

# INTRODUCTORY GGPLOT2

## Visualization Guide

#SKKU #P-SAT #TEAM6\_PRESENTS #2017

Our company



**SAMANTHA BLACK**  
sales director

### EXPERIENCE

**POSITION TITLE** for company ltd  
Present  
Short description of the position and the responsibilities you had in this position.

**POSITION TITLE** for company ltd  
2013 - 2016  
Short description of the position and the responsibilities you had in this position.

**ADDRESS**  
125 Name Street,  
Town / City,  
State / Country,  
Postal / ZIP code

**HOBBIES**  
creating websites  
swimming  
photography  
body building

**PHONE**  
0028 012 345 678

**EMAIL**  
info@samblaq.com

**WEBSITE**  
www.mypage.com

**SKYPE**  
skype: sambqak



### PROFESSIONAL STATEMENT

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse suscipit efficitur lectus. Fusce iaculis, leo nec vulputate efficitur, lorem interdum elit, ut vestibulum nisl metus non mi.

Aliquam dictum porta erat nec commodo. Maecenas vestibulum massa in justo pellentesque, non eleifend dolor ornare. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse suscipit efficitur lectus. Fusce iaculis, leo nec vulputate efficitur, lorem interdum elit, ut vestibulum nisl metus non mi.

Aliquam dictum porta erat nec commodo. Maecenas vestibulum massa in justo pellentesque, non eleifend dolor ornare. Lorem ipsum dolor sit amet, consectetur nunc.

**HIGH SCHOOL UNIVERSITY**  
2008 - 2014  
Short description of the school and the responsibilities you had in this position. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Inusani qui spe volut new.

**SCHOOL TITLE LOREM**  
2004 - 2008  
Short description of the position and the responsibilities you had in this position.

### SKILLS

PHOTOGRAPHY

PHOTOSHOP

INDESIGN

WORDPRESS

TIME KE

ORGA



# Contents

1. GRAMMAR OF GGPLOT2	3
2. VARIOUS GEOMETRY	13
3. MANUAL OPTIONS	31
+. ADDITIONAL NOTES	75





# 1

## GRAMMAR OF GGPLOT2





## GGplot

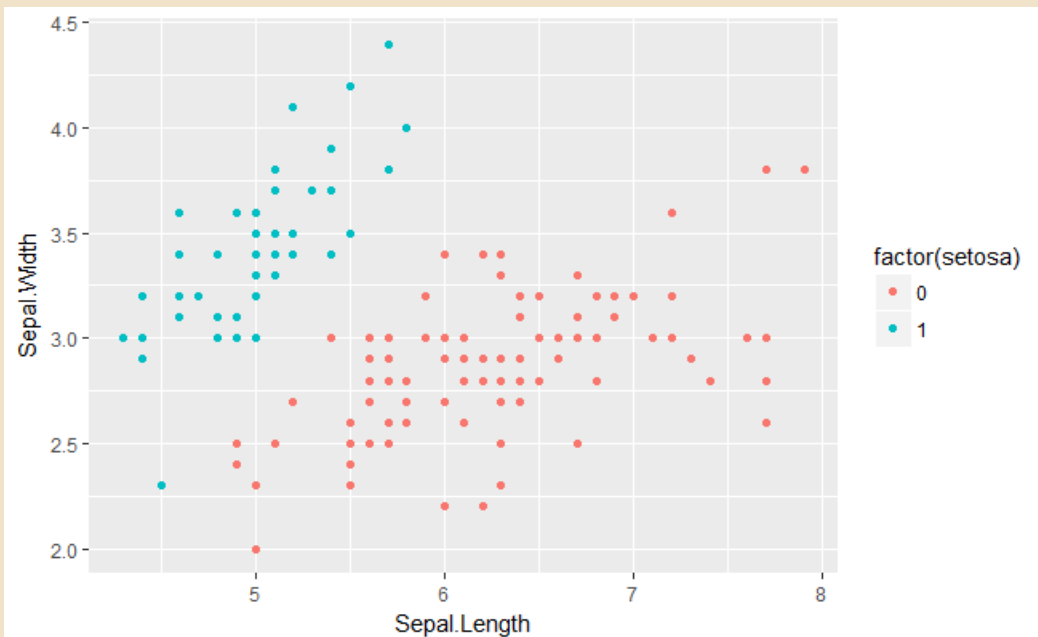
: **G**rammar of **G**raphics **p**lot

: 그래프를 구성하는 요소를 문법처럼 체계화한 시각화 툴

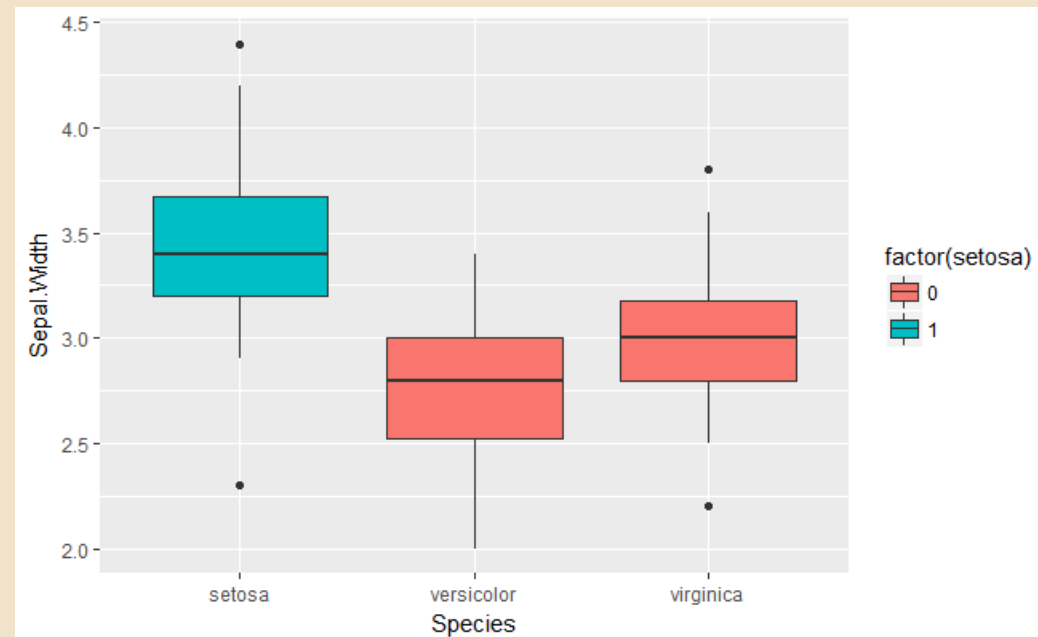
→ 문법과 단어만 알면 쉽고 빠르게 시각화 능력을 배양할 수 있음

## 1

# Why ggplot?



```
> ggplot(iris, aes(x=Sepal.Length, y=Sepal.Width, color=factor(setosa))) +  
+   geom_point()
```



```
> ggplot(iris, aes(x=Species, y=Sepal.Width, fill=factor(setosa))) +  
+   geom_boxplot()
```

시각화 하고자 하는 데이터

ggplot

+

시각화 하고자 하는 방법

geometry

일관된 문법적 구성으로 plot을 그리는 시각화 방법 → 문법과 단어만 알면 쉽게 시각화 가능!

1

Simple sentence we'll pass on to R

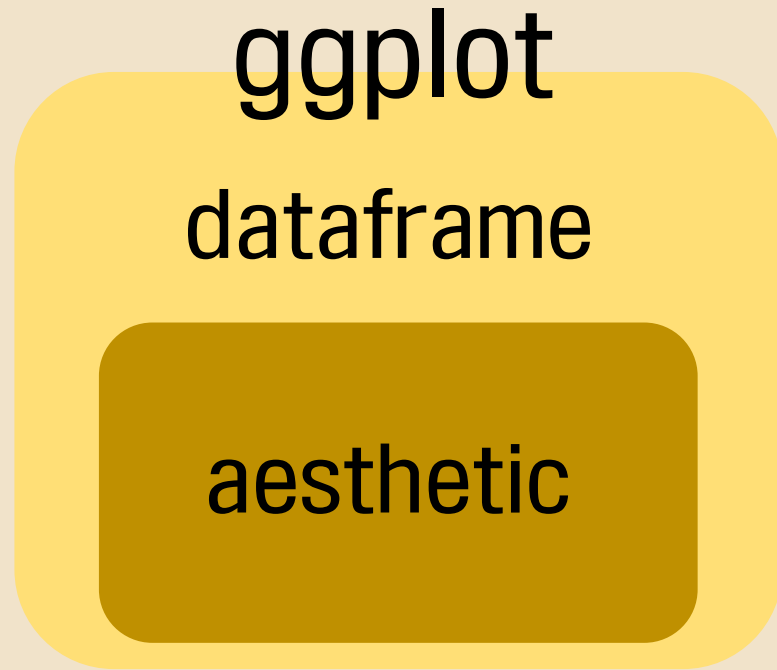
“

이 ggplot을 이 geometry로  
(이 옵션을 적용해서) 시각화해라.

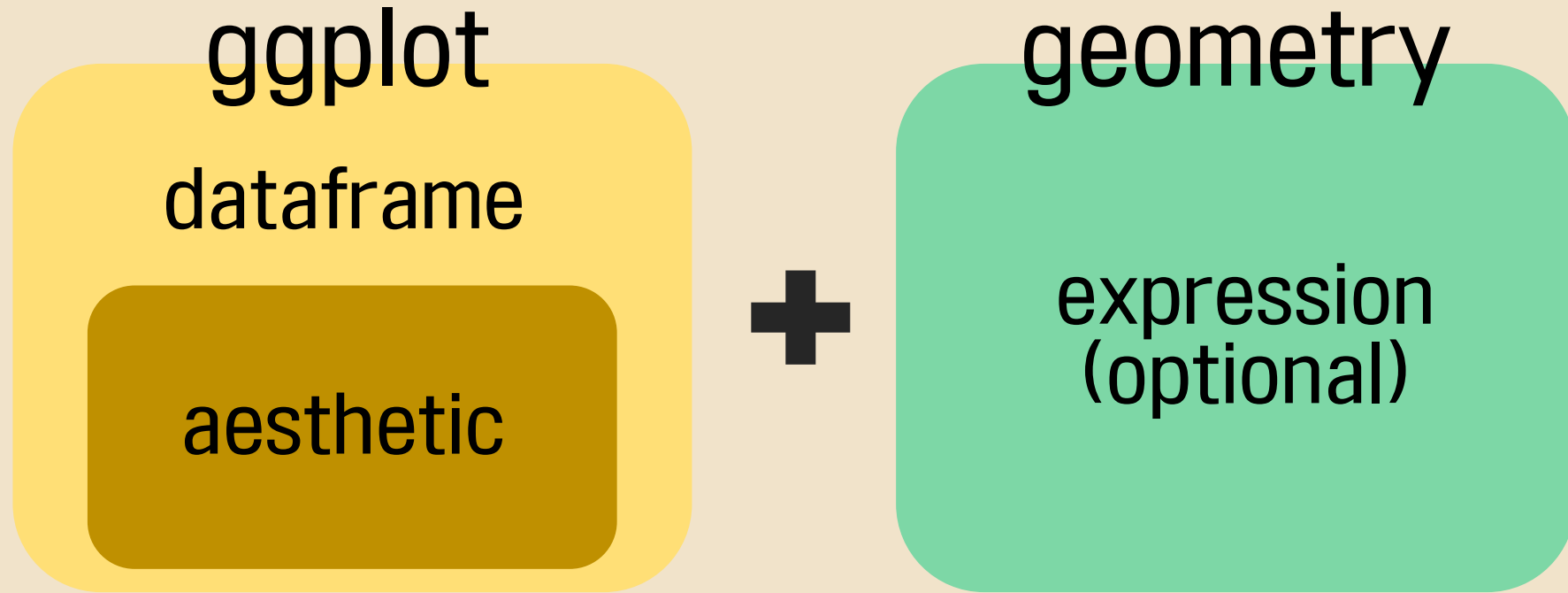
”

1

# Concept: ggplot2



시각화할 dataframe과, 시각화에 사용할 column과 그 역할을  
ggplot()안에 정의한다

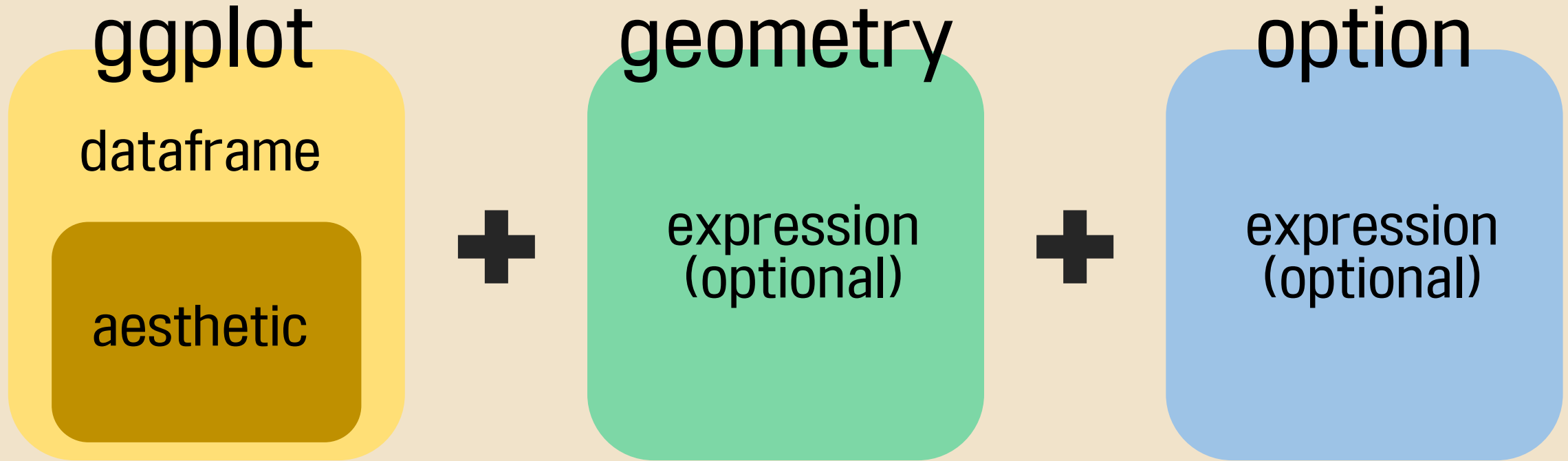


시각화가 어떤 형태로 이루어져야 할지를 특정해서  
Data Visualization을 완성한다.



1

# Concept: ggplot2



각주 추가, 그래프의 색 변경, 축의 외형 변경 등  
세부적인 조정을 가하고 싶을 때 상황별로 사용할 수 있는 option

## ggplot object

`ggplot(dataframe, aesthetic)`

└─ 시각화하고자 하는 데이터프레임

**aesthetic**

시각화에 사용하고자 하는 데이터프레임의 column과 그 역할을 지정하는 부분

문법: `ggplot(dataframe, aes(역할1=column1, 역할2=column2, ...))`

---

ex. `ggplot(iris, aes(x=Species, y=Sepal.Length, fill=Species))`  
: iris 데이터 프레임의 Species를 x축으로 사용하고 Sepal.Length를 y축으로 사용하겠음. 또한, Species변수를 기준으로 색칠하겠음.

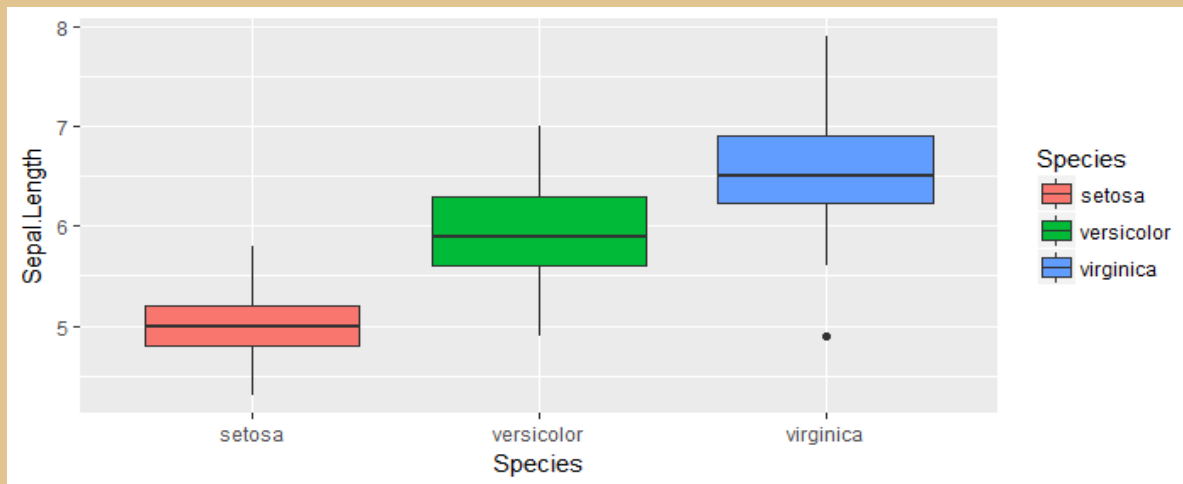


## geometry

그렇게 형성한 ggplot object가 시각화되길 바라는 형태.  
ggplot 다음에 +로 ggplot과 연결

`ggplot(dataframe, aesthetic) + geometry`

```
> ggplot(iris, aes(x=Species, y=Sepal.Length, fill=Species)) + geom_boxplot()
```

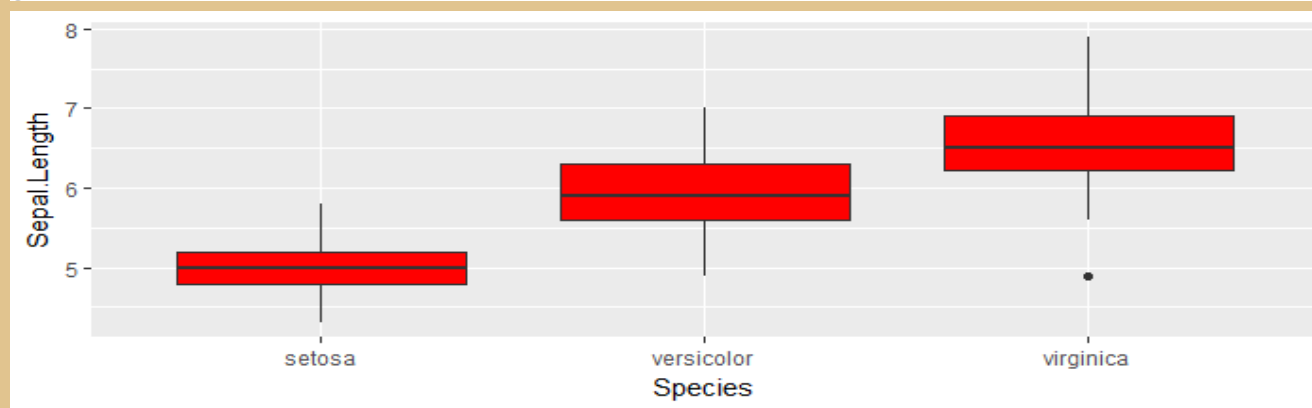


앞선 설정이 boxplot으로  
구현되길 원하므로  
`geom_boxplot()`을 통해  
geometry를 boxplot으로 설정

## Expression inside geometry

ggplot object 내 aesthetic 표현은 범주별로 서식을 달리하는 표현이었지만, geometry 내 표현은 그 geometry 자체의 서식을 조작하는 표현.

```
ggplot(iris,aes(x=Species,y=Sepal.Length,fill=Species)) + geom_boxplot(fill='red')
```



geom\_boxplot(fill='red')를 통해 boxplot을 빨간색으로 칠함  
→ geometry 내의 표현은 ggplot의 aes를 overwrite한다!

\* aesthetic 내에서 사용할 수 있는 다른 다양한 표현들은 부록 참조



# 2

## VARIOUS GEOMETRY





## ◆ Barplot

- [geom\\_bar](#)
- [geom\\_col](#)

## ◆ Line graph

- [geom\\_line](#)

## ◆ Scatterplot

- [geom\\_point](#)

## ◆ Plotting Distribution

- [geom\\_histogram](#)
- [geom\\_density](#)
- [geom\\_boxplot](#)

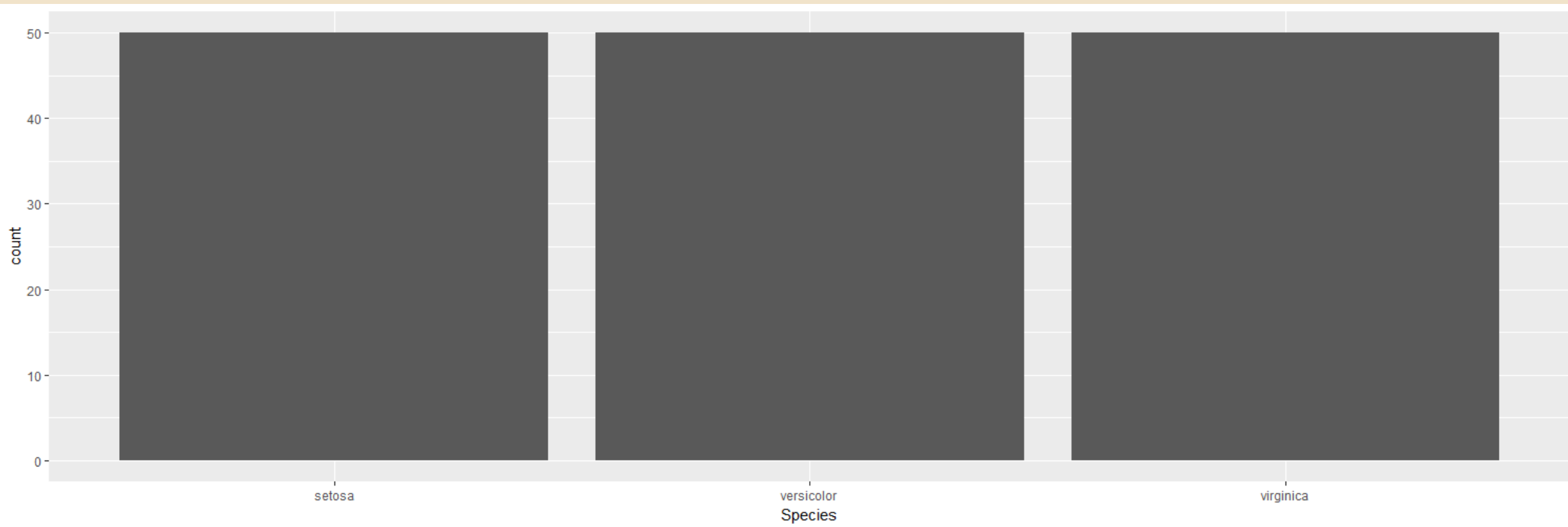


# 2

## Barplot: geom\_bar

CODE

```
ggplot(iris, aes(x=Species)) + geom_bar()
```



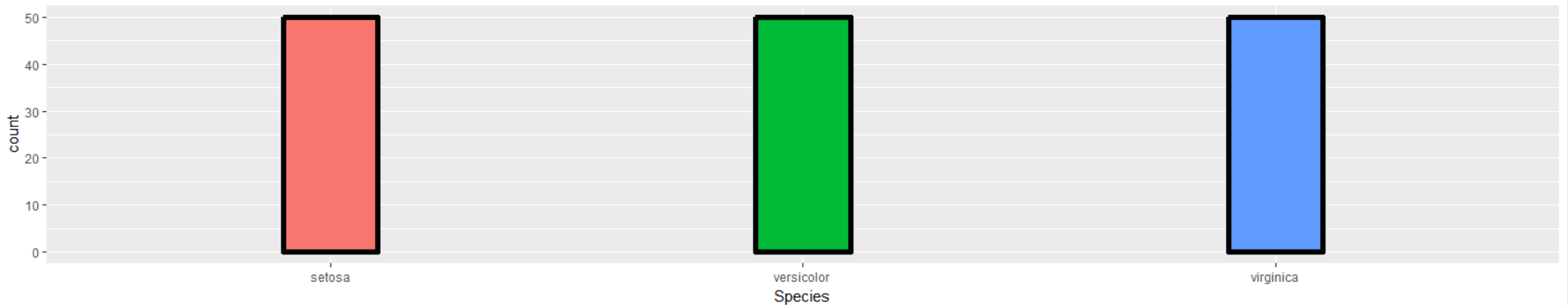
입력받은 범주형 자료 x의 범주별 개수를 세어 나타내고 싶을 때 사용  
범주형 자료 iris\$Species에는 세 범주가 각각 50개씩 있어서 위와 같은 결과 출력

## 2

## example: geom\_bar

CODE

```
ggplot(iris,aes(x=Species,fill=Species)) +  
geom_bar(color='black',size=2,width=0.2,show.legend=FALSE)
```



ggplot : iris의 Species열을 x축에 놓고,  
fill=Species : Species별로 색칠할 것이다.

geom\_bar : 이를 geom\_bar로 구현(범주별 value의 개수를 셈)할 것이고,

color='black' : 테두리 색은 black

size=2 : 테두리 두께는 2

width=0.2 : bar 두께는 0.2

show.legend=FALSE : 범례는 없애라

ggplot + aesthetic

geometry + expression



## 2

## Barplot: geom\_col

CODE

```
ggplot(iris, aes(x=Species,y=Sepal.Width)) + geom_col()
```



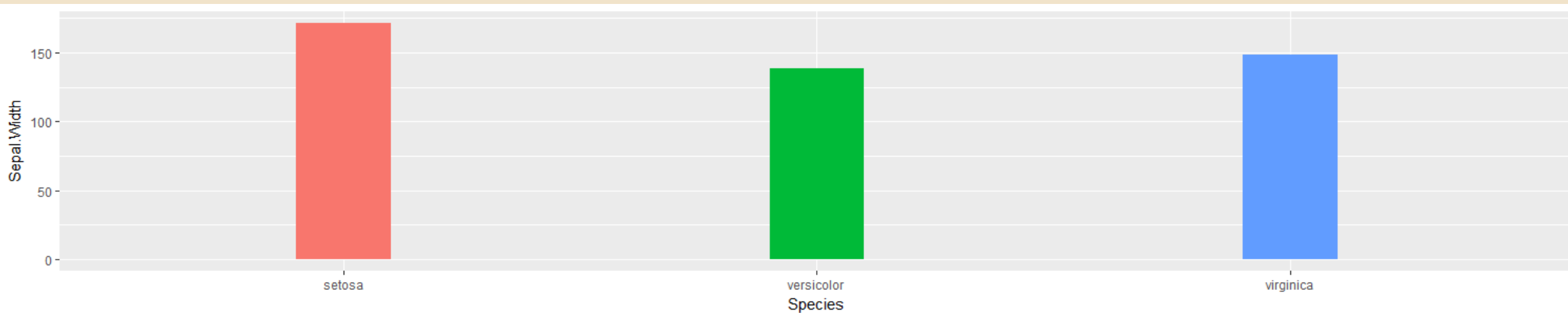
입력받은 범주형 자료 x별 숫자형 자료 y의 합을 나타내고 싶을 때 사용  
Species별로 각각 Sepal.Width의 합이 171.4, 138.5, 148.70이라 위와 같은 결과 출력

## 2

## example: geom\_col

CODE

```
ggplot(iris,aes(x=Species,y=Sepal.Width,fill=Species)) +  
geom_col(width=0.2,show.legend=FALSE)
```



ggplot : Species를 x, Sepal.Width를 y에 놓는다.  
fill=Species : Species별로 색칠할 것이다.

