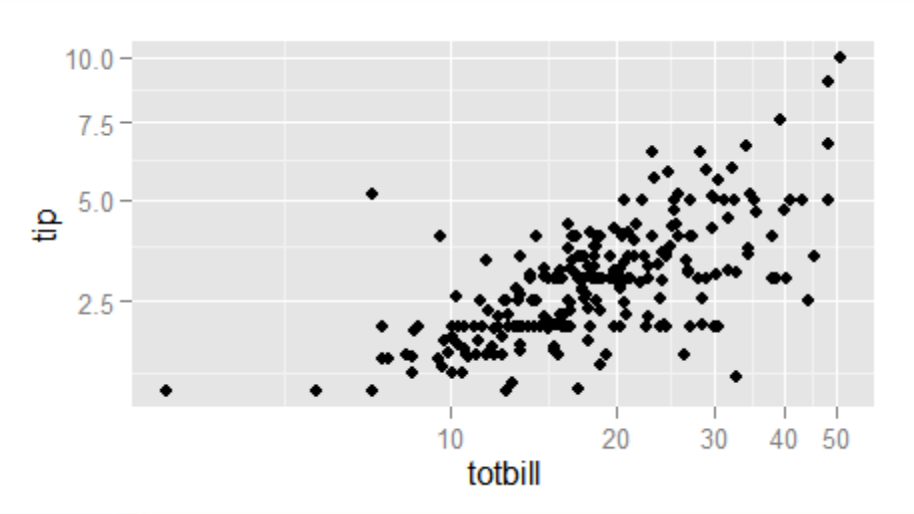
- scale_x_continuous : x축에 대한 옵션을 제시하는 함수
  - breaks = c(20,40) : 20과 40에 눈금을 표시
- scale_y_continuous : y축에 대한 옵션을 제시하는 함수

```
> plot.scale2 <-
+       ggplot(tipping,aes(x=totbill,y=tip))+
+       geom_point()
> plot.scale2 +
+     scale_x_continuous(breaks=c(20,40)) +
+     scale_y_continuous(breaks=1:10)
```



Scale (4)

- coord_trans : 각 축을 함수를 이용하여 변환할 수 있도록 하는 함수
  - xtrans = “log10” : log10 함수를 이용하여 x축을 변환
  - ytrans = “log10” : log10 함수를 이용하여 y축을 변환

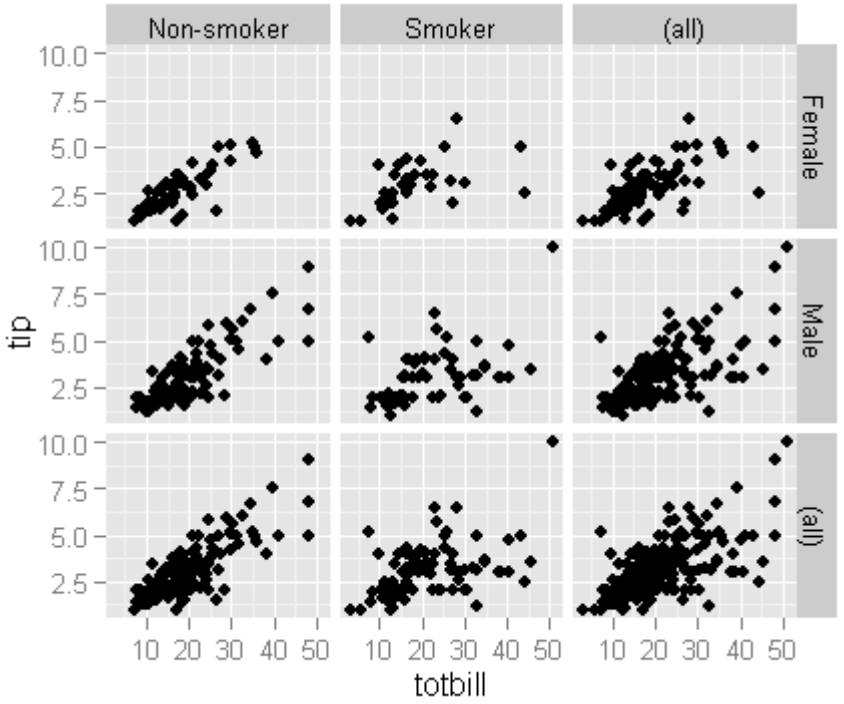



```
> plot.scale2 +
+     coord_trans(xtrans="log10",ytrans="sqrt")
```

Faceting (1)

- facet_grid(A~B) : 변수 A의 범주를 행으로, 변수 B의 범주를 열로 하여 행렬형태로 그림을 그리는 방법
- margins=TRUE : 각 행, 열의 마지막에 주변분포의 그림을 그려줌