} ggplot + aesthetic

Geom\_bar : 이를 geom\_col로 구현(x별 y를 합할 것)할 것이고,  
width=0.2 : bar 두께는 0.2

show.legend=FALSE : 범례는 없애라

} geometry + expression

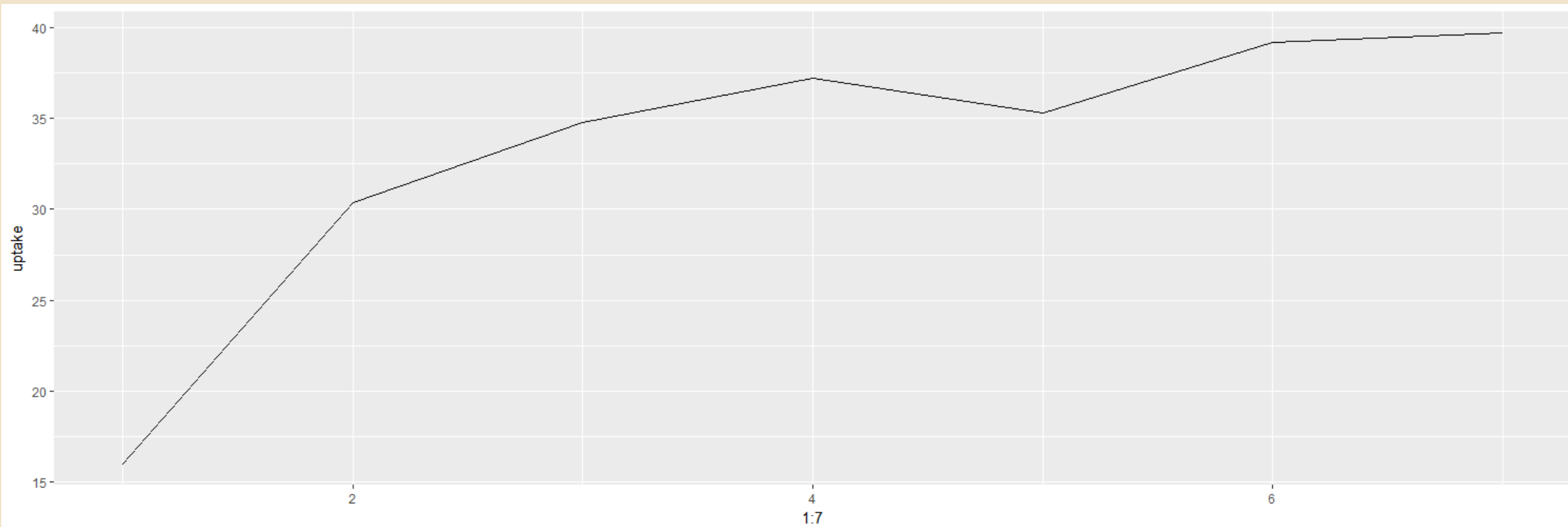


## 2

## Line graph : geom\_line

CODE

```
filter(CO2, Plant == 'Qn1') %>% ggplot(aes(x=1:7, y=uptake)) + geom_line()
```



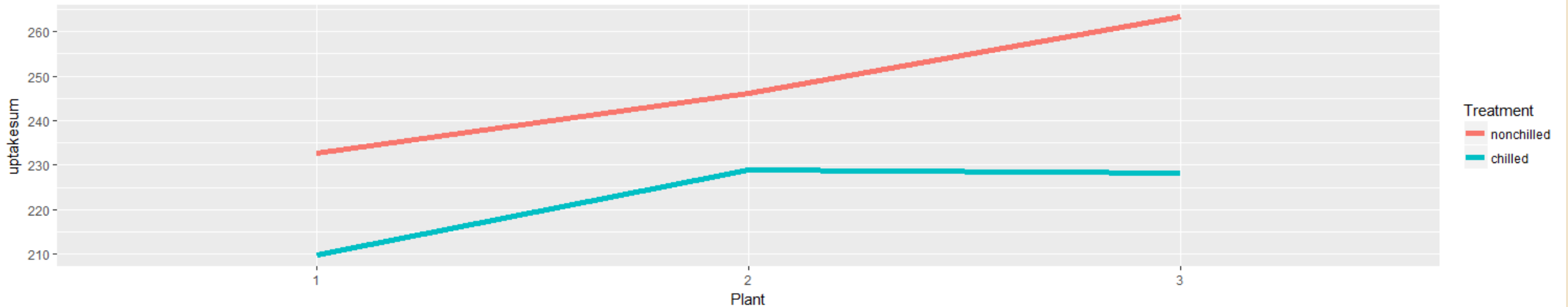
x를 x축, y를 y축으로 하는 선 그래프를 그리고 싶을 때 사용  
Qn1의 uptake가 16.0 30.4 34.8 37.2 35.3 39.2 39.70이라서 이런 결과 출력

## 2

## example: geom\_line

## CODE

```
psat_C02 <- group_by(C02, Type, Plant, Treatment) %>%  
  summarize(uptakesum=sum(uptake)) # Type, Plant, Treatment별 uptake합  
psat_C02$Plant <- substr(psat_C02$Plant, 3, 3) # 식물 이름 바꾸기  
filter(psat_C02, Type=='Quebec') %>%  
  ggplot(aes(x=Plant, y=uptakesum, group=Treatment, color=Treatment)) +  
  geom_line(size=2) # Type=Quebec인 row만 가지고 Treatment별 그래프 생성
```



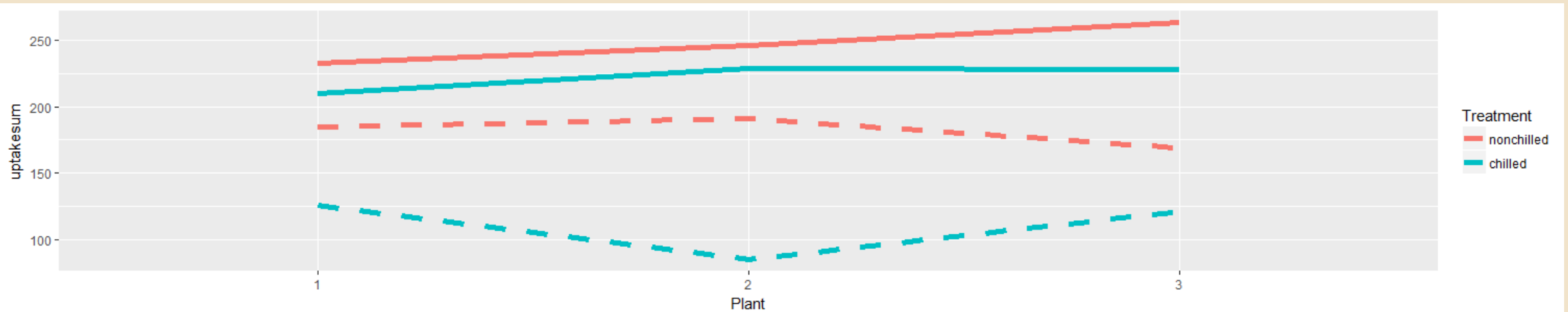
Type=Quebec의 Treatment여부 별 식물 별 uptake의 합의 선 그래프

## 2

## example: geom\_line

## CODE

```
ggplot() +  
  geom_line(data=filter(psat_CO2, Type=='Quebec'), aes(x=Plant, y=uptakesum,  
    group=Treatment, color=Treatment), size=2) +  
  geom_line(data=filter(psat_CO2, Type=='Mississippi'), aes(x=Plant, y=uptakesum,  
    group=Treatment, color=Treatment), size=2, linetype='dashed')
```



Type별(실선/점선) Treatment별(색) uptakesum 그래프

\*ggplot 내의 aesthetic에선 group을 하나밖에 지정할 수 없어서 geometry를 두개로 나눠서 코딩함.

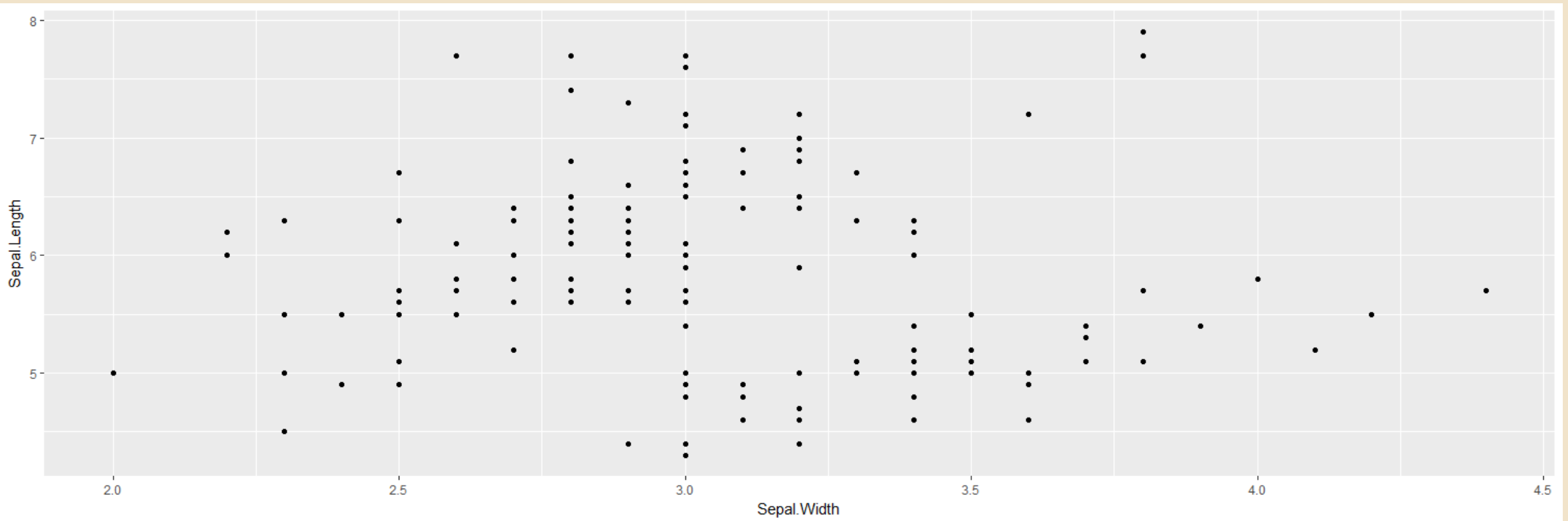


# 2

# Scatterplot: geom\_point

**CODE**

```
ggplot(iris,aes(x=Sepal.Width,y=Sepal.Length)) + geom_point()
```



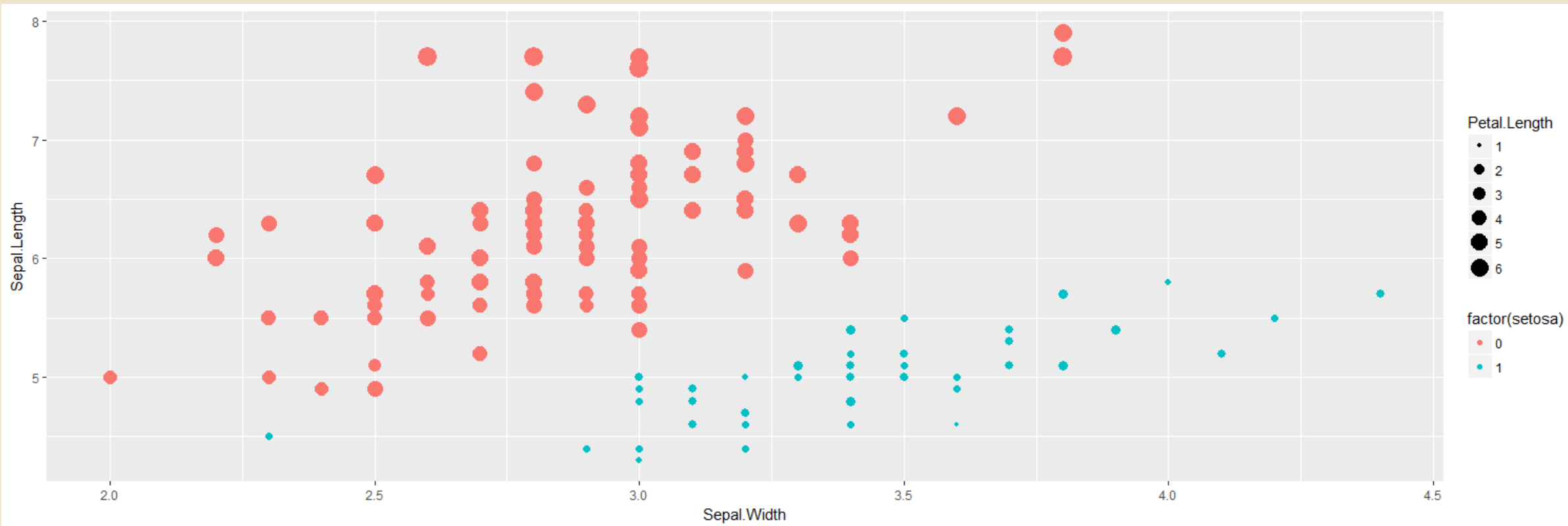
x를 x축, y를 y축으로 하는 scatter plot을 그리고 싶을 때 사용

## 2

## example: geom\_point

CODE

```
ggplot(iris,aes(x=Sepal.Width,y=Sepal.Length,color=factor(setosa),  
size=Petal.Length)) + geom_point()
```

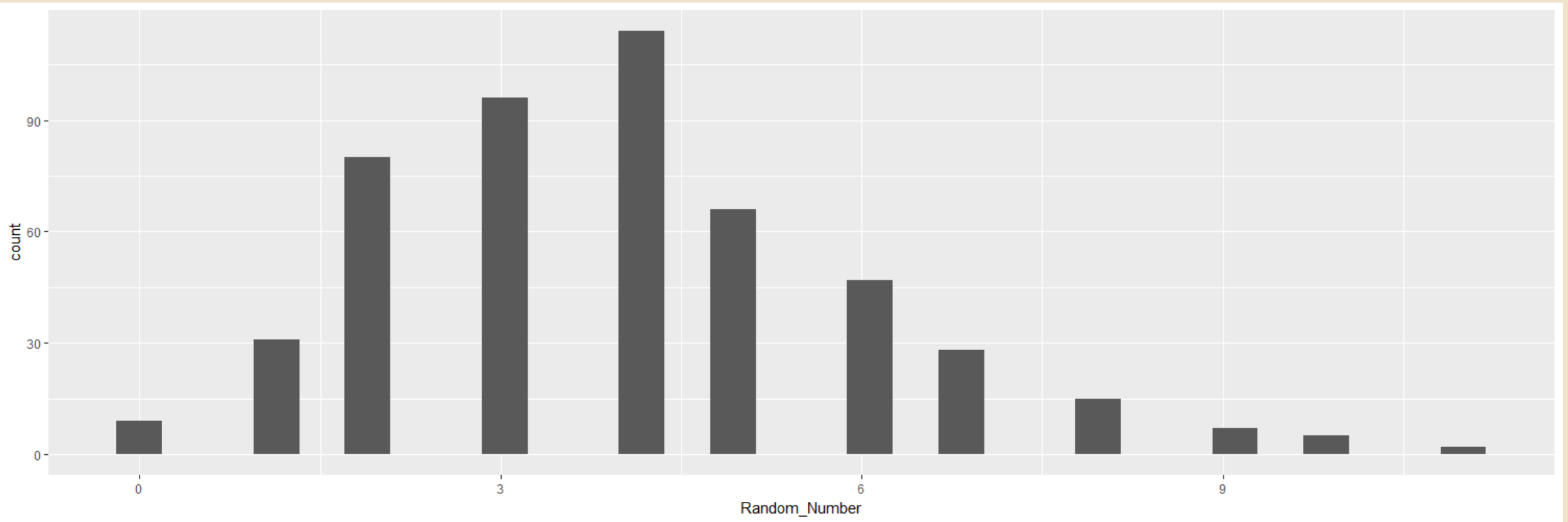


Aesthetic에 Petal.Length를 설정해주면 Petal.Length의 크기에 비례해 점의 크기가 커지도록 할 수 있다.  
Setosa가 다른 종들에 비해 Sepal.Width는 크지만 다른 요소들은 다른 종들에 비해 작다는 것을 한번에 볼 수 있다.

## 2 Plotting Distribution: geom\_histogram

CODE

```
set.seed(127); poissonNumber <- data.frame(Random_Number=rpois(500,4))  
ggplot(poissonNumber,aes(x=Random_Number)) + geom_histogram()
```



$\lambda=4$ 인 poisson random number를 생성하고 이를 히스토그램으로 나타낸 것.  
Aesthetic으로 x만 지정해주면 끝.

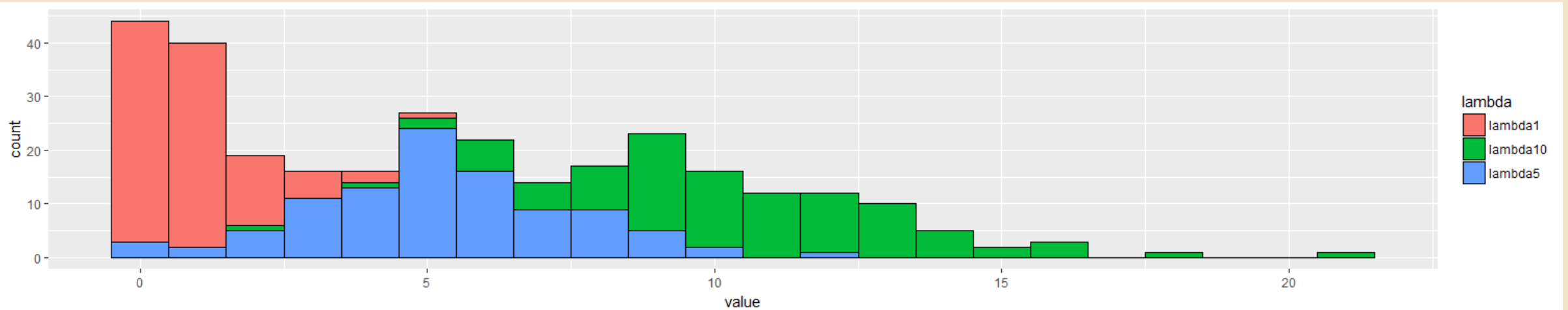


## 2

## example: geom\_histogram

## CODE

```
poisson_random <-  
data.frame(lambda1=rpois(100,1),lambda5=rpois(100,5),lambda10=rpois(100,10))  
poisson_random <- gather(poisson_random,lambda,value)  
ggplot(poisson_random,aes(x=value,fill=lambda)) +  
geom_histogram(color='black',binwidth=1)
```



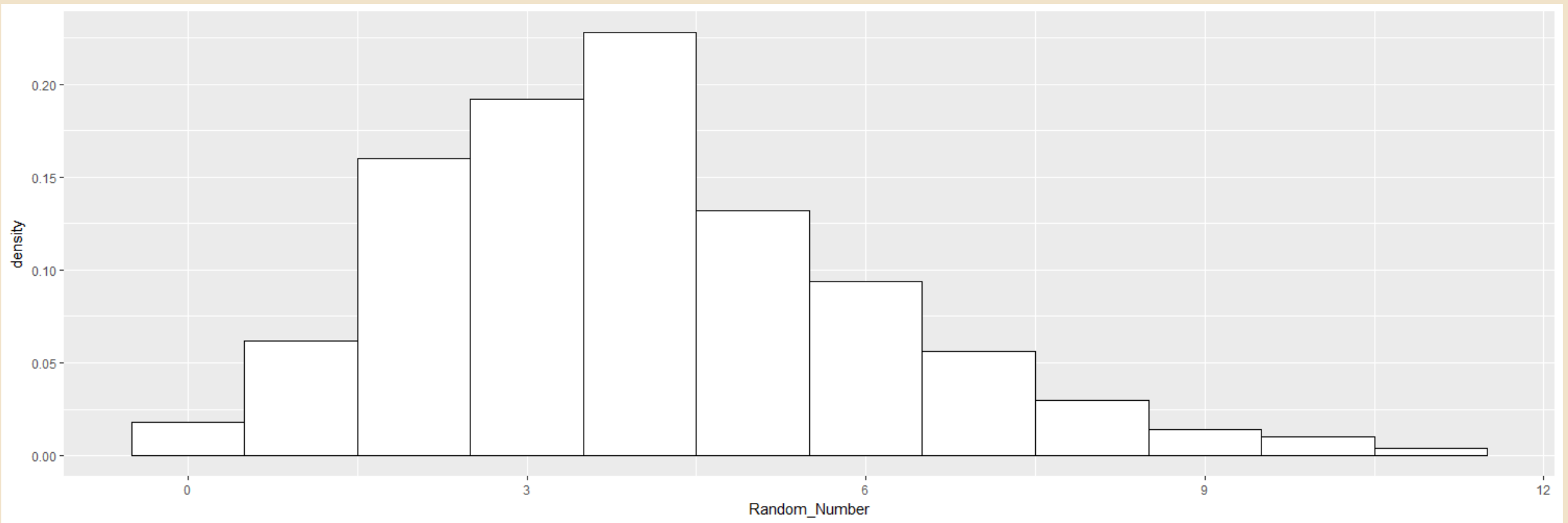
geometry 내의 expression으로 구간의 길이(binwidth)를  
설정할 수 있다는 것이 geom\_bar보다 좋은 점이다.

# 2

## example: geom\_histogram

### CODE

```
ggplot(poissonNumber,aes(x=Random_Number,y=..density..)) +  
geom_histogram(binwidth=1,fill='white',color='black')
```

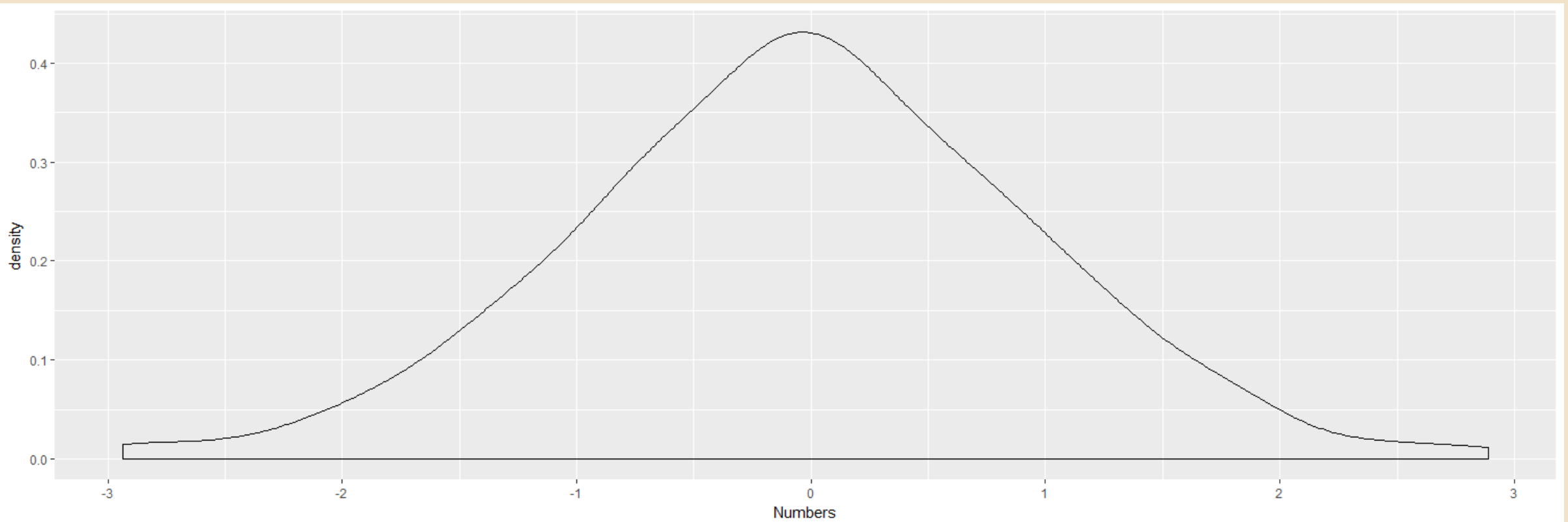


Aesthetic의 y에 ..density..를 입력하면 도수가 아니라 비율로 그려진 히스토그램을 얻는다.

## 2 Plotting Distribution: geom\_density

CODE

```
set.seed(127);normalNumber <- data.frame(Numbers=round(rnorm(500,0,1),4))  
ggplot(normalNumber,aes(x=Numbers))+geom_density()
```



평균이 0이고 분산이 1인 정규분포에서 500개의 숫자를 임의추출해 만든 밀도곡선  
여기서도 Aesthetic으로 x만 지정해주면 끝.

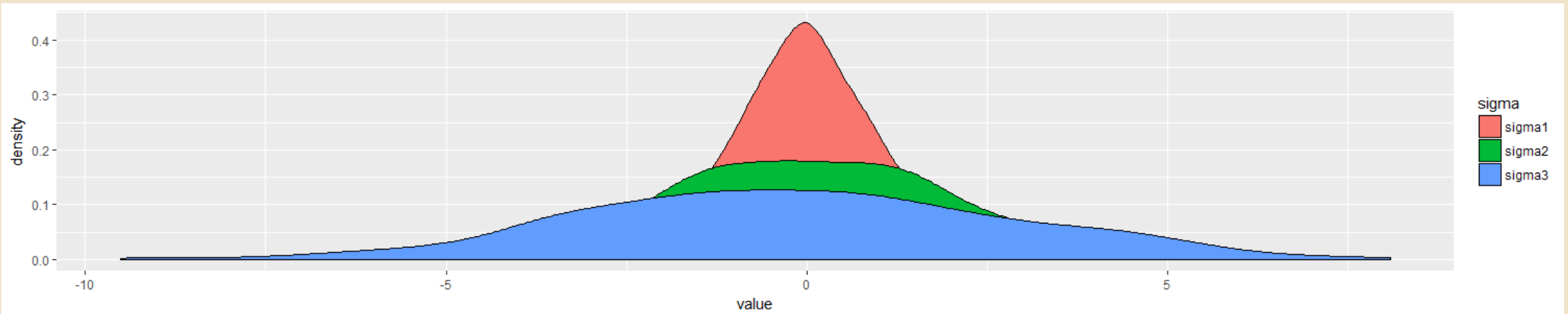


## 2

## example: geom\_density

## CODE

```
set.seed(127)
data.frame(sigma1=round(rnorm(500,0,1),4),sigma2=round(rnorm(500,0,2),4),
sigma3=round(rnorm(500,0,3),4)) %>%
gather(sigma,value) %>%
ggplot(aes(x=value,fill=sigma))+geom_density()
```

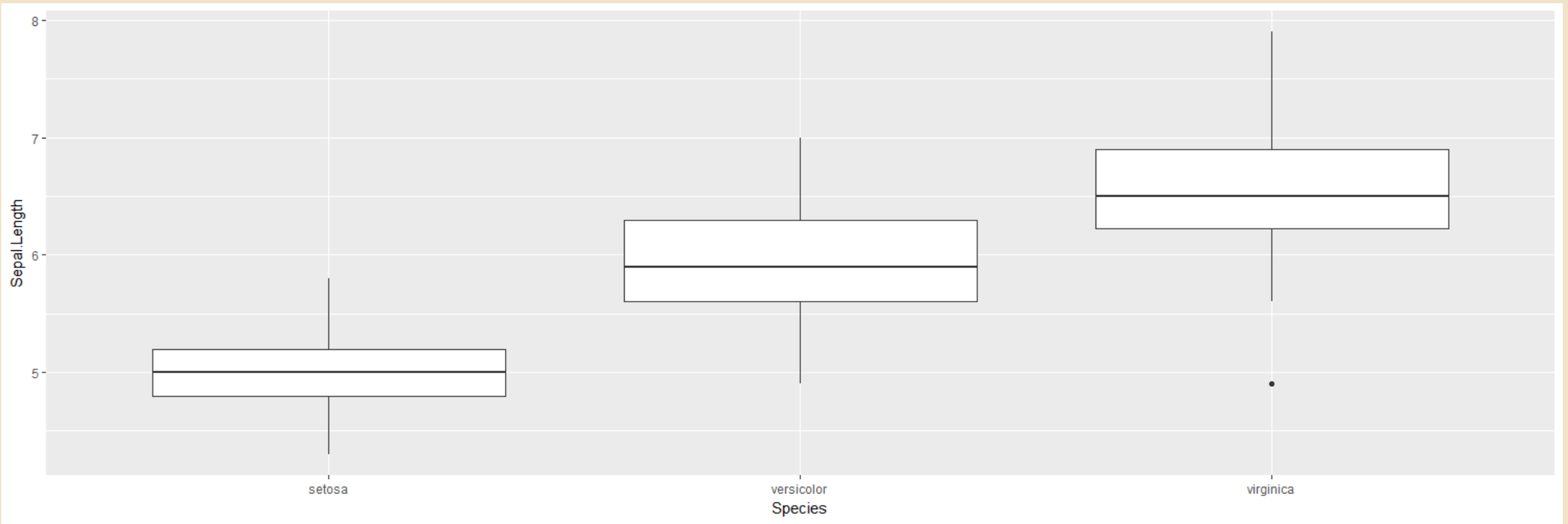


평균은 0이고 분산이 각각 1,2,3인 세 개의 정규분포 밀도곡선이다.  
여러 분포를 이렇게 겹쳐서 그릴 수도 있다.

# 2 Plotting Distribution: geom\_boxplot

**CODE**

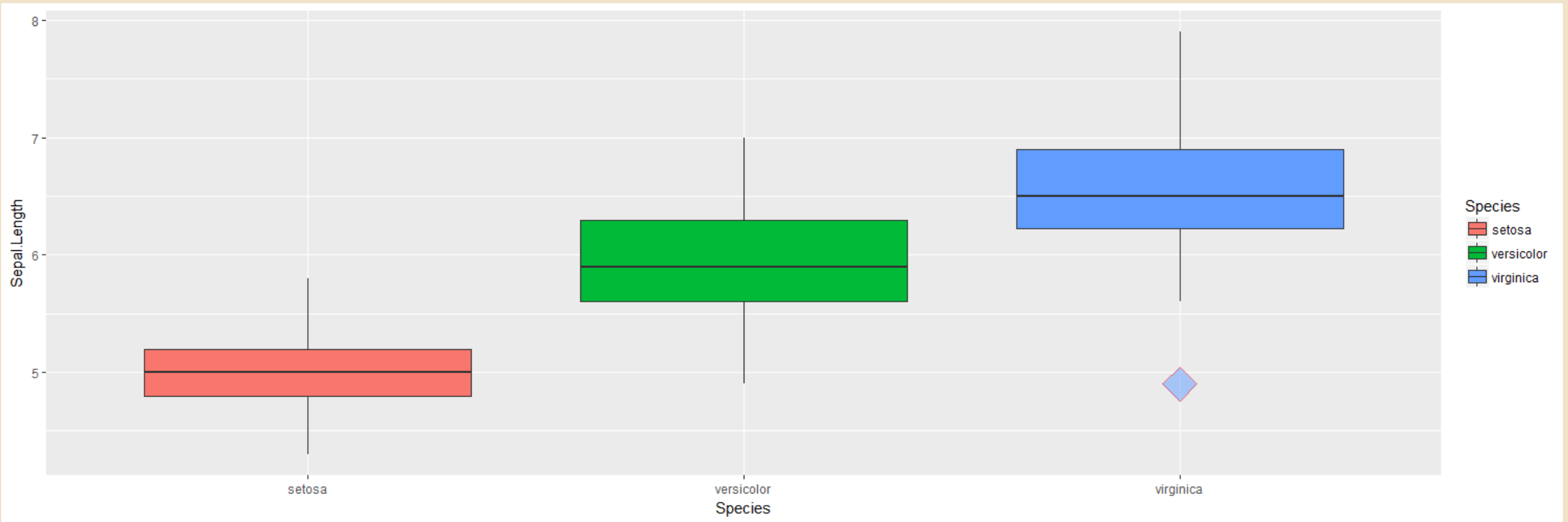
```
ggplot(iris,aes(x=Species,y=Sepal.Length))+geom_boxplot()
```



## 2 Plotting Distribution: geom\_boxplot

CODE

```
ggplot(iris,aes(x=Species,y=Sepal.Length,fill=Species))+  
geom_boxplot(outlier.color='red',outlier.shape=23,outlier.size=9,outlier.alpha=0.5)
```



geometry 내 expression에서 outlier의 외형에 대한 설정을 해 줄수 있다.



# 3 MANUAL OPTIONS





# 3

## Manual options

### ◆ Footnote: 각주 추가

- [annotate](#)
- [geom\\_abline](#)
- [geom\\_hline\(vline\)](#)

### ◆ Axis: 축 설정

- [scale\\_A\\_B](#)

### ◆ [Legend](#): 범례 설정

### ◆ [Faceting](#): 면 분할하기

### ◆ Manual graphic handling: 세부 서식 설정

- [Scale](#): geometry 내부 서식
- [Theme](#): geometry 외부 서식

# 3

## Footnote: annotate

CODE

```
ggplot(iris,aes(x=Species,y=Sepal.Length))+geom_boxplot() +  
annotate('text',x='virginica',y=5,label='outlier')
```



**annotate(geometry, location, expression)**

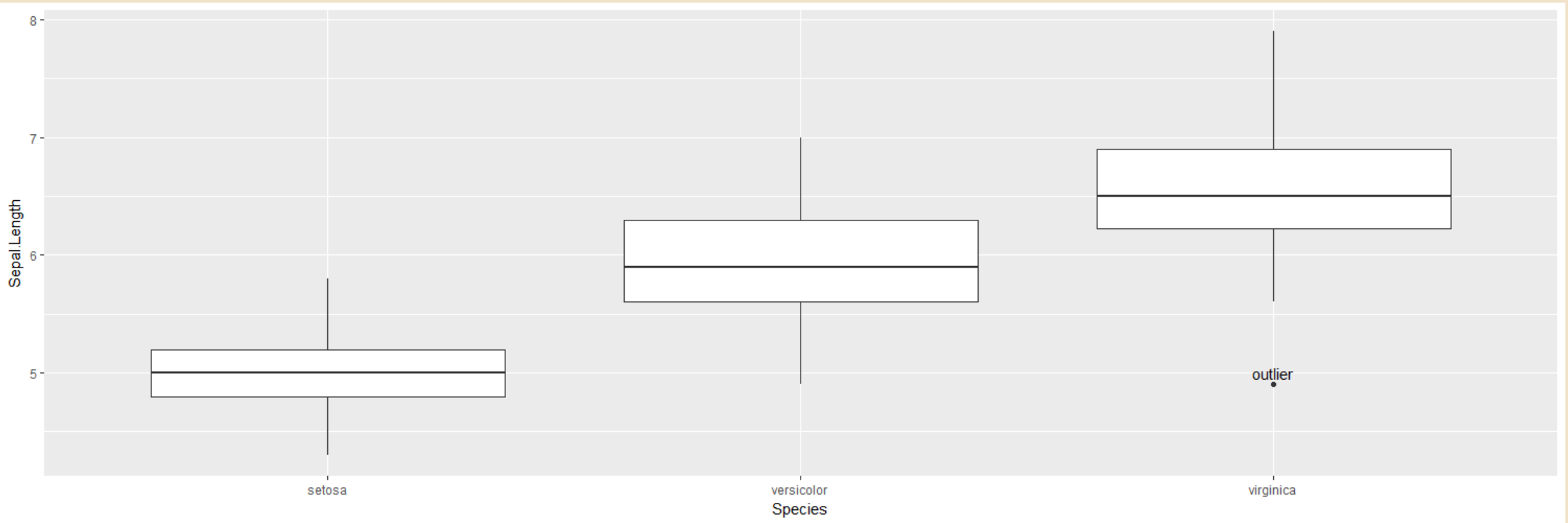
- **geometry**: 사용하고자 하는 각주의 형태  
→ text, rect, segment 셋 중 하나
- **location**: 각주의 위치에 대한 표현
  - ✓ text: x,y좌표(x는 텍스트의 중심이 위치하는 좌표)
  - ✓ rect: xmin,xmax,ymin,ymax(사각형의 네 꼭지점)
  - ✓ segment: x,xend,y,yend(x(y)부터 xend(yend)까지 선분 생성)
- **expression**: 각주를 구체화하기 위한 기타 표현

# 3

## Footnote: annotate

CODE

```
ggplot(iris,aes(x=Species,y=Sepal.Length))+geom_boxplot() +  
annotate('text',x='virginica',y=5,label='outlier')
```



`annotate('text',x='virginica',y=5,label='outlier'):`  
x='virginica',y=5에 label='outlier'인 text 주석 레이어를 입혔다.

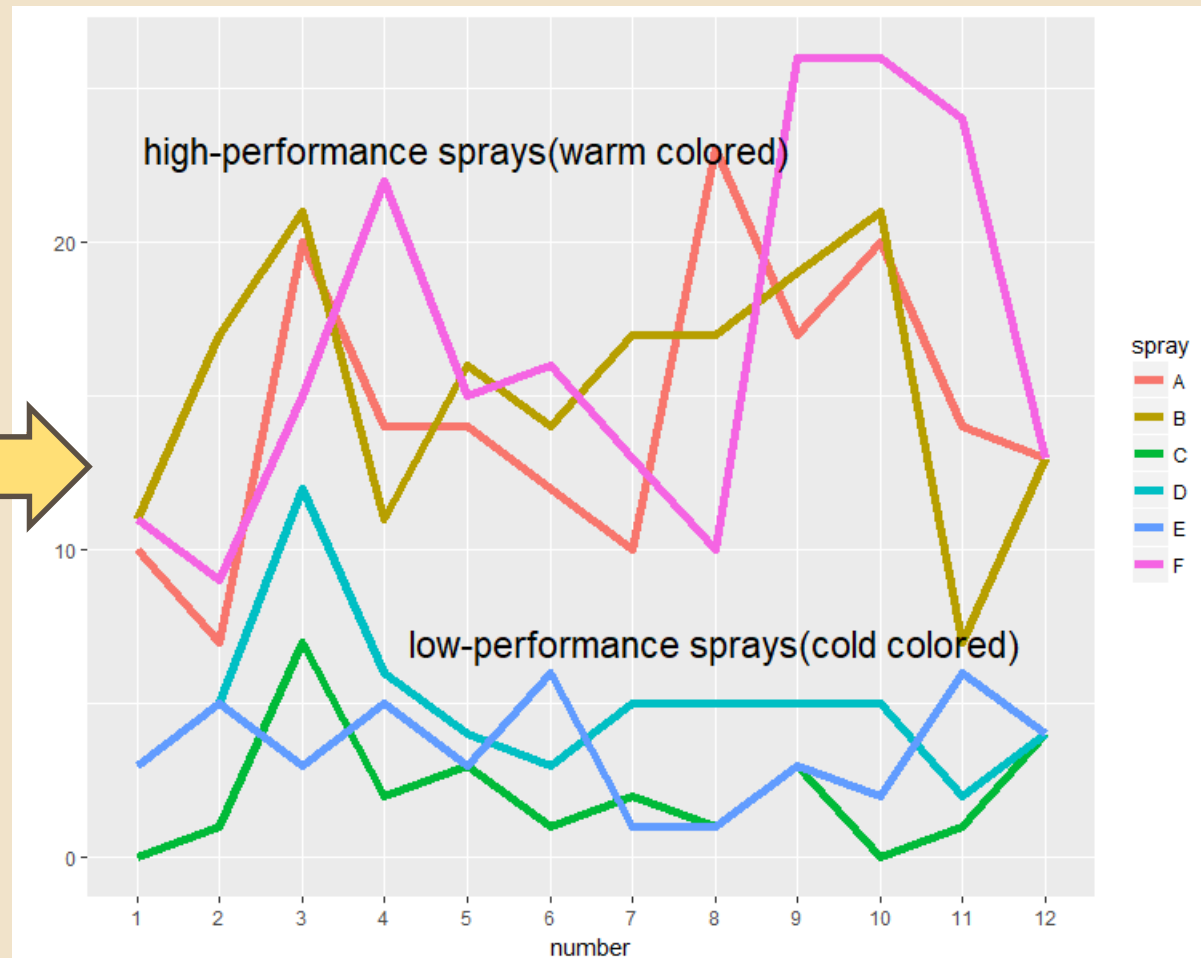
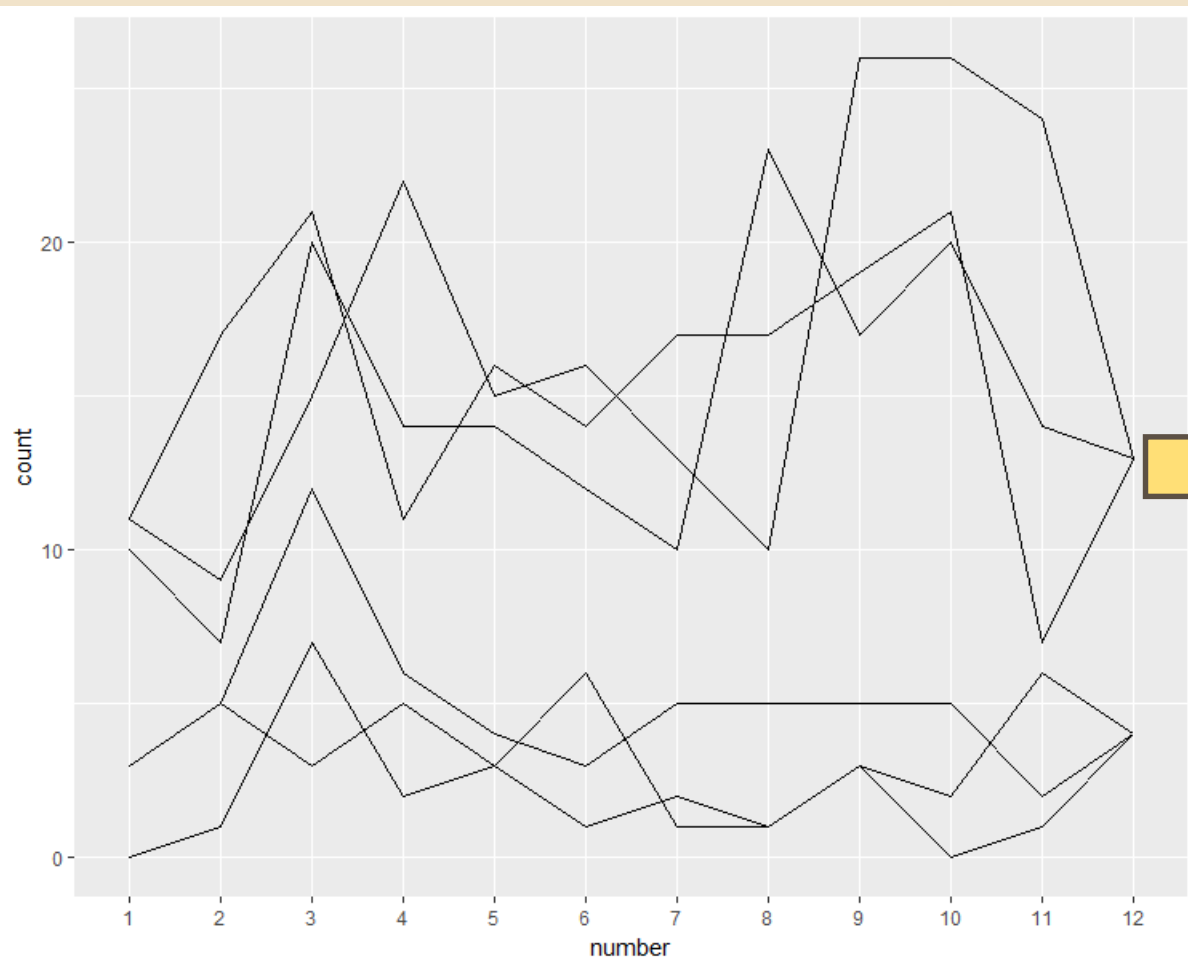


## 3

## example: annotate(text)

## CODE

```
ggplot(InsectSprays, aes(x=number, y=count, group=spray, color=spray))+geom_line(size=2)+  
  annotate('text', x=5, y=23, label='high-performance sprays(warm colored)', size=6)+  
  annotate('text', x=8, y=7, label='low-performance sprays(cold colored)', size=6)
```

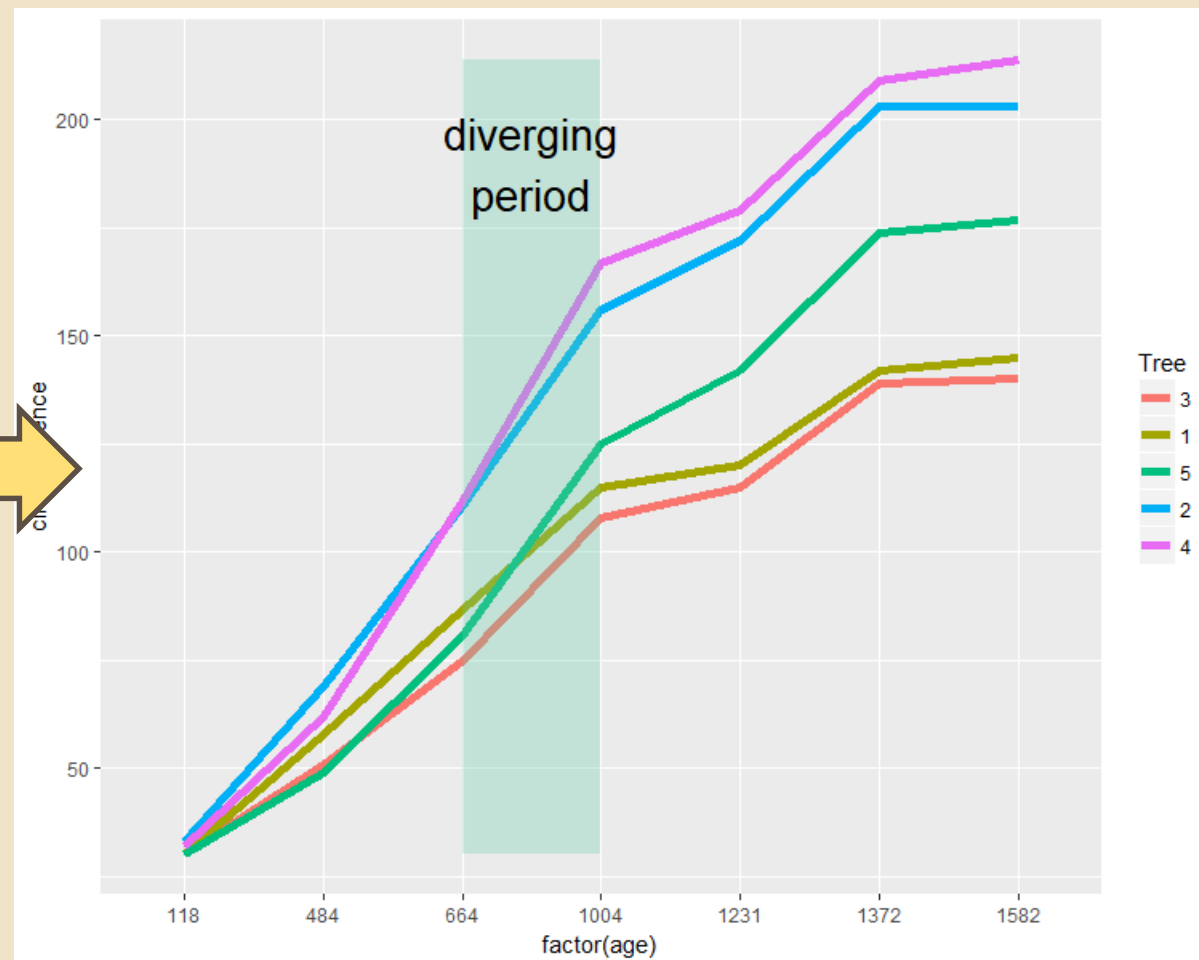
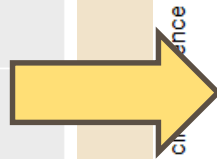
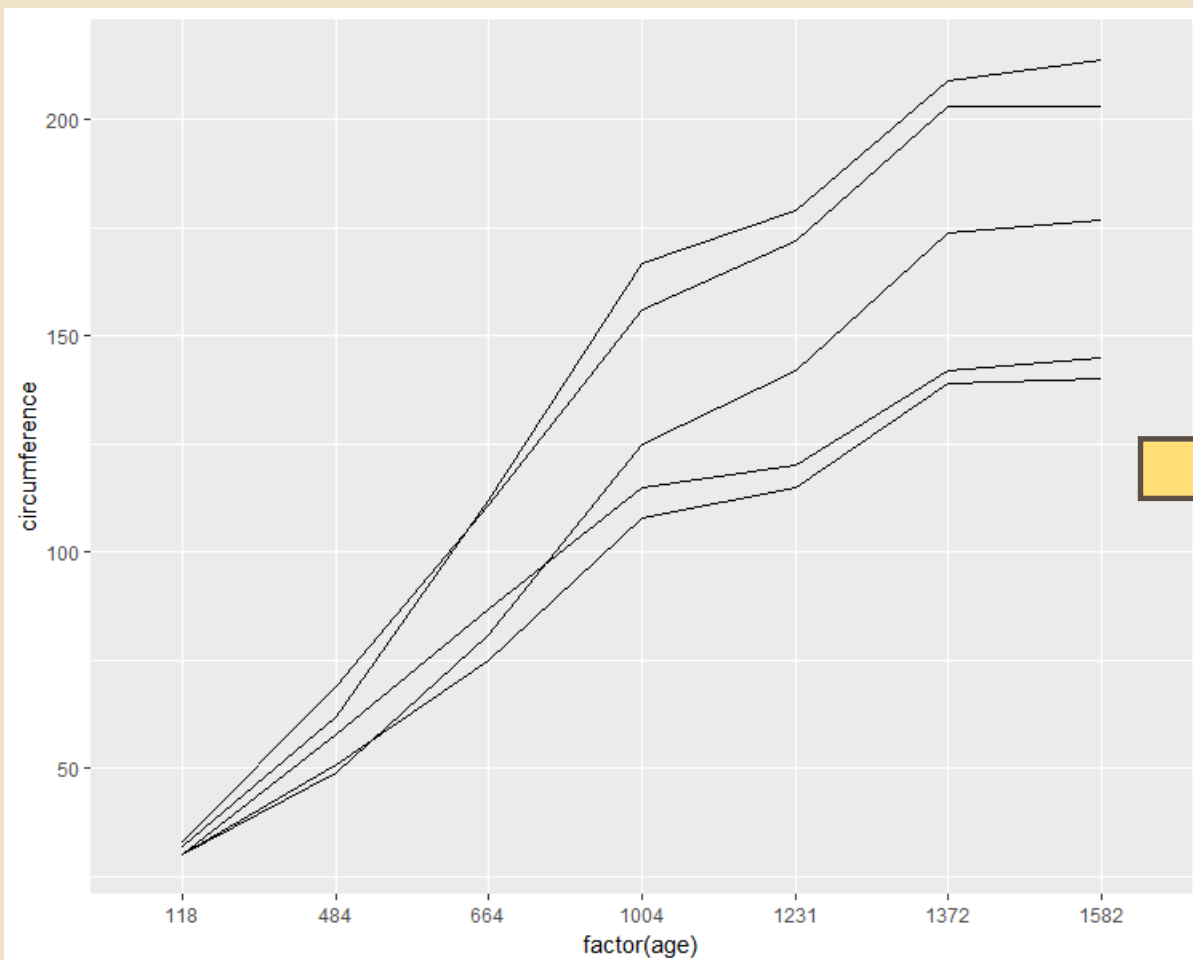