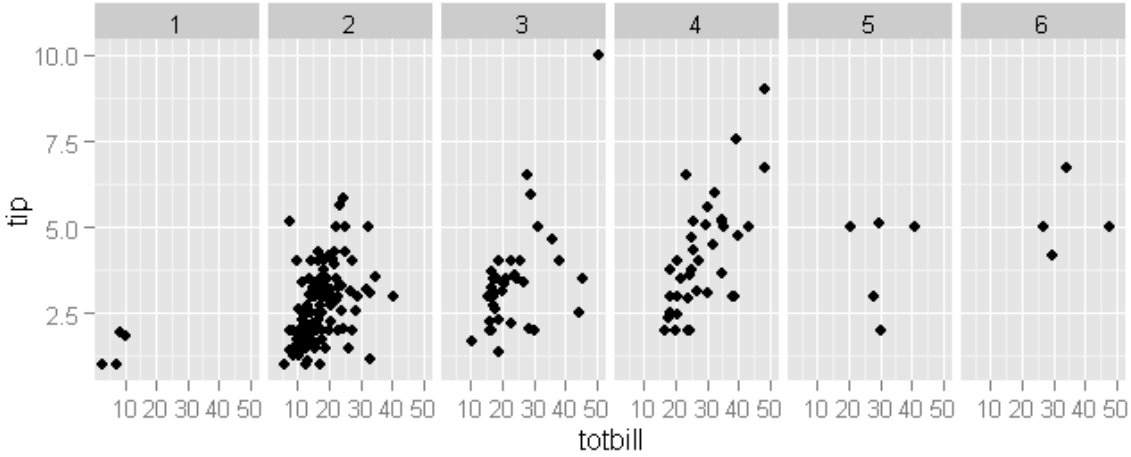
```
> plot.facet <- ggplot(tipping,
+                      aes(x = totbill, y = tip)) +
+                      geom_point()
> plot.facet +
+          facet_grid(sex ~ smoker, margins = TRUE)
```



Faceting (2)

- facet_wrap(~A) : 하나의 범주형 변수 A를 이용하는 방법으로 ncol, nrow 옵션을 이용하여 그림 행렬의 모양을 지정

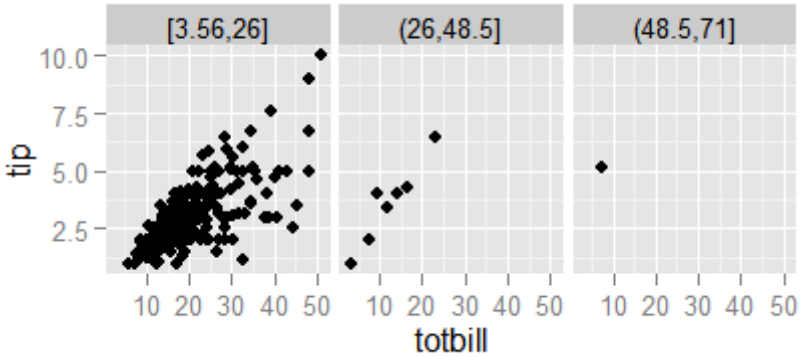
```
> plot.facet + facet_wrap(~ size, ncol = 6)
```



Faceting (3)

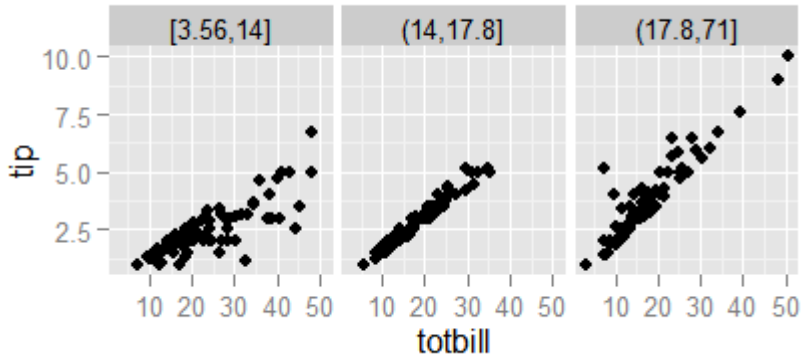
- cut_interval, cut_number 함수를 이용하여 연속변수를 범주화
  - cut_interval : 자료 값의 범위를 일정하게 하여 n개의 그룹으로 나누어 주는 함수
  - cut_number : 각 그룹의 자료수가 같아지도록 n개의 그룹으로 나누어주는 함수

```
> tipping$tipgroup1 <- cut_interval(tipping$tiprate,n=3)
> tipping$tipgroup2 <- cut_number(tipping$tiprate,n=3)
> plot.newfacet<-ggplot(tipping,aes(x=totbill,y=tip))+
+                       geom_point()
> plot.newfacet + facet_wrap(~tipgroup1)
```