## 3

## example: annotate(text, rect)

## CODE

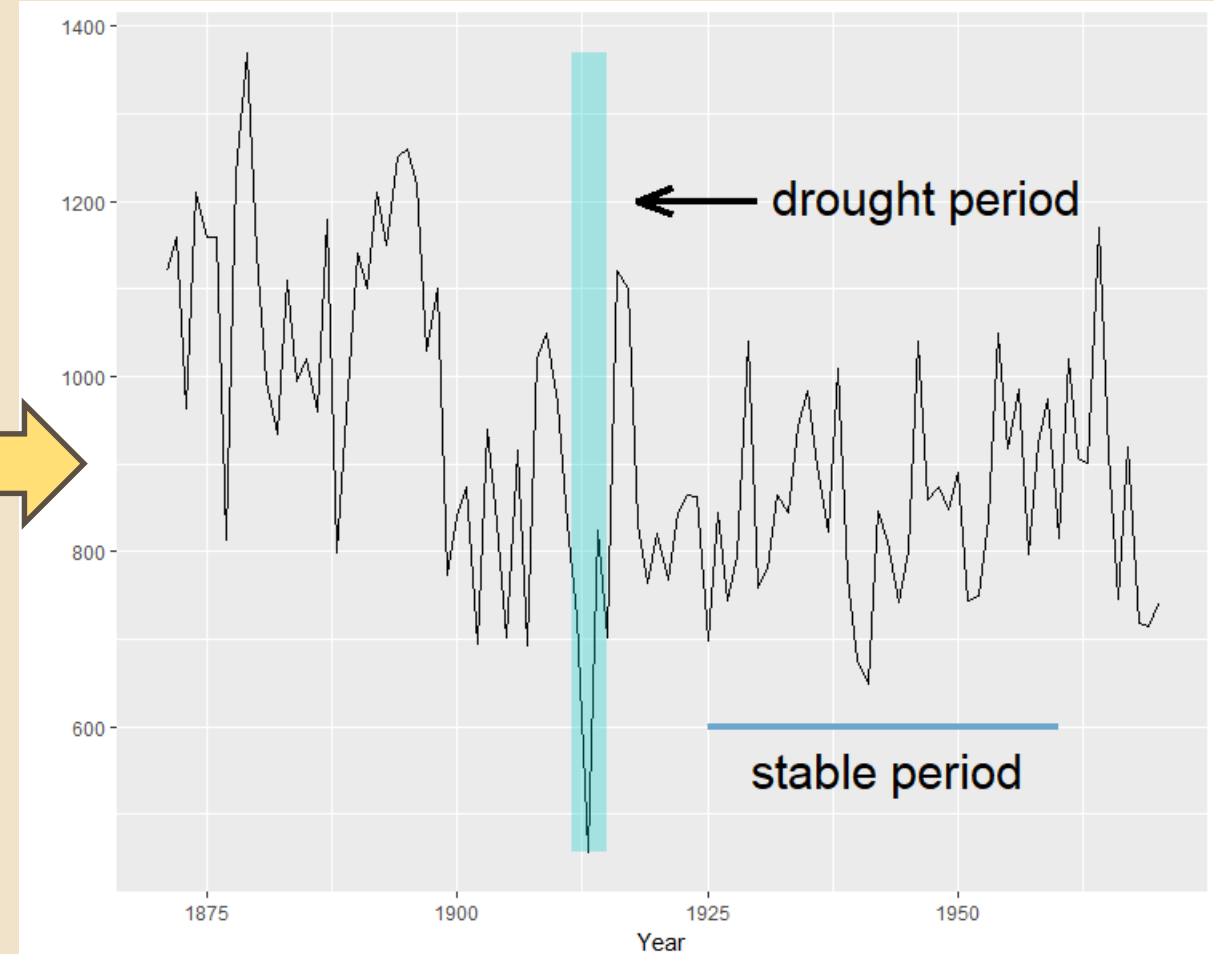
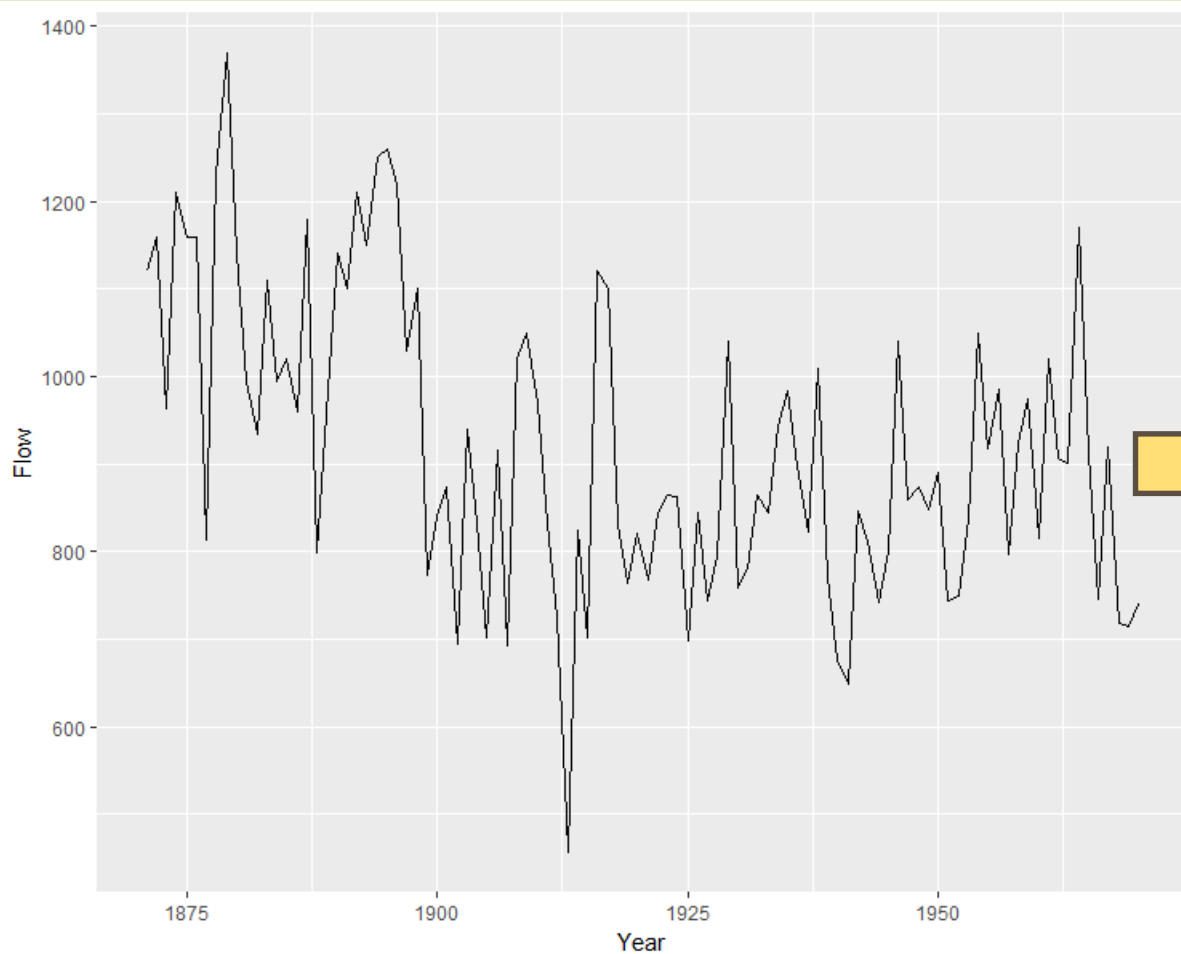
```
ggplot(Orange, aes(x=factor(age), y=circumference, group=Tree, color=Tree))+geom_line(size=2)+
  annotate('rect', xmin=3, xmax=4, ymin=30, ymax=max(Orange$circumference), fill='aquamarine3', alpha=0.3)+
  annotate('text', x=3.5, y=190, label='diverging w period', size=7)
```



# 3 example: annotate(text, rect, segment)

## CODE

```
data.frame(Year=1871:1970,Flow=as.numeric(Nile)) %>% ggplot(aes(x=Year,y=Flow))+geom_line()+  
  annotate('rect',xmin=1871+which(Nile==min(Nile))-2.5,xmax=1871+which(Nile==min(Nile))+1,ymin=min(Nile),ymax=max(Nile), alpha=0.3,fill='cyan3')+  
  annotate('segment',x=1871+which(Nile==min(Nile))+4,xend=1871+which(Nile==min(Nile))+16,y=1200,yend=1200,arrow=arrow(ends='first',angle=20),size=1.5)+  
  annotate('text',x=1871+which(Nile==min(Nile))+33,y=1205,label='drought period',size=8) +  
  annotate('segment',x=1925,xend=1960,y=600,yend=600,size=1.5,color='skyblue3')+  
  annotate('text',x=1943,y=550,label='stable period',size=8)
```

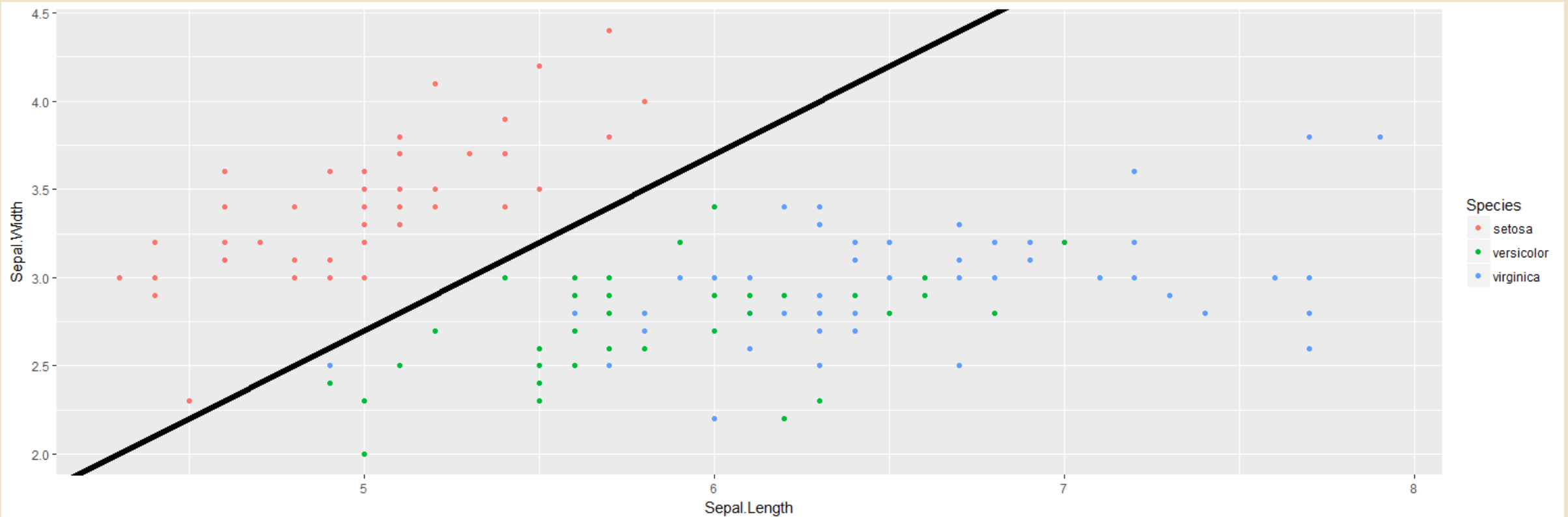


## 3

## Footnote: geom\_abline

CODE

```
ggplot(iris,aes(x=Sepal.Length,Sepal.Width,color=Species))+  
geom_point()+geom_abline(intercept=-2.3,slope=1,size=2)
```



지정한 옵션(intercept,slope)에 맞는 직선( $y=ax+b$ )을 그려주는 함수

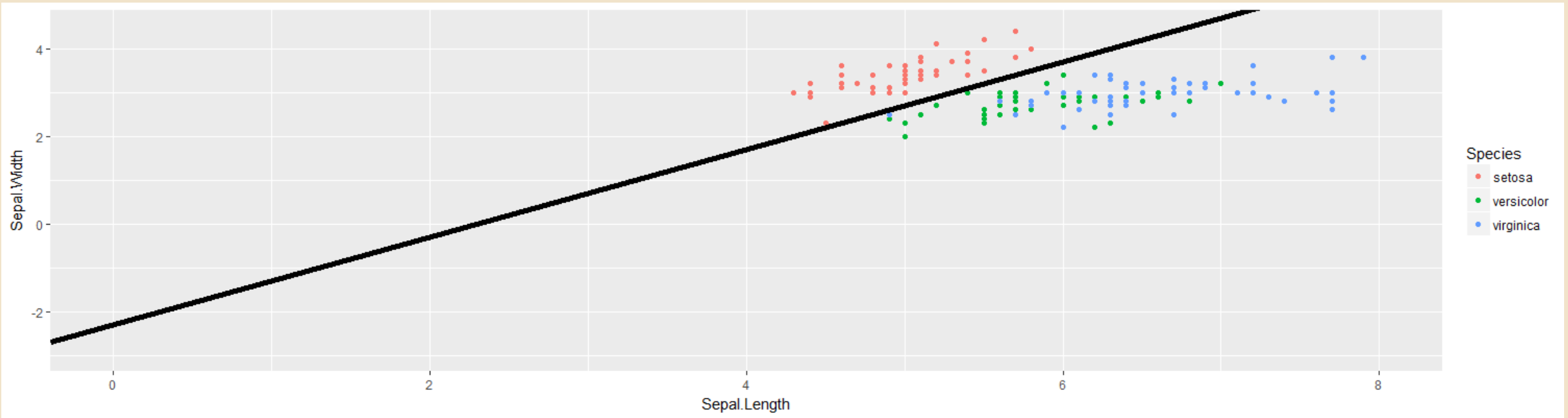


## 3

## Footnote: geom\_abline

CODE

```
ggplot(iris,aes(x=Sepal.Length,Sepal.Width,color=Species))+geom_point()+  
geom_abline(intercept=-2.3,slope=1,size=2)+xlim(0,8)+ylim(-3,4.5)
```



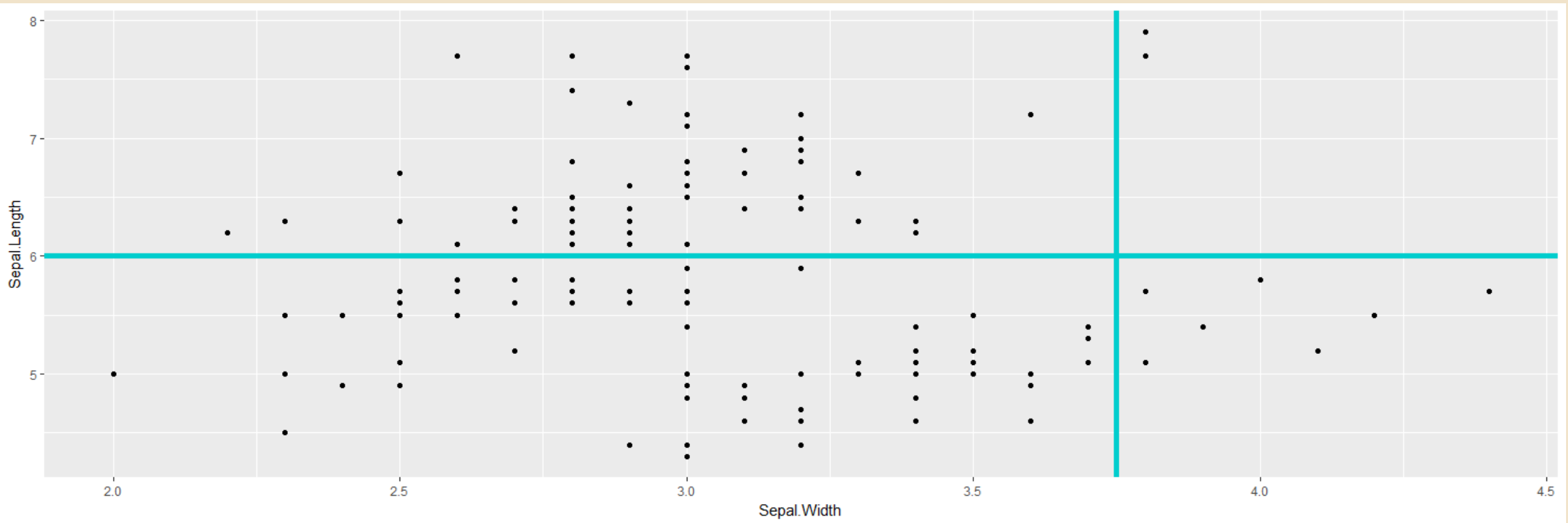
이때, intercept는 원점을 기준으로 하는 값(y절편)이기 때문에  
원점을 포함하도록 xlim,ylim을 조정해 zoom out한 다음에  
intercept와 slope의 값을 여러 개 시도해 보며 시행착오를 거치는게 합리적.

# 3

## Footnote: geom\_hline, geom\_vline

### CODE

```
ggplot(iris,aes(x=Sepal.Width,y=Sepal.Length))+geom_point()+  
geom_hline(yintercept=6,size=2,color='cyan3')+  
geom_vline(xintercept=3.75,size=2,color='cyan3')
```



수평선(horizontal line)이나 수직선(vertical line)을 생성하고자 할 때  
abline 대신 사용할 수 있는 geometry

CODE

```
ggplot(iris,aes(x=Species,y=Sepal.Length))+geom_boxplot()+  
scale_x_discrete(limits=c('virginica','setosa','versicolor'))
```

↓

scale\_A\_B(expression,...)

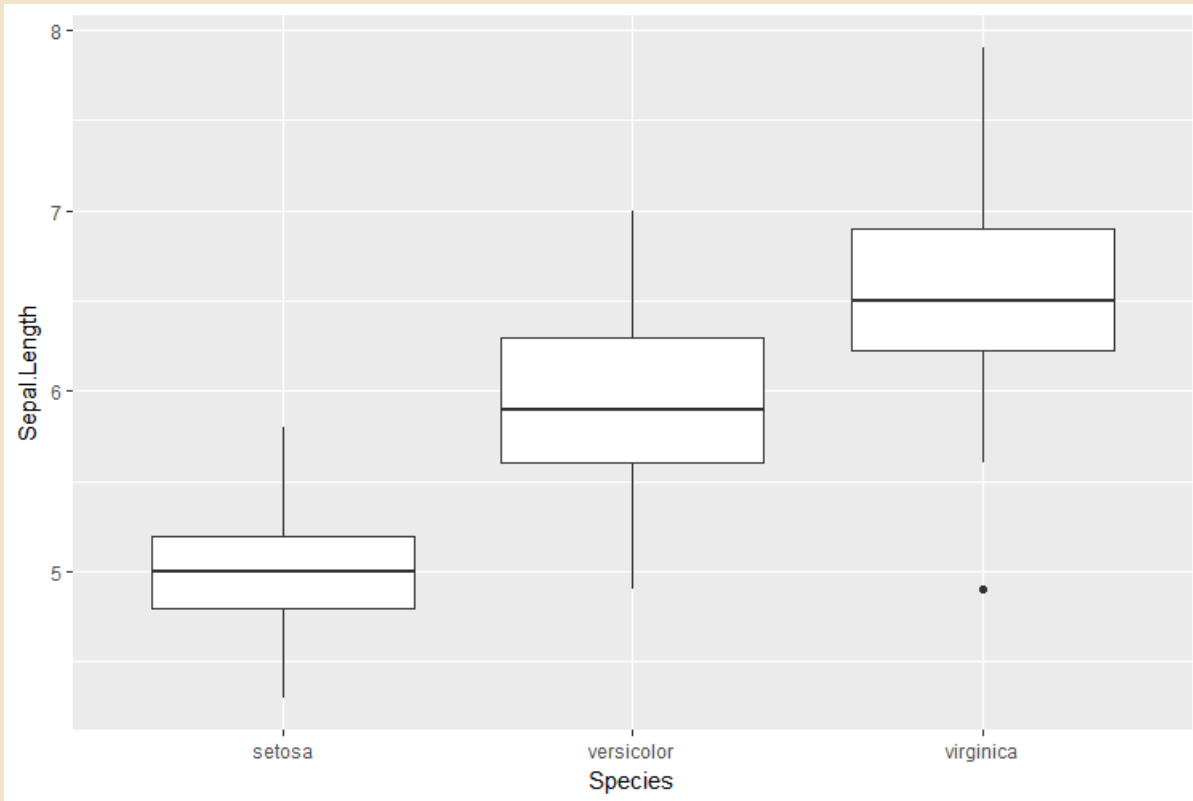
- A : 조정할 축(x, y)
- B : 그 축의 변수 type(discrete, continuous)
- expression
  - ✓ limits : (연속형) 그 축의 범위(xlim, ylim과 같은 역할)  
(이산형) 눈금에 표시할 값들의 목록(순서 유의미!)
  - ✓ breaks: 눈금을 표시할 위치를 특정하는 expression  
(breaks=NULL이면 눈금을 다 지워줌)
  - ✓ label: 표시할 위치에 표현할 눈금의 이름

## 3

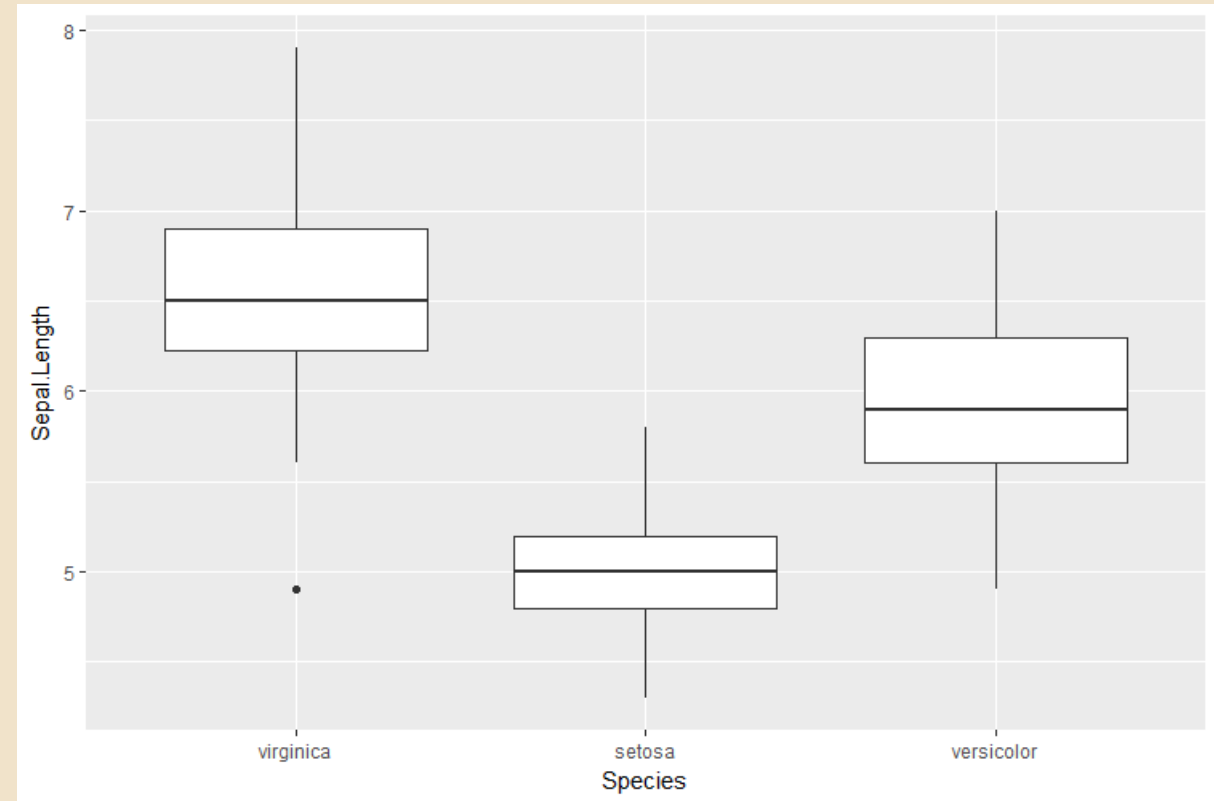
## Axis: scale\_x(y)\_B

CODE

```
ggplot(iris,aes(x=Species,y=Sepal.Length))+geom_boxplot()+
scale_x_discrete(limits=c('virginica','setosa','versicolor'))
```



< scale\_x\_discrete layer가 없는 코드의 결과 >



< limits 표현대로 순서가 바뀌었음을 확인할 수 있음 >

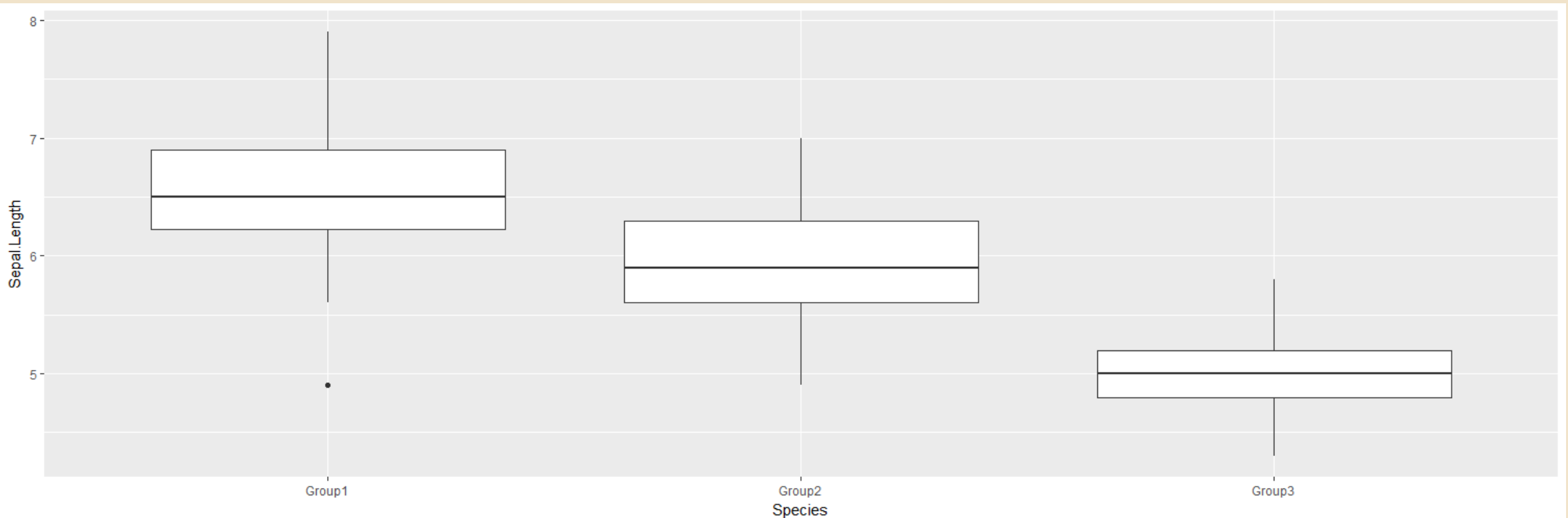


## 3

## Axis: scale\_x(y)\_B

CODE

```
ggplot(iris,aes(x=Species,y=Sepal.Length))+geom_boxplot()+  
scale_x_discrete(limits=c('virginica','versicolor','setosa'),  
labels=c('Group1','Group2','Group3'))
```

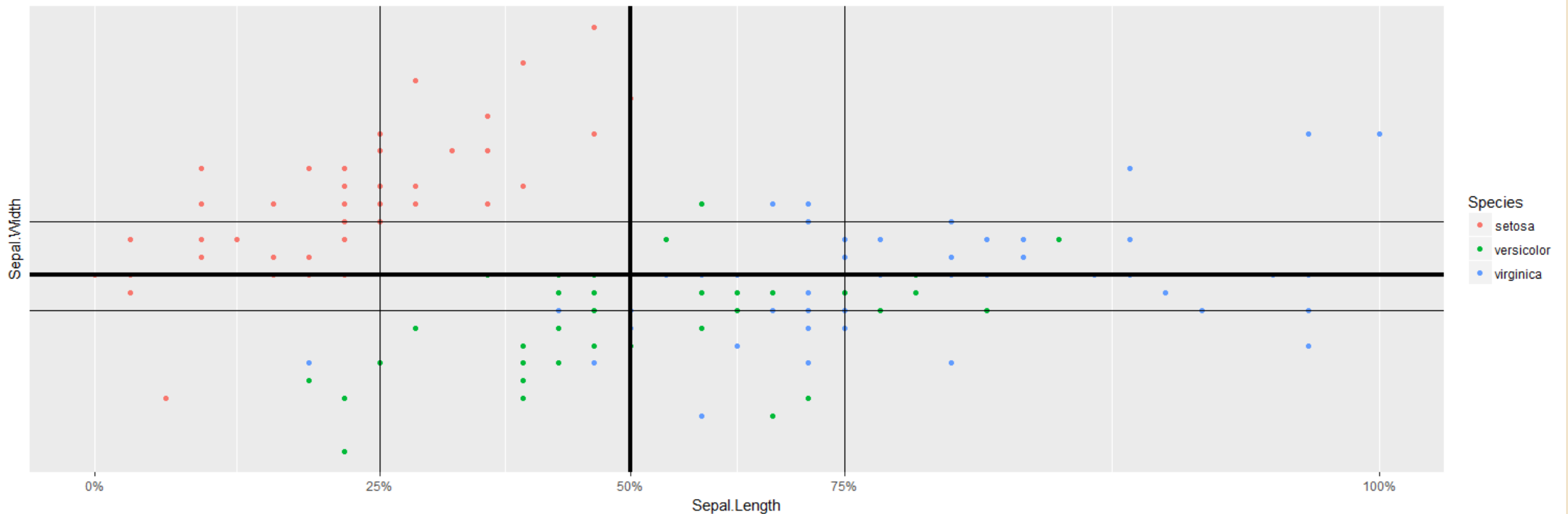


limits 표현대로 순서가 바뀌고, labels 표현대로 이름이 바뀐 것을 확인할 수 있음

# 3 example: scale\_x(y)\_B & geom\_h(v)line

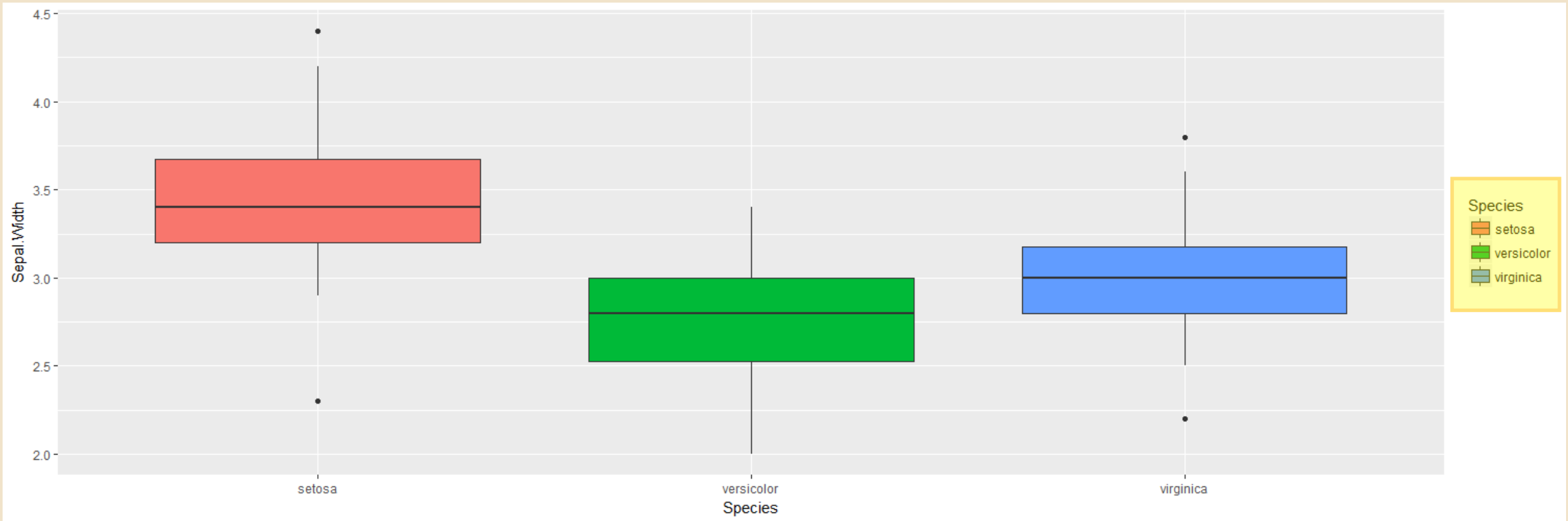
## CODE

```
ggplot(iris,aes(x=Sepal.Length,y=Sepal.Width,color=Species))+geom_point()+  
scale_x_continuous(breaks=quantile(iris$Sepal.Length))+  
geom_vline(xintercept = quantile(iris$Sepal.Length)[2])+  
geom_vline(xintercept = quantile(iris$Sepal.Length)[3],size=1.5)+  
geom_vline(xintercept = quantile(iris$Sepal.Length)[4])+  
scale_y_continuous(breaks=NULL)+  
geom_hline(yintercept = quantile(iris$Sepal.Width)[2])+  
geom_hline(yintercept = quantile(iris$Sepal.Width)[3],size=1.5)+  
geom_hline(yintercept = quantile(iris$Sepal.Width)[4])
```



# 3

## Legend



## Legend(범례)

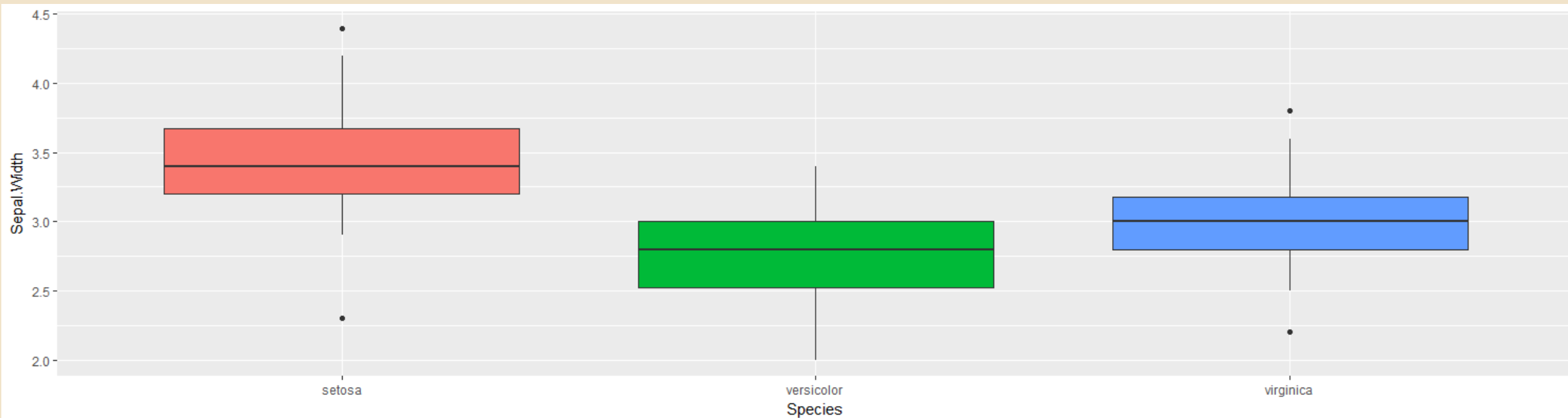
항목별로 구별된 서식을 항목별로 명시해주는 안내표(guide)

# 3

## Legend

CODE

```
ggplot(iris,aes(x=Species,y=Sepal.Width,fill=Species))+geom_boxplot()+  
guides(fill=FALSE)
```



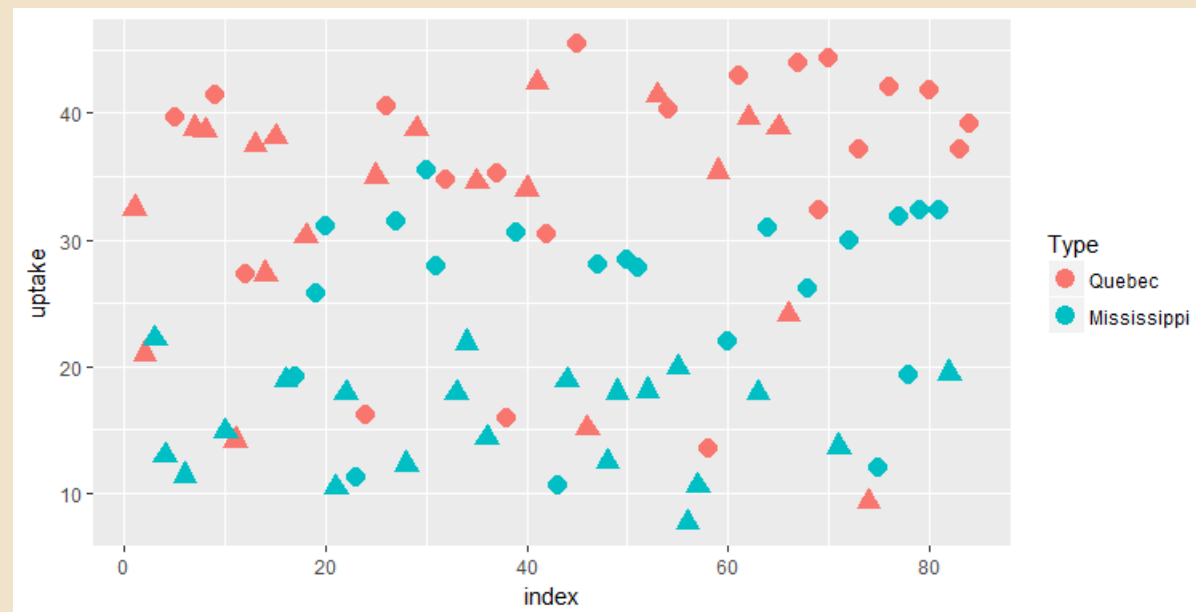
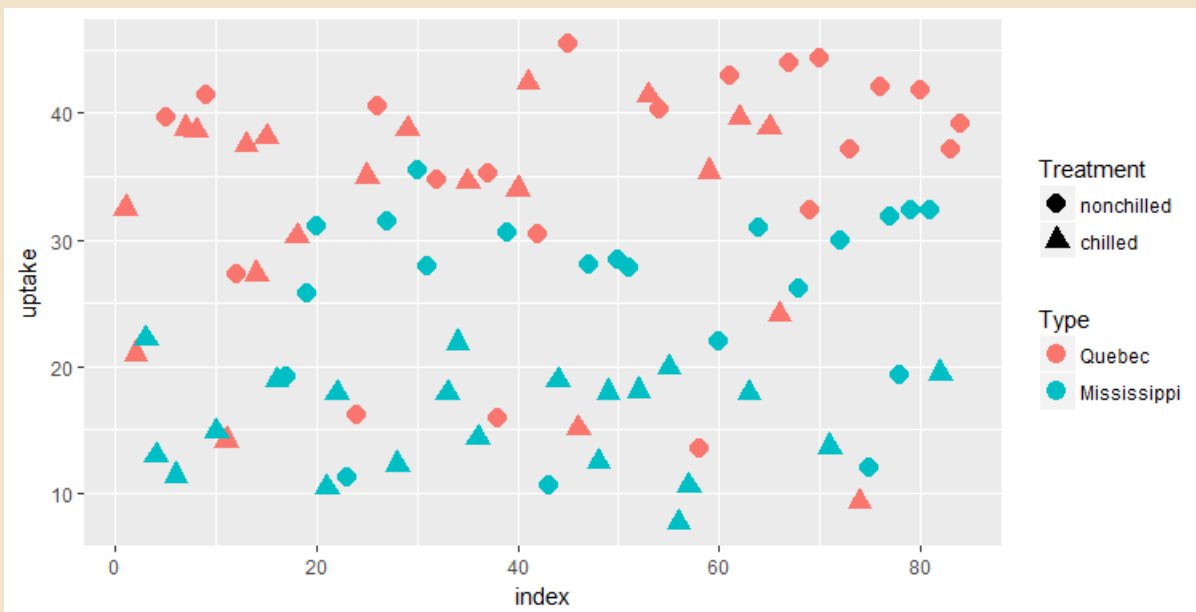
만약 범례를 없애고 싶다면 guides 레이어 안에다가  
구분된 서식을 FALSE라고 표현해주면 된다.

## 3

## example: deleting legend

## CODE

```
set.seed(1027);C02$index <- sample(1:84,replace=F)
ggplot(C02,aes(x=index,y=uptake,color=Type,shape=Treatment))+
  geom_point(size=4)+guides(shape=FALSE)
```



ggplot object에서 color와 shpae에 대한 서식을 구별할 것을 요구했다.  
guides 레이어를 통해 shape의 범례만을 제거했다.  
범례 서식에 대한 더 많은 내용은 뒤의 theme 부분에서 한번에 전개될 예정

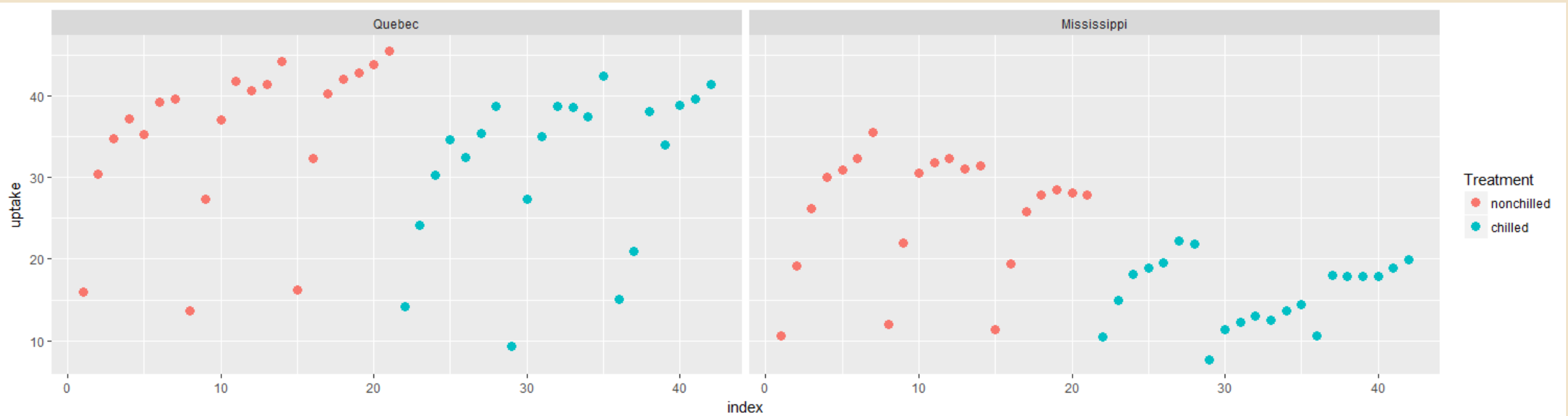


## 3

## Faceting : facet\_grid

CODE

```
C02$index <- rep(1:42,2)
ggplot(C02,aes(x=index,y=uptake,color=Treatment))+geom_point(size=3)+
facet_grid(~Type)
```



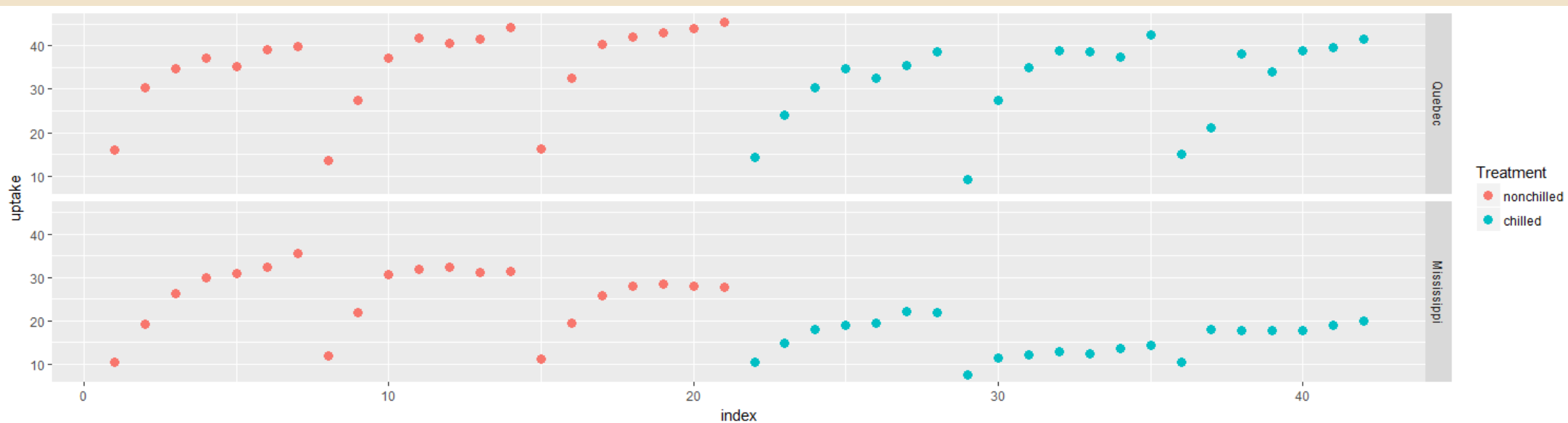
각기 다른 범주에 대한 시각화를 서로 다른 면에 표현할 수 있게 해줌  
이때 분할 기준이 되는 범주는 반드시 **aesthetic**에 포함될 필요는 없음!  
(Type 변수를 기준으로 면을 분할했는데 이 변수는 **aesthetic**에 포함되어 있지 않음을 알 수 있음)

## 3

## Faceting : facet\_grid

CODE

```
C02$index <- rep(1:42,2)
ggplot(C02,aes(x=index,y=uptake,color=Treatment))+geom_point(size=3)+
facet_grid(Type~.)
```



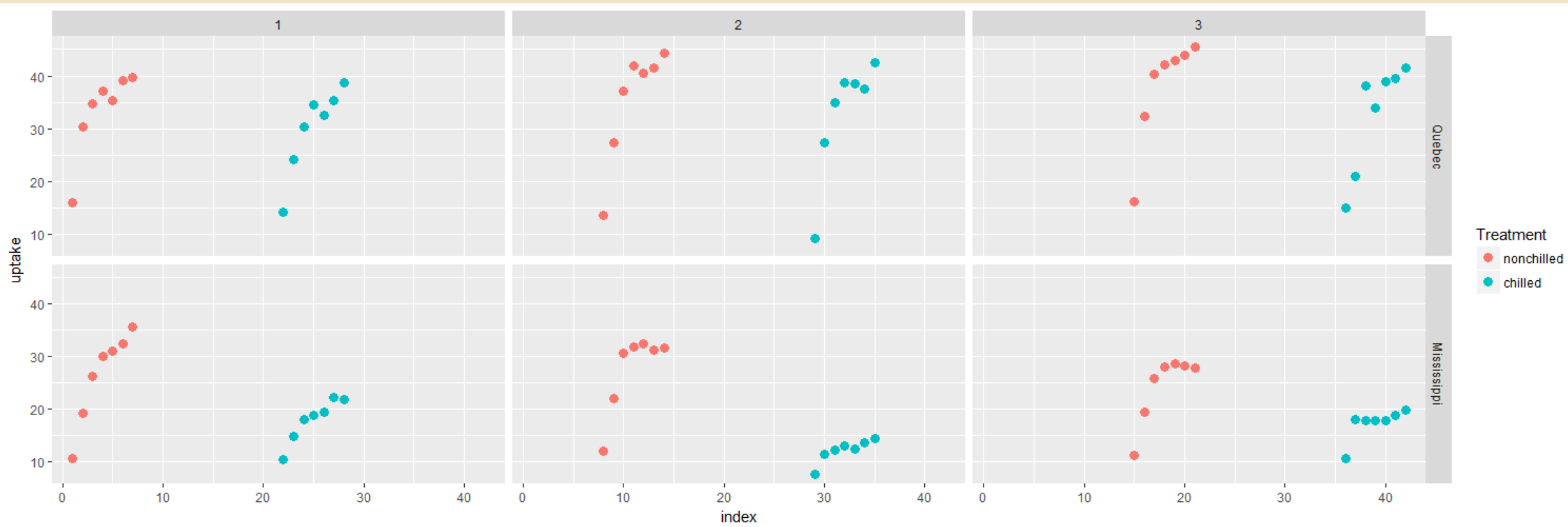
분할 기준이 될 변수 A에 대해,  
facet\_grid(A~.) : 가로 방향 분할  
facet\_grid(~A) : 세로 방향 분할

## 3

## Faceting : facet\_grid

CODE

```
C02$index <- rep(1:42,2);C02$Plant_type <- substr(C02$Plant,3,3)
ggplot(C02,aes(x=index,y=uptake,color=Treatment))+geom_point(size=3)+
facet_grid(Type~Plant_type)
```