Faceting (4)

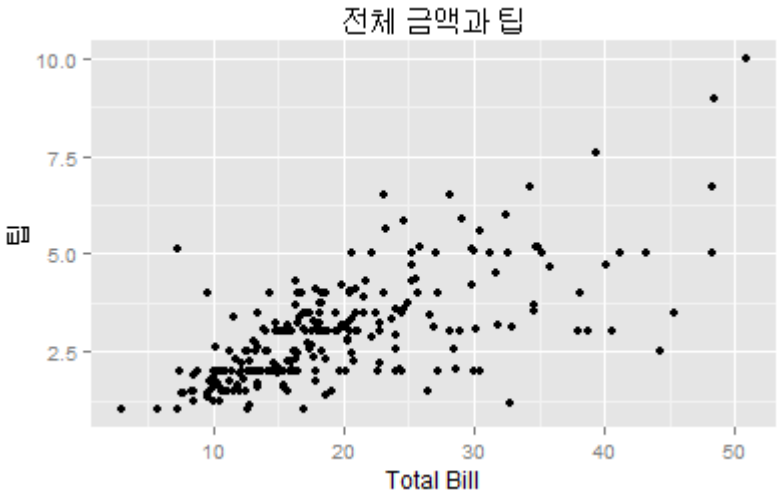
```
> plot.newfacet + facet_wrap(~tipgroup2)
```



Theme (1)

- 보고서 작성을 위해 그림을 다듬는 데에 필요한 옵션들
  - xlab/ylab : x축/y축 이름을 지정
  - ggtitle : 그림의 제목을 지정

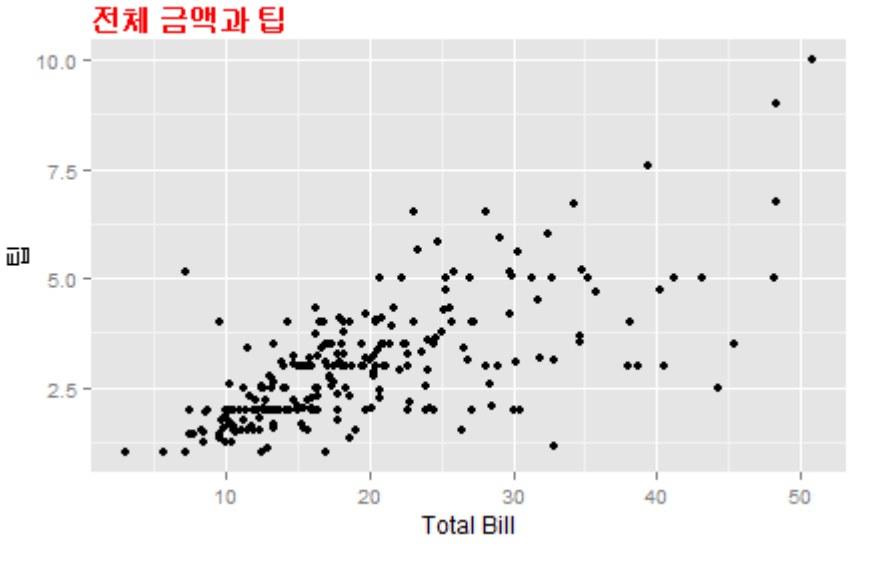
```
> plot.theme <- ggplot(tipping, aes(x = totbill, y = tip))
> plot.theme + geom_point() + xlab("Total Bill") +
+              ylab("팁") + ggtitle("전체 금액과 팁")
```



Theme (2)

- Theme 함수를 이용하여 그림의 전반적인 사항들 지정
  - plot.title : 그림 제목에 관한 옵션 지정

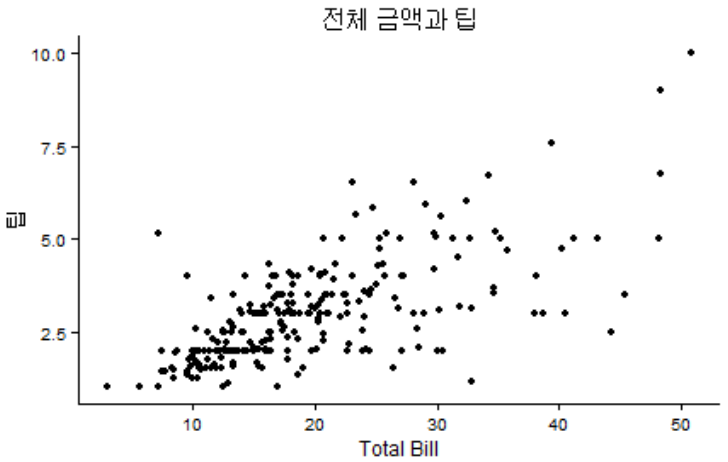
```
> plot.theme + geom_point() + xlab("Total Bill") +
+              ylab("팁") + ggtitle("전체 금액과 팁") +
+       theme(plot. title = element_text(color = "red",
+                                face = "bold", hjust = 0))
```