둘 다 써서 양방향으로 분할하는 것도 가능

## 3

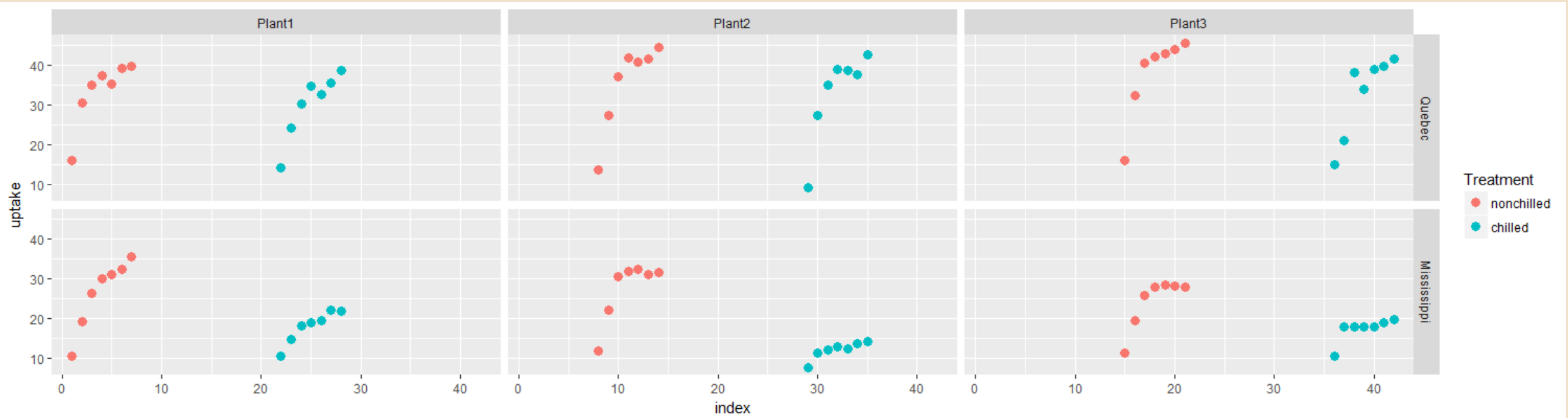
## Faceting : facet\_grid

## CODE

```

C02$index <- rep(1:42,2);C02$Plant_type <- substr(C02$Plant,3,3)
C02$Plant_type[C02$Plant_type==1] <- 'Plant1'
C02$Plant_type[C02$Plant_type==2] <- 'Plant2'
C02$Plant_type[C02$Plant_type==3] <- 'Plant3'
ggplot(C02,aes(x=index,y=uptake,color=Treatment))+geom_point(size=3)+
facet_grid(Type~Plant_type)

```



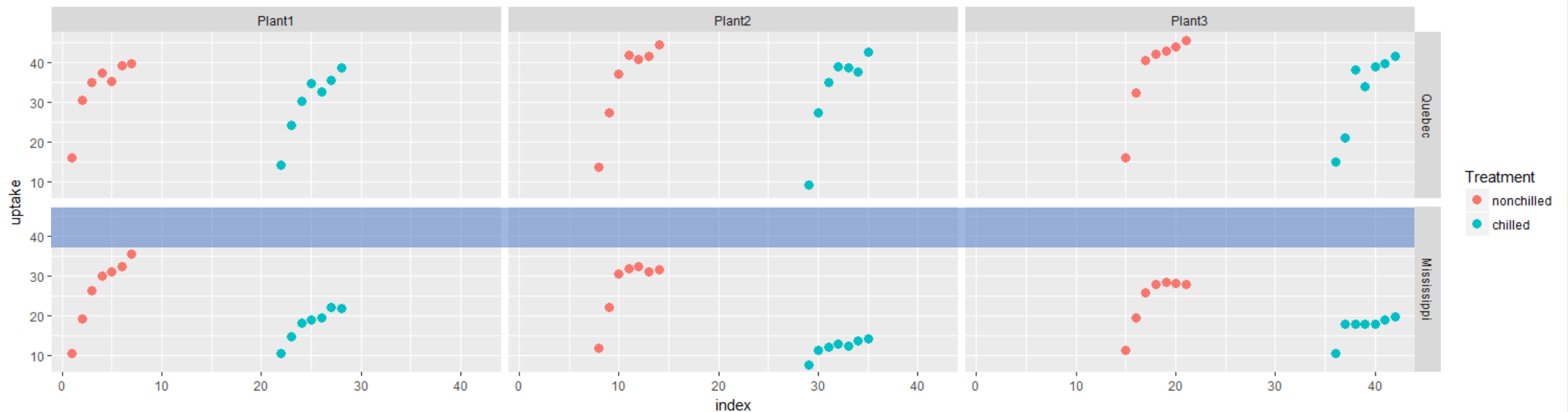
이때, 분할면의 이름을 바꾸고 싶다면 데이터를 직접 바꾸는 수밖에 없음  
 분할면 서식에 대한 더 많은 내용은 뒤의 theme 부분에서 한번에 전개될 예정

## 3

## Faceting : facet\_grid

## CODE

```
C02$index <- rep(1:42,2);C02$Plant_type <- substr(C02$Plant,3,3)
C02$Plant_type[C02$Plant_type==1] <- 'Plant1'
C02$Plant_type[C02$Plant_type==2] <- 'Plant2'
C02$Plant_type[C02$Plant_type==3] <- 'Plant3'
ggplot(C02,aes(x=index,y=uptake,color=Treatment))+geom_point(size=3)+
facet_grid(Type~Plant_type)
```



제일 값이 큰 면의 크기에 맞춰 다른 축들도 생성됨  
→ 크기가 작은 면에서 불필요한 공백 발생  
→ 면마다 축을 다르게 갖도록 지정해줄 수 있음



## 3

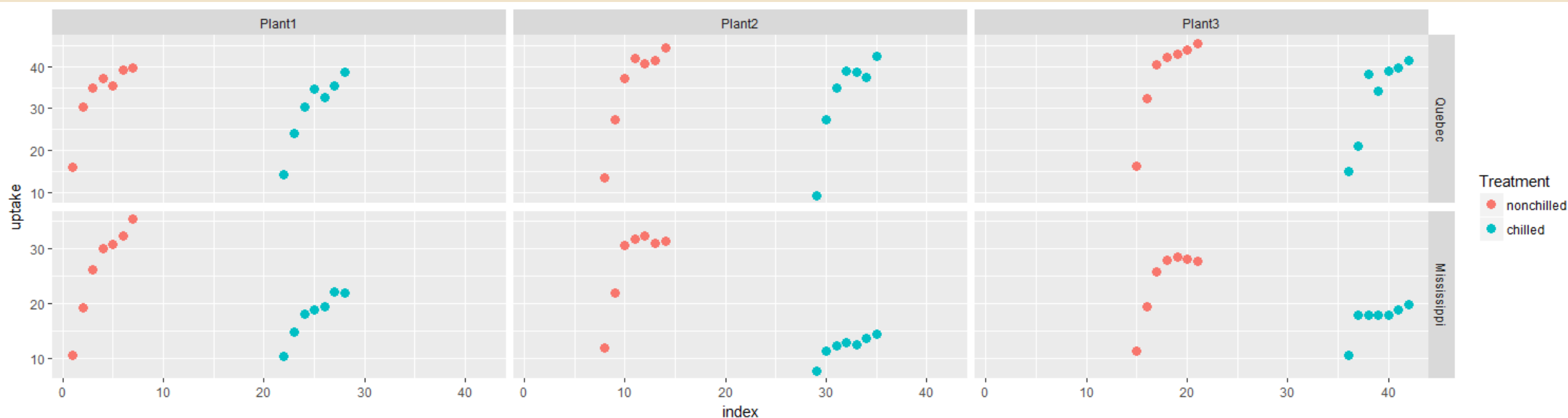
## Faceting : facet\_grid

## CODE

```

C02$index <- rep(1:42,2);C02$Plant_type <- substr(C02$Plant,3,3)
C02$Plant_type[C02$Plant_type==1] <- 'Plant1'
C02$Plant_type[C02$Plant_type==2] <- 'Plant2'
C02$Plant_type[C02$Plant_type==3] <- 'Plant3'
ggplot(C02,aes(x=index,y=uptake,color=Treatment))+geom_point(size=3)+
facet_grid(Type~Plant_type,scale='free_y')

```



facet\_grid 레이어 안에서 scale 변수를 조정해 줌으로써 가능  
: 'free\_x', 'free\_y', 'free' 중 하나 선택해서 사용

# 3 Geometry graphic option: scale\_A\_B

CODE

```
ggplot(iris,aes(x=Sepal.Width,y=Sepal.Length,color=Species))+geom_point()+  
scale_color_brewer(limits=c('virginica','setosa','versicolor'),palette='Set1')
```

scale\_A\_B(expression,...)

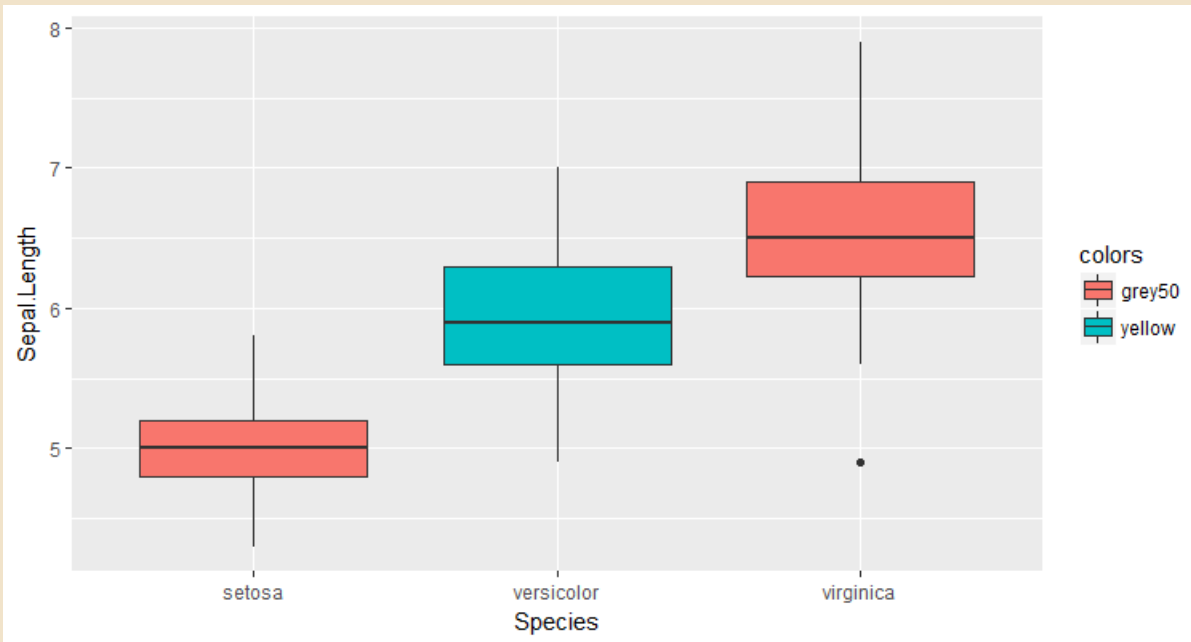
- A : 조정할 서식(aesthetic 안에서 설정한 서식)
  - ✓ alpha, color, fill, linetype, shape, size
- B : 조정할 방법(데이터 내의 값 사용 여부에 따라)
  - ✓ identity: 데이터 안에 있는 값을 그대로 사용하고자 할 때
  - ✓ manual: 내부의 값을 사용하지 않고 scale 레이어를 통해 직접 설정하고자 할 때
  - ✓ A가 color나 fill일때 그라데이션을 위한 B를 사용할 수 있음:
    - brewer(내장된 색상 조합을 사용)
    - gradient(시작 색과 끝 색을 지정하면 그 색 사이의 그라데이션을 색으로 사용)
    - grey(시작 값과 끝 값을 지정하면 흑백 그라데이션을 색으로 사용)

## 3

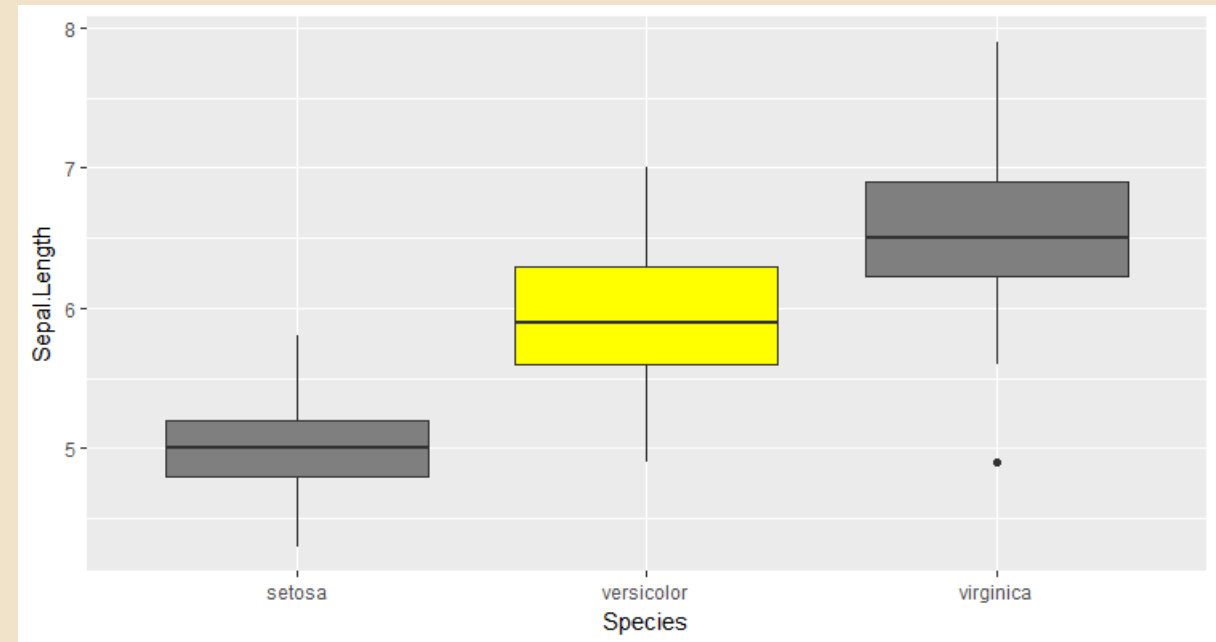
## Geometry graphic option : scale\_A\_identity

## CODE

```
iris$colors <- rep(c('grey50','yellow','grey50'),each=50)
ggplot(iris,aes(x=Species,y=Sepal.Length,fill=colors))+geom_boxplot()+
scale_fill_identity()
```



< scale\_fill\_identity layer를 추가하지 않았을 때 >



< scale\_fill\_identity layer를 추가했을 때>

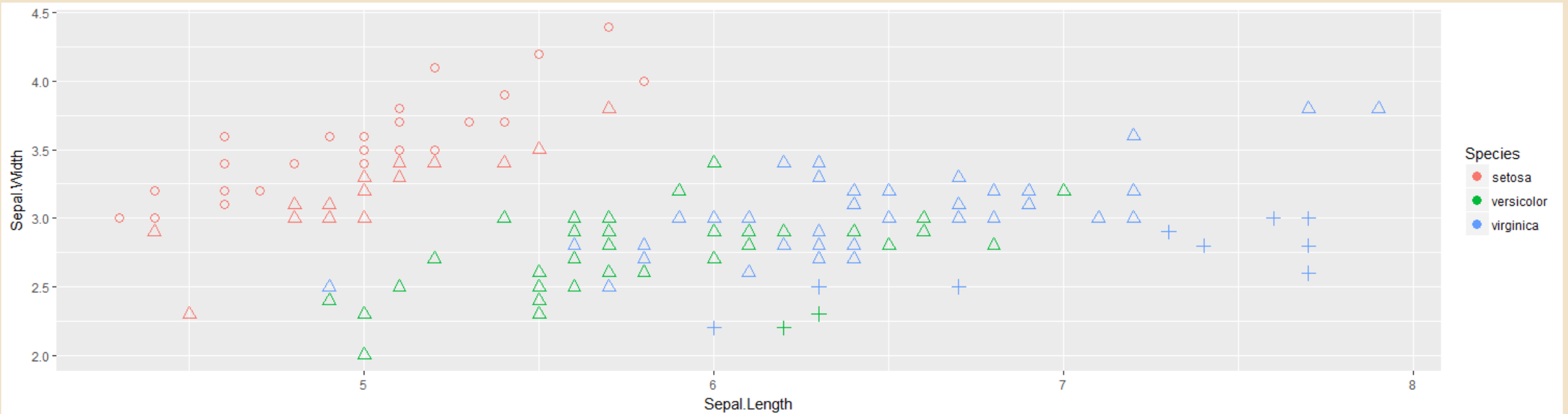
사용할 색 이름을 데이터 내부에 직접 입력하고, 데이터 내부에 있는 값을 직접 사용  
+. 다양한 색들과 색깔들 별 이름의 목록은 첨부한 Rcolor.pdf 참조!

## 3

## example : scale\_A\_identity

CODE

```
iris$Sepal.Ratio <- iris$Sepal.Length/iris$Sepal.Width
ggplot(iris,aes(x=Sepal.Length,y=Sepal.Width,shape=round(Sepal.Ratio,0),
color=Species))+geom_point(size=3)+scale_shape_identity()
```



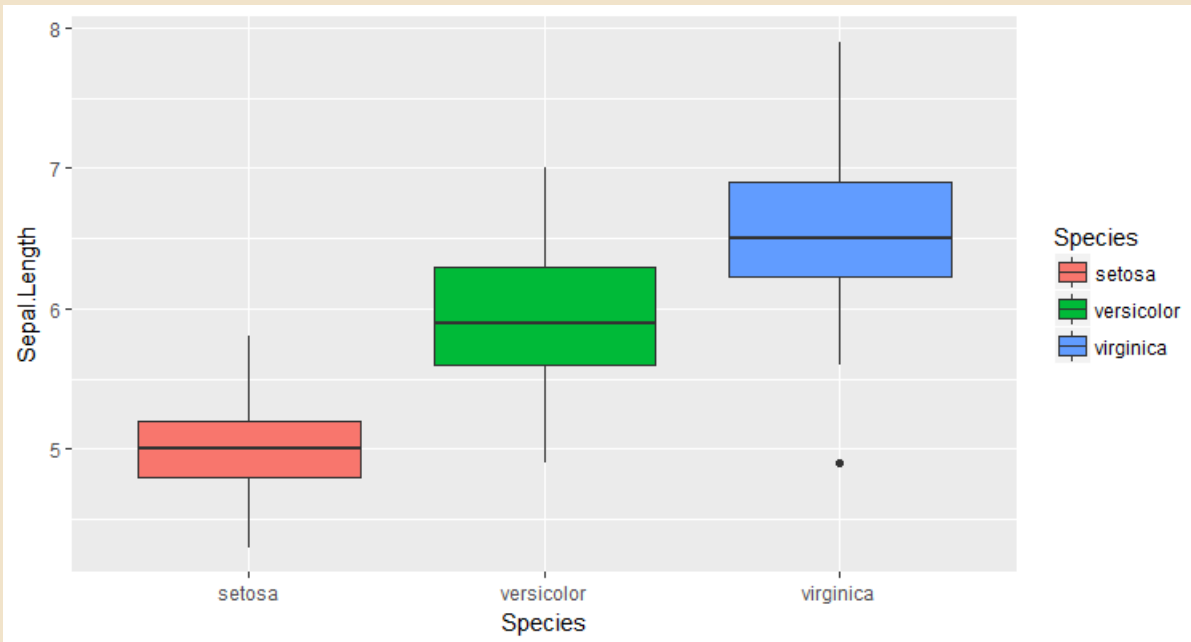
< 길이와 폭의 비를 반올림했을 때 1,2,3인 개체를 각각 동그라미, 세모, 더하기로 표시하고 이를 Species별로 관찰 >

**데이터 내부에서 값을 만들고 이를 바로 사용할 때 좋다.**

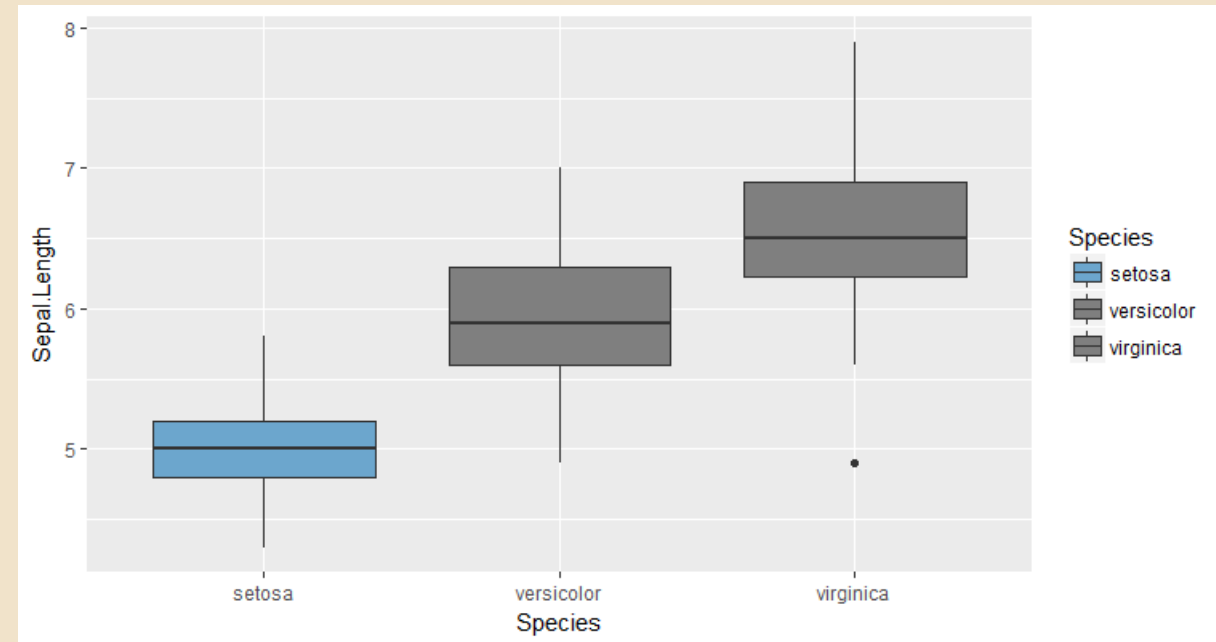
# 3 Geometry graphic option : scale\_A\_manual

CODE

```
ggplot(iris,aes(x=Species,y=Sepal.Length,fill=Species))+  
geom_boxplot()+scale_fill_manual(values=c('skyblue3','grey50','grey50'))
```



< scale\_fill\_manual layer를 추가하지 않았을 때 >



< scale\_fill\_manual layer를 추가했을 때>

Species별로 fill을 달리할 것이라고 aesthetic에서 명시한 다음  
scale\_fill\_manual에서 Species마다 적용할 fill을 values안에 명시

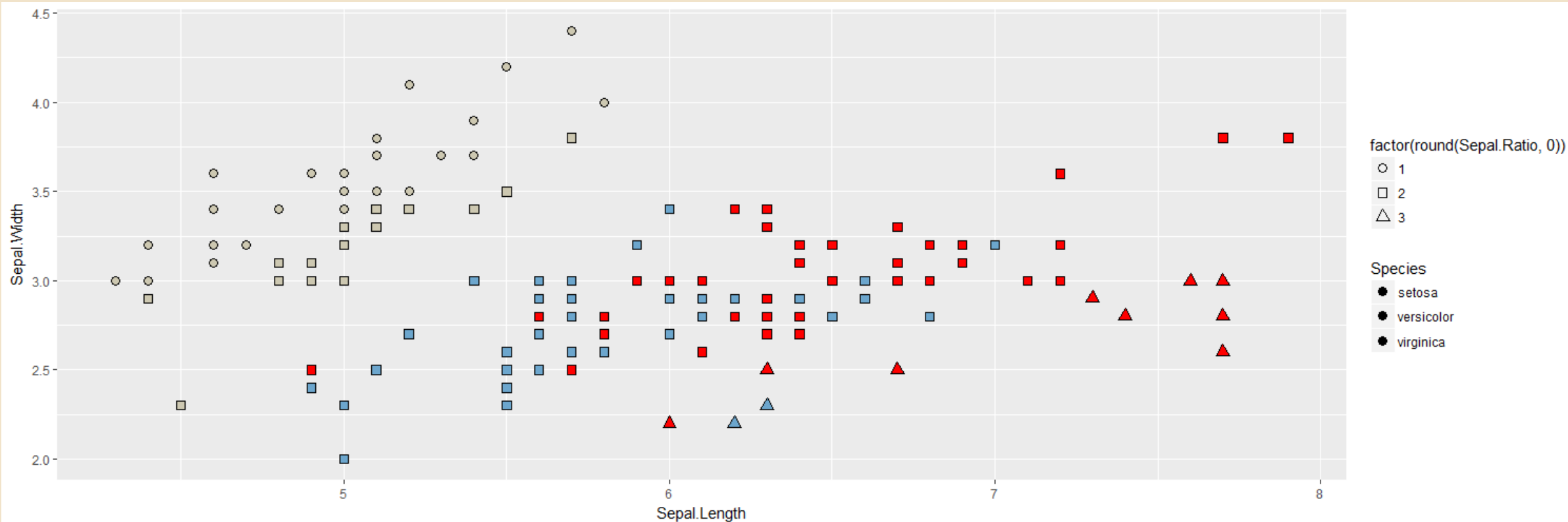


## 3

## example : scale\_A\_manual

## CODE

```
iris$Sepal.Ratio <- iris$Sepal.Length/iris$Sepal.Width
ggplot(iris,aes(x=Sepal.Length,y=Sepal.Width,shape=factor(round(Sepal.Ratio,0)),fill=Species))+
  geom_point(size=3)+scale_shape_manual(values=c(21,22,24)) +
  scale_fill_manual(values=c('cornsilk3','skyblue3','red'))
```



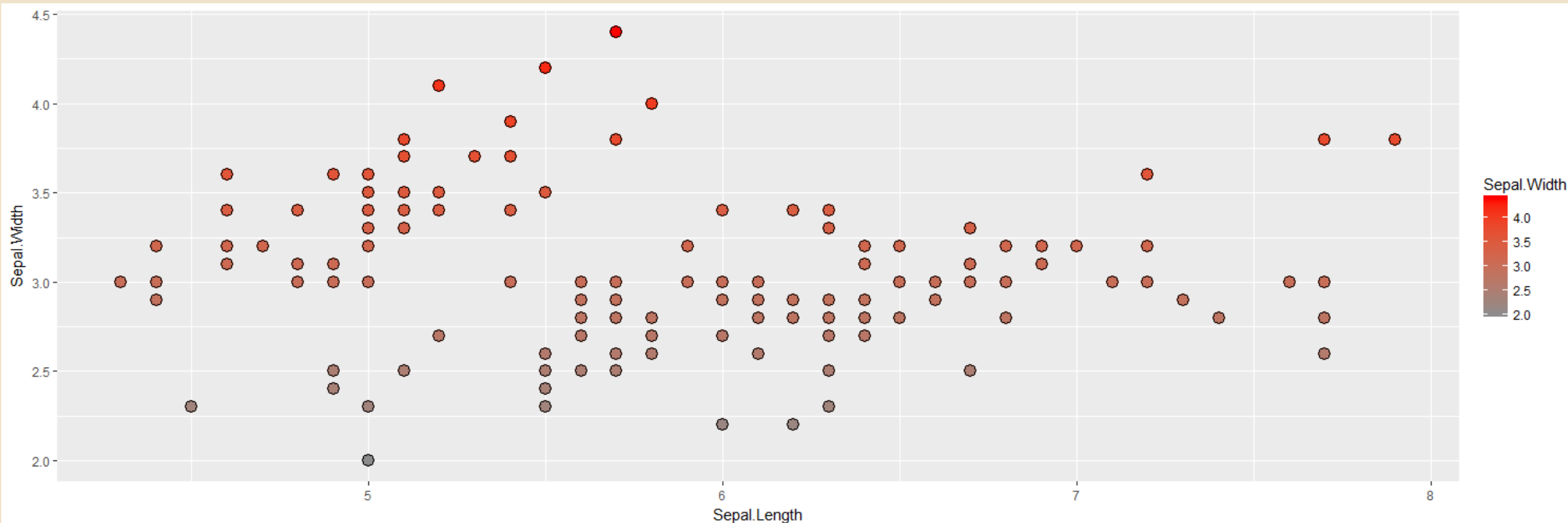
그래픽의 외형을 좀 더 세밀하게 조정할 때 좋다

## 3

## Geometry graphic option: scale\_color(fill)\_gradient

CODE

```
ggplot(iris,aes(x=Sepal.Length,y=Sepal.Width,fill=Sepal.Width))+  
geom_point(shape=21,size=4)+scale_fill_gradient(low='grey55',high='red')
```



**연속형 변수(Sepal.Width)의 색상을 그라데이션으로 나타낼 때 사용**

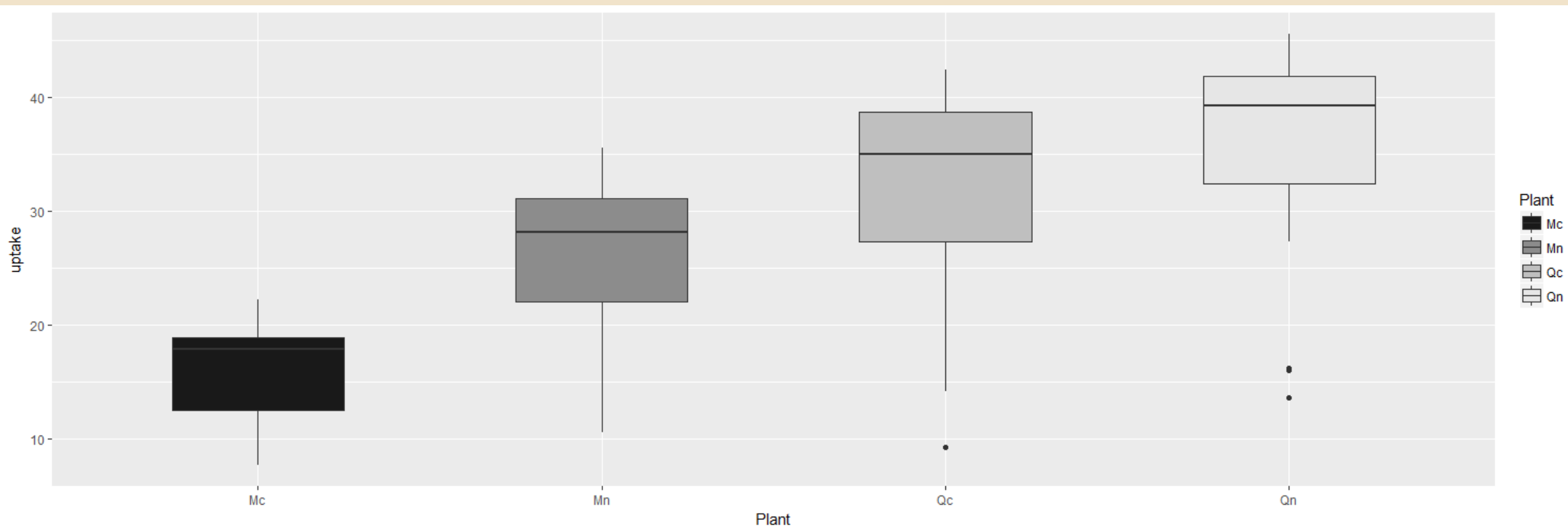
fill 기준변수로 설정된 Sepal.Width의 값이 낮을수록 회색에 가까운 fill이(low='grey55'),  
높을수록 빨간색에 가까운 fill이(high='red') 나타난다.

## 3

## Geometry graphic option: scale\_color(fill)\_grey

## CODE

```
co2 <- data.frame(index=rep(1:7,12),Plant=substr(as.character(CO2$Plant),1,2),uptake=CO2$uptake)
ggplot(co2,aes(x=Plant,y=uptake,fill=Plant))+geom_boxplot(width=0.5)+
scale_fill_grey(start=0.1,end=0.9)
```



**이산형 변수(Plant)의 색상을 흑백 그라데이션으로 나타낼 때 사용**

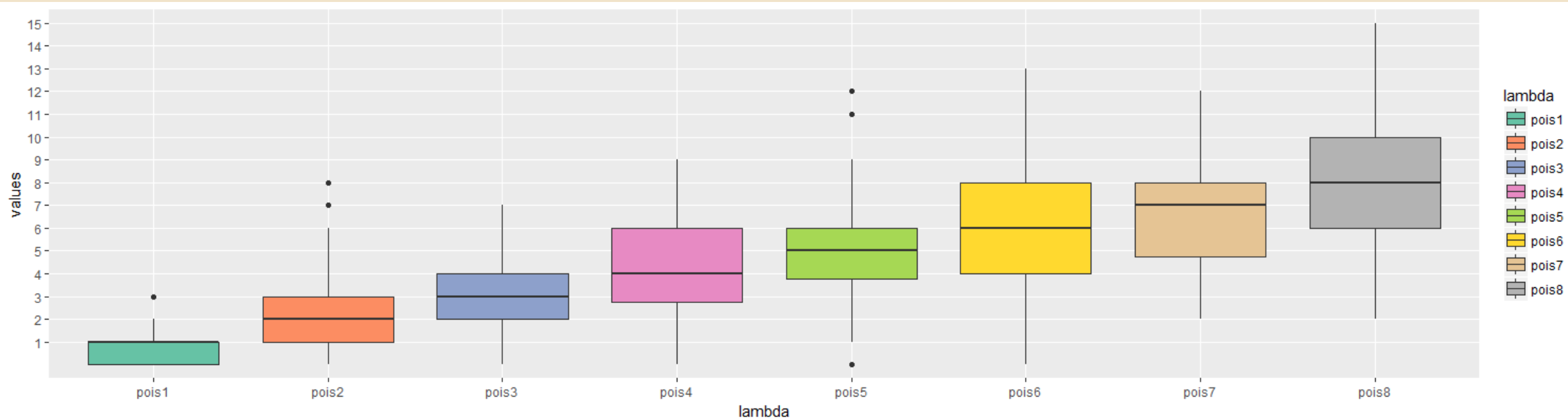
fill 기준변수로 설정된 Plant별로 0.1의 명도에서 시작(start)해  
0.9로 끝나는(end) 흑백 그라데이션을 부여

## 3

## Geometry graphic option: scale\_color(fill)\_brewer

## CODE

```
pois_data <- data.frame(pois1=rpois(100,1),pois2=rpois(100,2),pois3=rpois(100,3),pois4=rpois(100,4),
pois5=rpois(100,5),pois6=rpois(100,6),pois7=rpois(100,7),pois8=rpois(100,8))
library(tidyr); pois_data <- gather(pois_data,lambda,values,pois1:pois8)
ggplot(pois_data,aes(x=lambda,y=values,fill=lambda))+geom_boxplot()+
scale_y_discrete(limits=1:15)+scale_fill_brewer(palette='Set2')
```

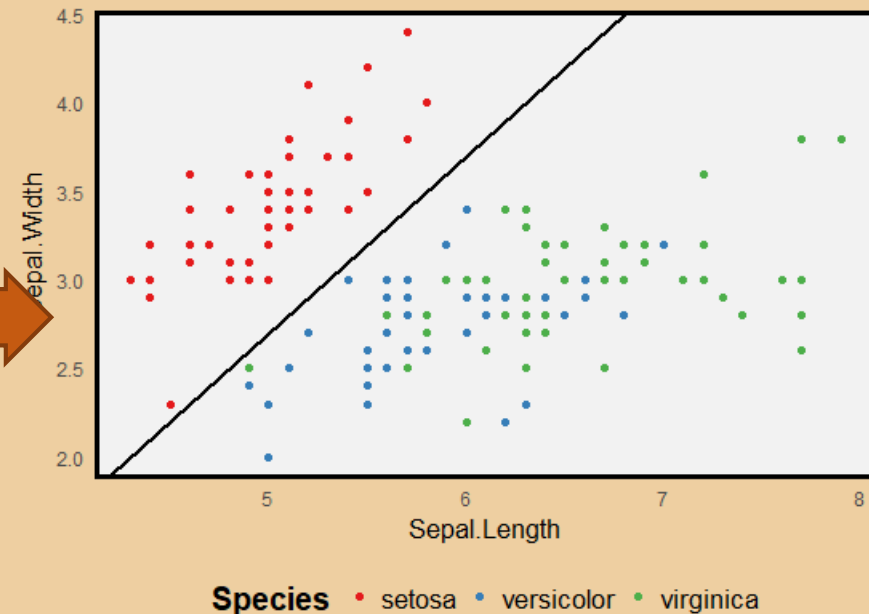
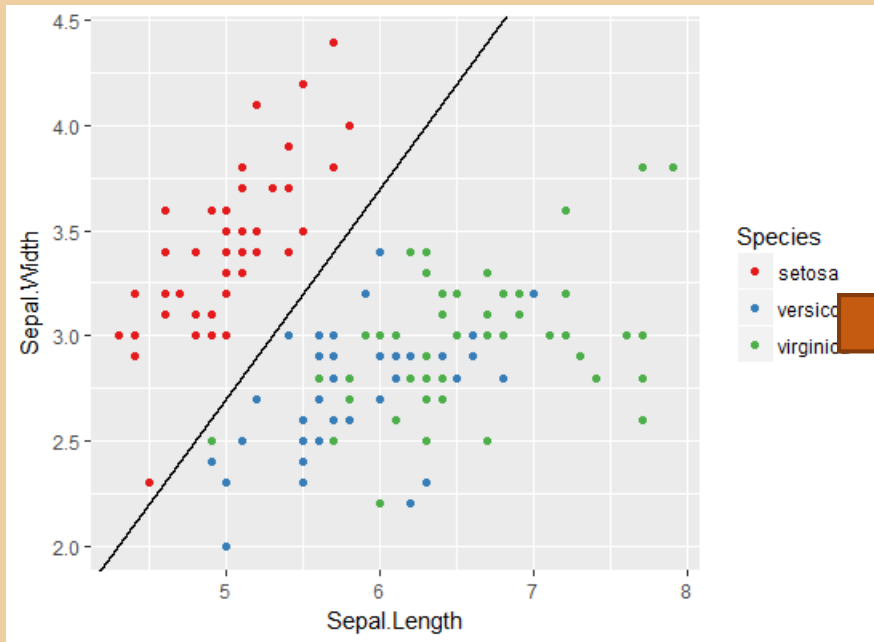


이산형 변수(Plant)의 색상을 R 내장 색상조합(Set2)으로 나타낼 때 사용  
다른 R 내장 색상조합이 궁금하다면 부록을...

# 3

## Extra-Geometry graphic handling: theme

theme layer



Geometry : 산점도(`geom_point`)

→ **Geometry 외부 요소 : 이 플랏에서 산점도 말고 다른 모든 것**

(플랏 배경색, 패널 배경색, 패널 격자 선, 축 제목 크기, 축 범주 텍스트 크기, 범례 위치, 범례 배경색 등등)

→ **theme layer는 Geometry 외부 요소에 대한 서식을 지정하는 레이어!**



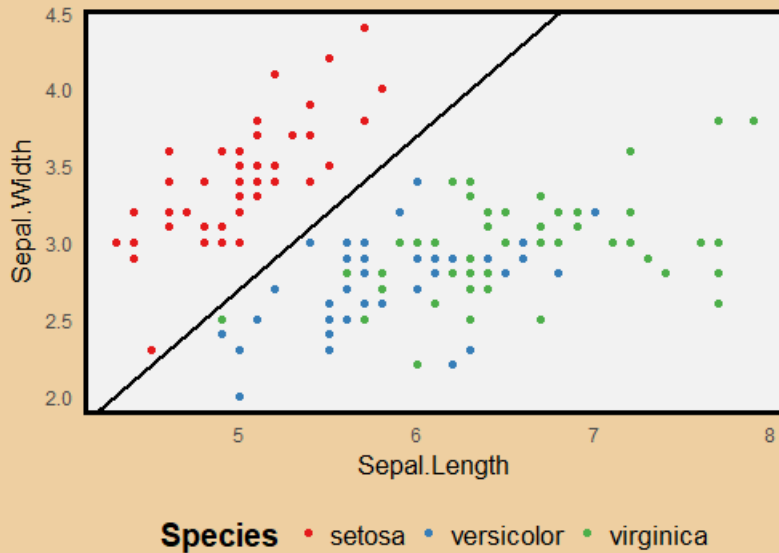
## theme grammar

**theme(element=element\_type(expression))**

- **element** : 지정할 geometry 외부요소
- **element\_type** : 그 외부요소의 종류
  - ✓ `element_blank()` : 그 외부요소를 지울 때 사용
  - ✓ `element_line()` : 그 외부요소가 선일 때 사용
  - ✓ `element_text()` : 그 외부요소가 텍스트일 때 사용
  - ✓ `element_rect()` : 그 외부요소가 사각형일 때 사용
  - ✓ `unit()` : 그 외부요소가 길이일 때 사용
- **expression** : 그 요소에 대한 서식을 지정할 표현

# 3

## Extra-Geometry graphic handling: theme



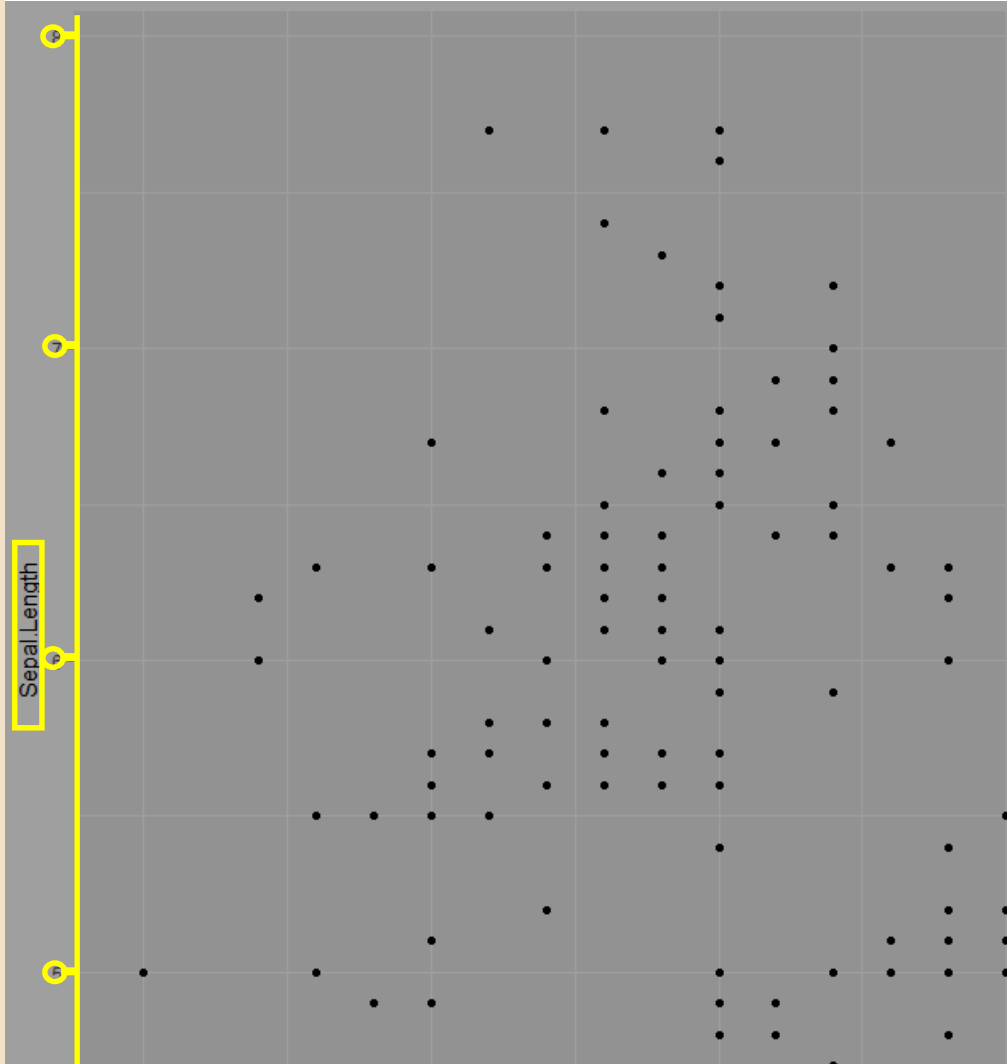
```
ggplot(iris,aes(x=Sepal.Length,y=Sepal.Width,color=Species))+geom_point()+  
geom_abline(intercept=-2.3,slope=1,size=1,color='black')+  
scale_color_brewer(limits=c('setosa','versicolor','virginica'),palette='Set1')+  
theme(panel.background = element_rect(fill='grey95',color='black',size=2),  
panel.grid = element_blank(),  
plot.background = element_rect(fill='navajowhite2'),  
axis.ticks = element_blank(),  
legend.position = 'bottom',  
legend.background = element_rect(fill='navajowhite2'),  
legend.key = element_rect(fill='navajowhite2',color='navajowhite2'),  
legend.title = element_text(size=15,face = 'bold'),  
legend.text = element_text(size=13),  
axis.title = element_text(size=13))
```

\* 앞선 슬라이드에서 theme layer의 효과를 보이기 위해 사용한 plot을 생성한 코드.  
theme layer가 element, element type, expression으로 이루어져 있음을 보여준다.

지정할 element의 이름과 그 element의 type만 알면  
theme 레이어를 사용할 수 있다!

# 3

## theme: axis



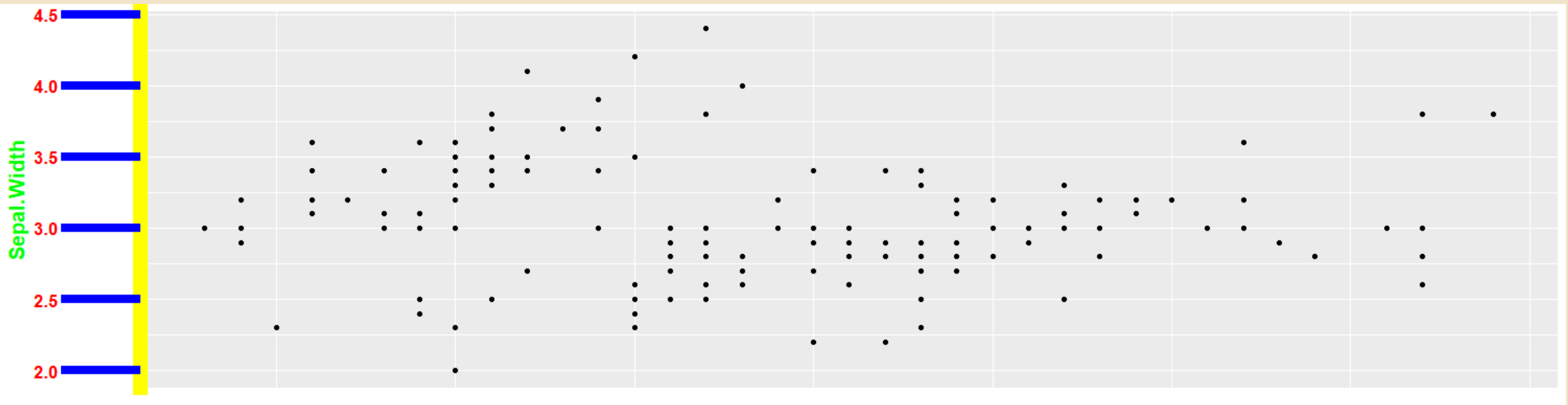
- `axis.line.x(or y)` : `x(or y)`축 선을 지목
- `axis.line` : `x,y`축 선들을 한번에 지목  
→ `element_line()`, `element_blank()` 사용
- `axis.ticks.x(or y)` : `x(or y)`축의 눈금을 지목
- `axis.ticks` : `x,y`축 눈금들을 한번에 지목  
→ `element_line()`, `element_blank()` 사용
- `axis.ticks.length` : `x,y`축 눈금의 길이를 설정  
→ `unit()` 사용
- `axis.text.x(or y)` : `x(or y)`축 글자를 지목
- `axis.text` : `x,y`축 글자들을 한번에 지목  
→ `element_text()`, `element_blank()` 사용
- `axis.title.x(or y)` : `x(or y)`축의 제목을 지목
- `axis.title` : `x,y`축 제목들을 한번에 지목  
→ `element_text()`, `element_blank()` 사용

## 3

## example: theme(axis)

## CODE

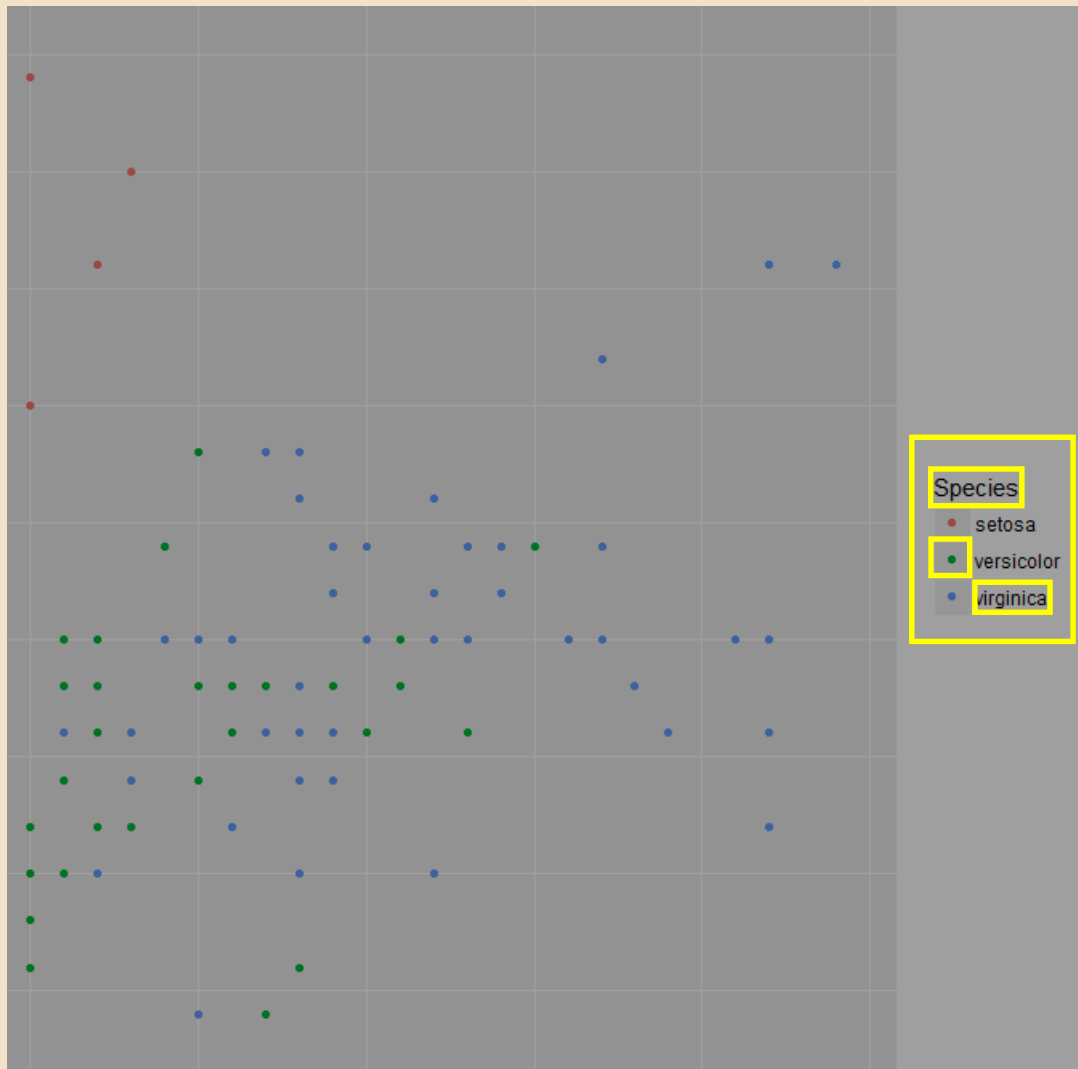
```
ggplot(iris,aes(x=Sepal.Length,y=Sepal.Width))+geom_point()+  
  theme(axis.title.y = element_text(size=15,color='green',face='bold'),  
        axis.text.y = element_text(size=13,face='bold',color='red'),  
        axis.line.y = element_line(size=5,color='yellow',lineend = 'square'),  
        axis.ticks.x = element_blank(),axis.title.x = element_blank(),axis.text.x = element_blank(),  
        axis.ticks.y = element_line(size=3,color='blue'),  
        axis.ticks.length = unit(2,'cm'))
```