Theme (3)

- last_plot : 마지막으로 그린 그림을 불러오는 함수
- theme_bw : 배경색을 흰색으로 변경하는 함수
- panel.grid.minor/panel.grid.major : 격자선에 관한 옵션
- panel.border : 그림 테두리에 관한 옵션
- axis.line : x, y축의 선에 관한 옵션

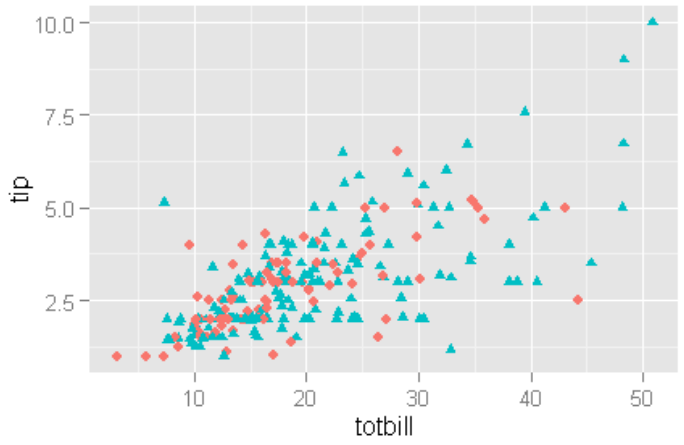
```
> last_plot()+theme_bw()+
+       theme(panel.grid.major=element_blank(),
+             panel.grid.minor=element_blank(),
+             panel.border=element_blank(),
+             axis.line = element_line())
```



Theme (4)

- legend.position = “none” : 범례를 생략하기 위한 옵션

```
> plot.theme +
+       geom_point(aes(color = sex, shape = sex)) +
+       theme(legend.position = "none")
```



Theme (5)

- geom_text : 그림에 문자를 삽입하기 위한 함수
- expression : 수식을 표현하기 위한 함수

```
> lm.result<-lm(tip~totbill,data = tipping)
> ab <- round(coef(lm.result),2)
> ggplot(tipping,aes(x=totbill,y=tip))+
+       geom_point()+
+       geom_smooth(method="lm")+
+       geom_text(data = NULL,x=10,y=8.5,
+           label=paste("y =",ab[1],"+",ab[2],"x"),hjust=0) +
+       ggtitle(expression(paste(hat(beta)[0],"+",hat(beta)[1],"x")))
```

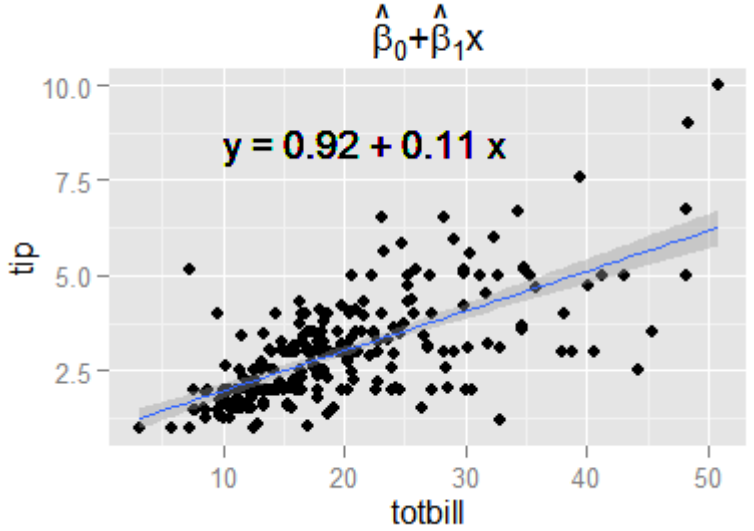


그림 저장

- ggsave
  - R 세션에서 활성화 되어 있는 R Graphics Device에 그려져 있는 그림을 저장해주는 함수
  - 파일 이름의 확장자에 맞는 형태의 그림으로 저장
  - MS 오피스나 아래한글, 혹은 웹 문서를 위해서는 png 파일을 이용

```
> ggsave("sample-plot.png")
```

다음시간 안내

고급 그래픽기법(1)