서식 색깔을 element별로 달리해서 element가 어느 부분에 있는 요소인지를 나타냈음

# 3

## theme: legend



- **legend.background** : 범례의 배경을 지목  
→ `element_rect()`, `element_blank()` 사용
- **legend.key** : 범례 항목의 배경을 지목  
→ `element_rect()`, `element_blank()` 사용
- **legend.box.background** : 범례가 2개 이상일때, 범례 영역 전체의 배경을 지목  
→ `element_rect()`, `element_blank()` 사용
- **legend.title** : 범례의 제목을 지목  
→ `element_text()`, `element_blank()` 사용
- **legend.text** : 범례 항목 텍스트를 지목  
→ `element_text()`, `element_blank()` 사용

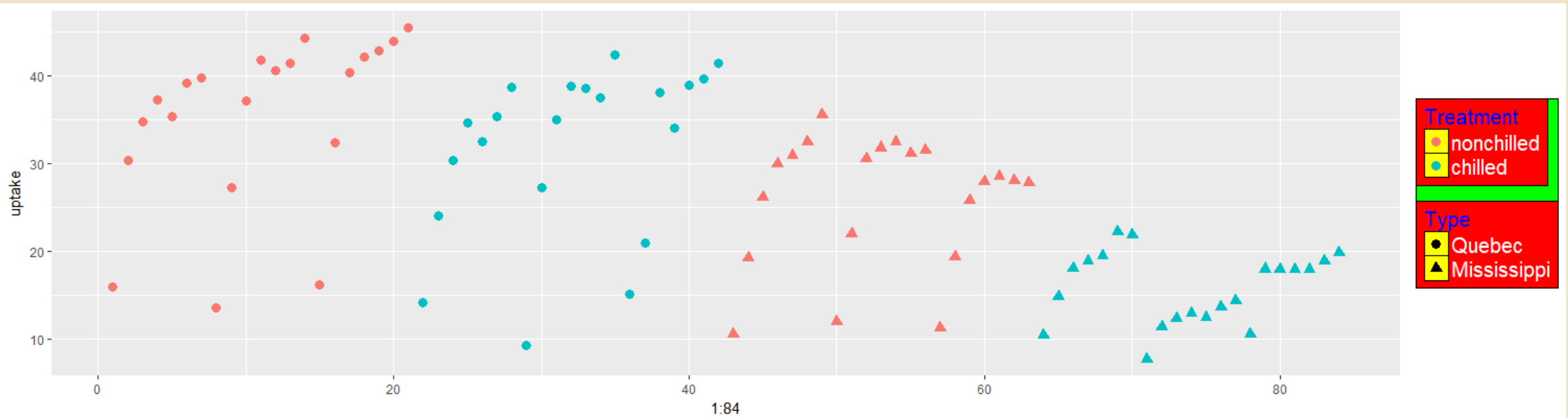


## 3

## example: theme(legend)

## CODE

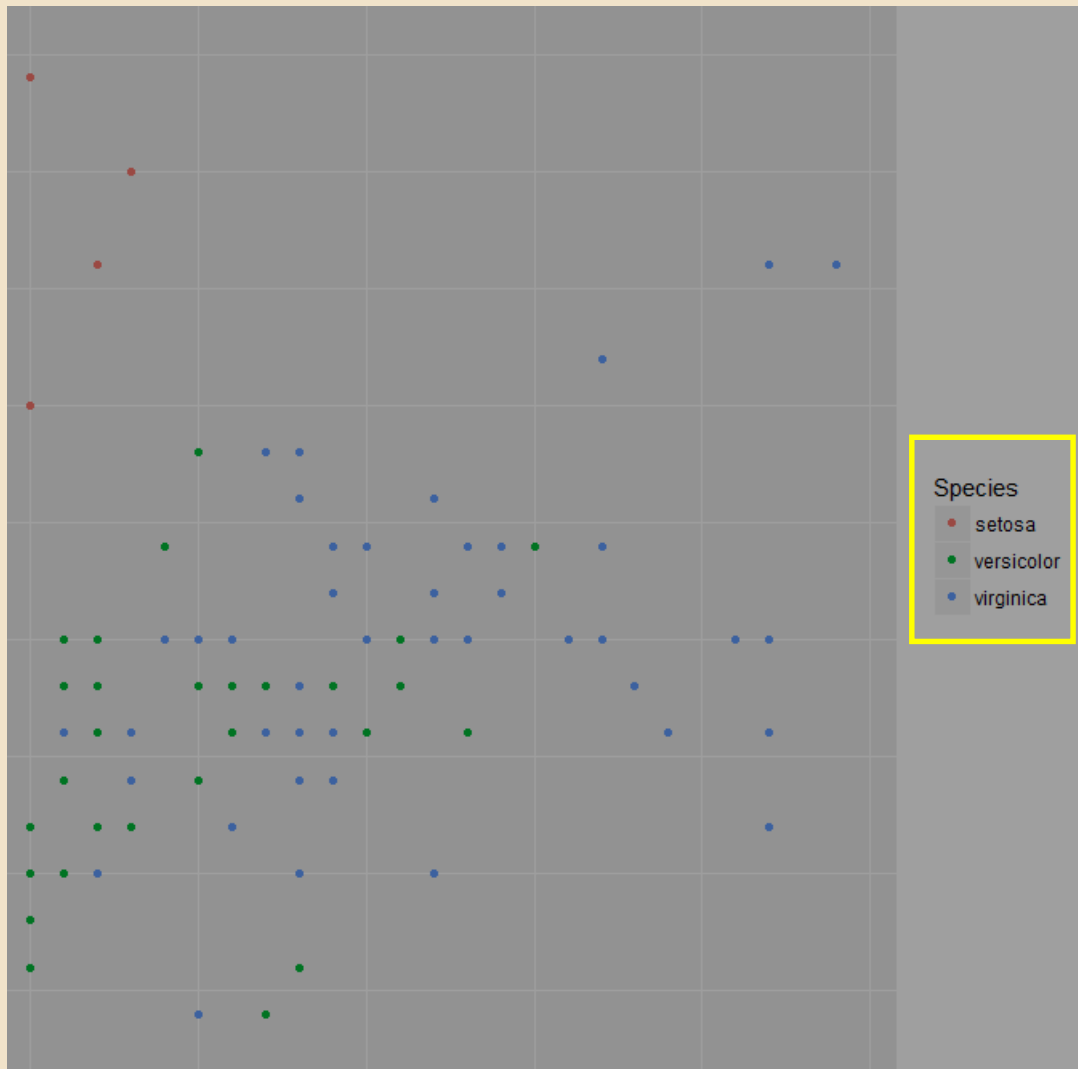
```
ggplot(CO2,aes(x=1:84,y=uptake,shape=Type,color=Treatment))+geom_point(size=3)+
  theme(legend.background = element_rect(fill='red',color='black'),
        legend.key = element_rect(fill='yellow',color='black'),
        legend.box.background = element_rect(fill='green'),
        legend.title = element_text(size=15,color='blue'),
        legend.text = element_text(size=15,color='white'))
```



서식 색깔을 element별로 달리해서 element가 어느 부분에 있는 요소인지를 나타냈음

# 3

## theme: legend



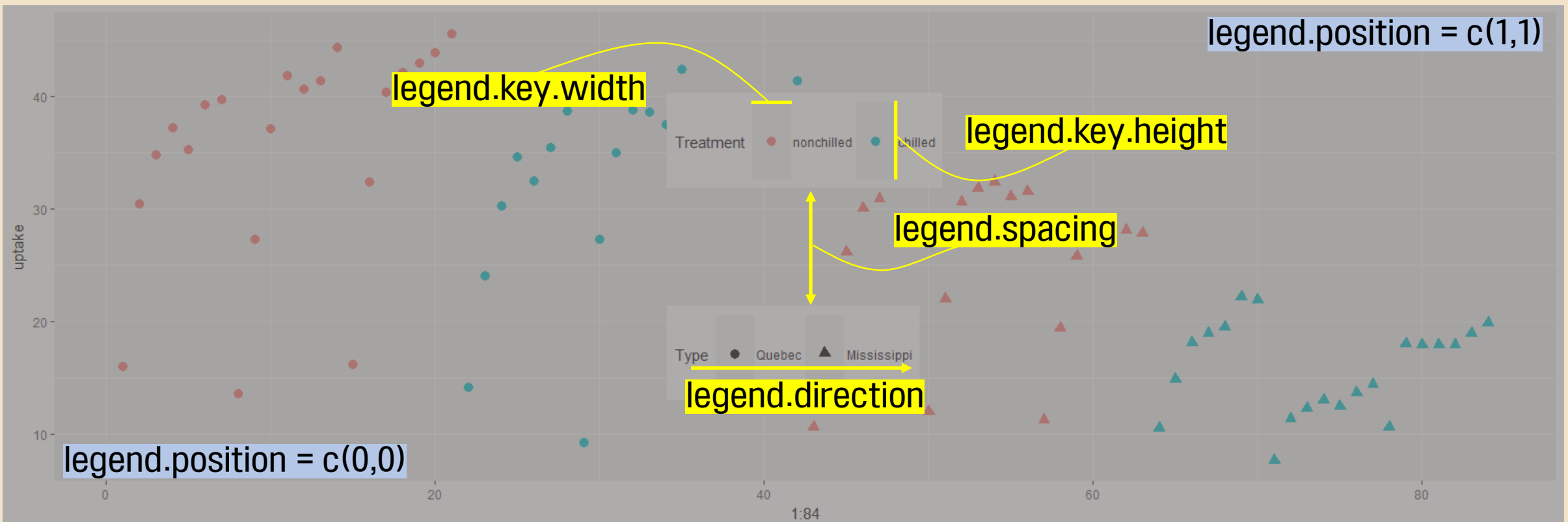
- **legend.position** : 범례의 위치  
→ 'top', 'bottom', 'left', 'right' 중 사용하면 플랏 외부  
→ 좌측 최하단을 (0,0), 우측 최상단을(1,1)로 하고,  
그 사이의 좌표를 입력하여 플랏 내부에 범례를  
위치시킬 수도 있음
- **legend.direction** : 범례 작성 방향  
→ 'horizontal', 'vertical' 둘 중 하나
- **legend.key.height** : 범례 항목 박스 높이  
→ unit 사용
- **legend.key.width** : 범례 항목 박스 폭  
→ unit 사용
- **legend.box.spacing** : 플랏과 범례와의 거리  
→ unit 사용
- **legend.spacing** : 범례가 두 개 이상일때 범례 간 거리  
→ unit 사용

## 3

## example: theme(legend)

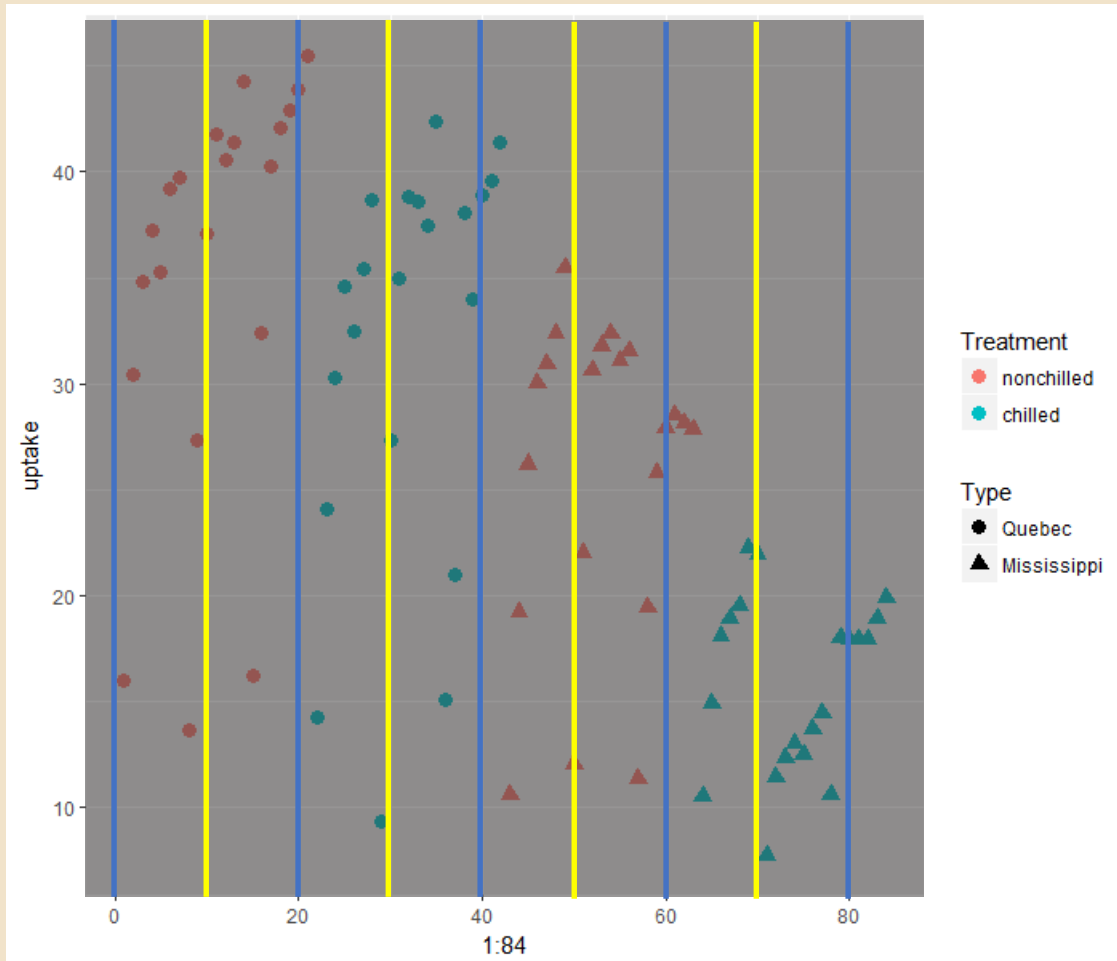
## CODE

```
ggplot(CO2,aes(x=1:84,y=uptake,shape=Type,color=Treatment))+geom_point(size=3)+
  theme(legend.position = c(0.5,0.5),
        legend.direction = 'horizontal',
        legend.key.height = unit(2,'cm'),
        legend.key.width = unit(1,'cm'),
        legend.spacing = unit(3,'cm'))
```



## 3

## theme: panel



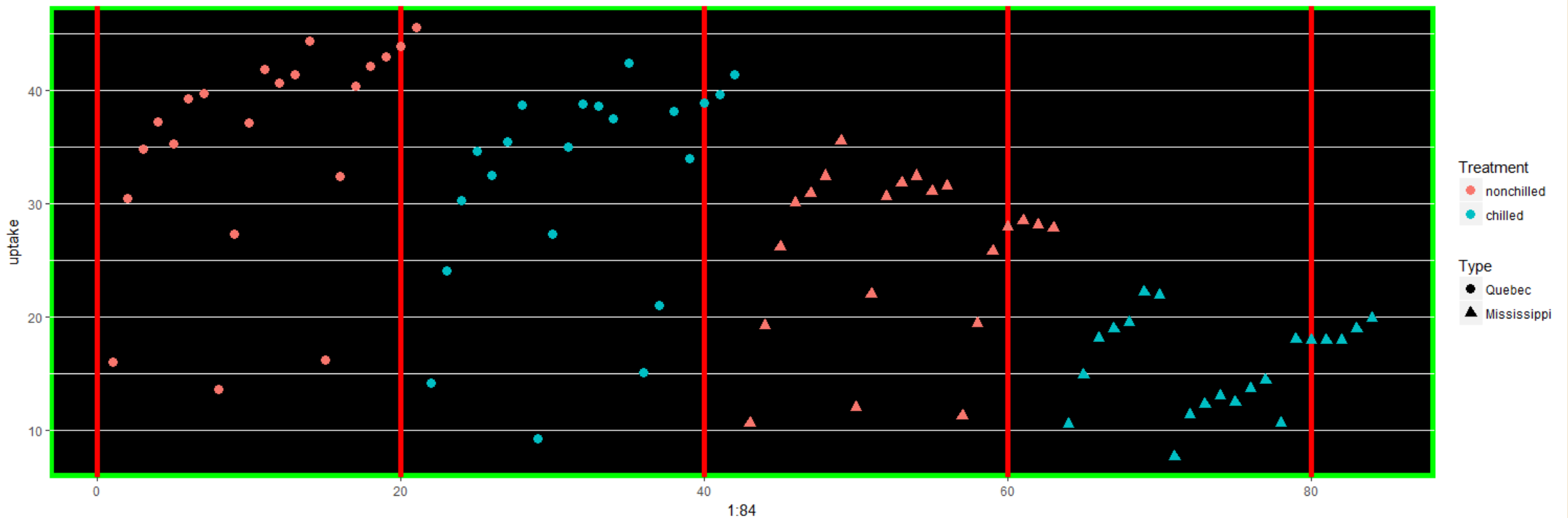
- **panel.background** :  
 → geometry가 그려진 바탕(panel)의 사각형을 지칭  
 → element\_rect(), element\_blank() 사용
- **panel.grid.major.x(y)** :  
 → x(y)축의 major grid(파란색 선)를 지칭  
 → element\_line(), element\_blank() 사용
- **panel.grid.minor.x(y)** :  
 → x(y)축의 minor grid(노란색 선)를 지칭  
 → element\_line(), element\_blank() 사용

## 3

## example: theme(panel)

## CODE

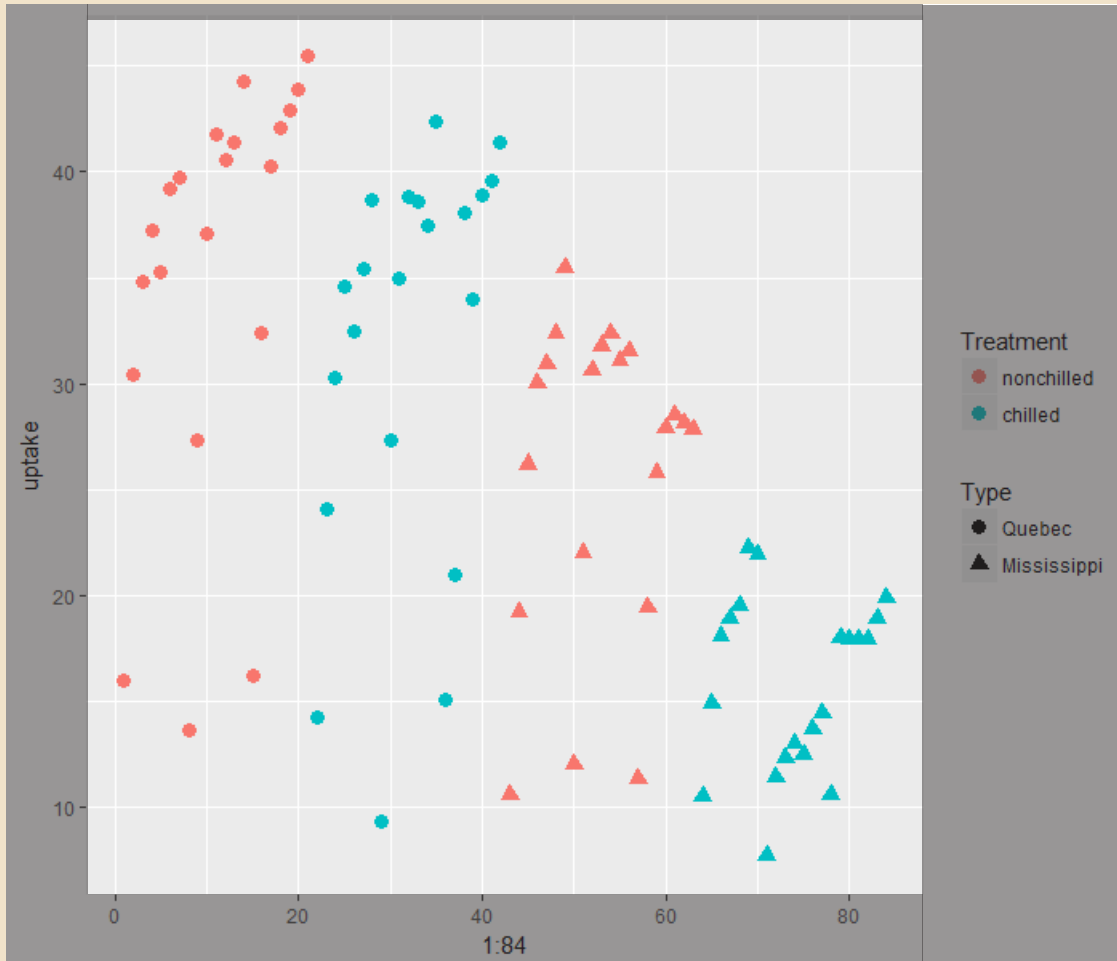
```
ggplot(CO2,aes(x=1:84,y=uptake,shape=Type,color=Treatment))+geom_point(size=3)+
  theme(panel.background = element_rect(fill='black',color='green',size=3),
        panel.grid.major.x = element_line(color='red',size=2),
        panel.grid.minor.x = element_blank())
```



x축의 minor grid가 element\_blank()로 인해 사라졌음

# 3

## theme: plot



- `plot.background` :
  - plot이 그려진 바탕의 사각형을 지칭
  - `element_rect()`, `element_blank()` 사용
- `ggtitle()` :
  - plot의 제목을 만들 때 사용
- `plot.title` :
  - 제목의 서식을 수정할 때 사용
  - `element_text()`, `element_blank()` 사용



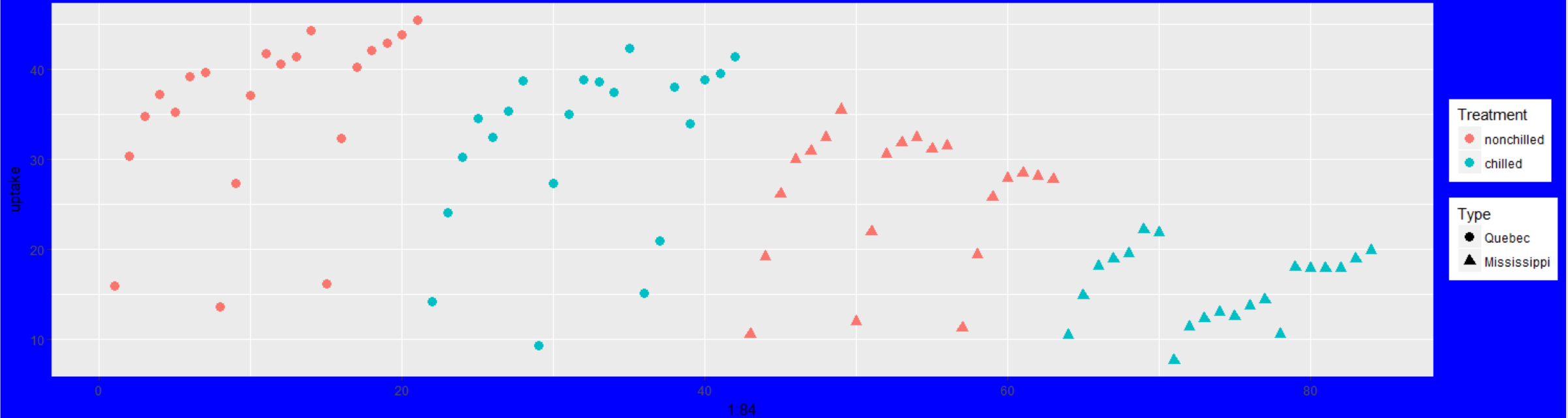
## 3

## example: theme(plot)

## CODE

```
ggplot(CO2,aes(x=1:84,y=uptake,shape=Type,color=Treatment))+geom_point(size=3)+  
ggtitle('P-SAT plot')+  
theme(plot.background = element_rect(fill='blue'),  
      plot.title = element_text(color='white',size=20))
```

P-SAT plot



theme 레이어 앞에서 ggtitle를 명시해 제목을 생성하고  
이에 대한 서식을 theme 레이어 안에서 수행



# + ADDITIONAL NOTES







# Details: aesthetic

## Some aesthetic options

x	x축에 사용할 column
y	y축에 사용할 column
color	범주별로 color를 달리할 column(그래픽의 테두리-선-의 색을 의미)
linetype	범주별로 linetype를 달리할 column(실선, 점선같은 선의 모양을 의미)
shape	범주별로 shape를 달리할 column(점의 모양을 의미)
size	범주별로 size를 달리할 column(그래픽 크기를 의미: 점의 크기, 선의 굵기 등)
fill	범주별로 fill을 달리할 column(그래픽 테두리 내부의 색을 의미)
alpha	범주별로 alpha를 달리할 column(그래픽 투명도를 의미)
group	group별로 그래픽을 생성하고 싶을 때, group의 범주를 지정할 column (주로 x나 y가 연속형이지만 어느 범주로 구분될 필요가 있을 때 사용)

\*geometry내 expression으로 사용해도 그 의미는 같음. 단지 범주별 서식이 이루어지지 않을 뿐임.



# Details: shape aesthetic

0 	1 	2 	3 	4 
5 	6 	7 	8 	9 
10 	11 	12 	13 	14 
15 	16 	17 	18 	19 
20 	21 	22 	23 	24 
	25 			

fill aesthetic도 적용할 수 있는 shape



# Details: R Palettes

