데이터 분석 기초 - dplyr과 ggplot2를 이용한 데이터 분석

환경생태데이터사이언스 실습 October 1, 2019

오늘의 학습 목표

dplyr과 ggplot2를 이용한 데이터 분석

1. csv 및 excel 형식의 데이터 읽기
2. 변수명 (열이름) 바꾸기
3. 조건에 맞는 데이터 추출
4. 데이터 정렬
5. 파생변수 (새로운 열) 만들기
6. 집단 별 요약
7. 데이터 합치기
8. 결측치 제거
9. ggplot2를 이용한 그래프

김영우 (2017) 쉽게 배우는 R 데이터 분석. 이지스퍼블리싱 (Chapter 5, 6, 7, 8)

데이터 읽기

기본 패키지로 데이터 읽고 쓰기

- R에서 기본 함수로 테이블을 읽는 함수는?
- R에서 가장 많이 다뤄지는 외부 데이터인 csv 파일을 읽을 때 이용되는 함수는?
- 데이터 프레임을 저장하는 함수는?
- 데이터 프레임을 R에서 가장 많이 다뤄지는 외부 데이터인 csv 파일로 저장할 때 쓰는 함수는?
- 실습: 이번 주에 제공된 데이터를 읽어보세요.

데이터 확인 기초

- 읽은 외부 파일의 형식을 확인하는 함수는?
- 읽은 외부 파일의 일부 행을 확인하는 함수는?
- 실습: 앞에서 읽은 데이터의 형식을 확인하세요.

엑셀 형식의 데이터 읽기

"readxl" package의 read_excel 함수를 이용하여 excel 파일 읽기

```
# install.packages("readxl", repos="cran.seaoul.go.kr")
library(readxl)
Weather <- read_xls("./Data/KMA_20190916_0922.xls")
FineDust <- read_xls("./Data/PM25_20190930.xls",
                     na="-", col_types=c("text", "text",
                                         rep("numeric", 24)))
```

readr 패키지를 이용하여 데이터 읽기

readr 패키지를 이용하면 데이터를 읽으면서 변수형을 지정할 수 있다..

```
library(readr)
Weather <- read_csv("./Data/KMA_20190916_0922.csv",
                    col_types=list("f", "c", "d", "d", "d", "d") )
Weather$Time <- as.POSIXct(Weather$Time,
                           format=c("%Y-%m-%d %H:%M"))
```

dplyr을 이용한 데이터 정제 및 가공

열 이름 바꾸기

기본 내장 함수 이용: `colnames(Weather)[1] <- c("New_Name")`

dplyr 패키지를 이용: `rename(Weather, New_Name = Old_Name)`

```
# Using internal basic function
FineDust_1 <- FineDust
# colnames(FineDust_1)
colnames(FineDust_1)[3] <- c("T1")
colnames(FineDust_1)[4:ncol(FineDust_1)] <-
    paste("T",
          gsub("□", "", colnames(FineDust)[4:ncol(FineDust)]), sep="")

# Using dplyr package
library(dplyr)
FineDust_1 <- rename(FineDust_1, Type = □□□,
                     Location = □□□□)
```

조건에 맞는 데이터 추출

dplyr 패키지의 filter와 select * filter: 원하는 조건에 맞는 행 추출 * select: 원하는 변수 (열) 추출

```
FineDust_f <- FineDust_1 %>% filter(!is.na(T2))
FineDust_fs <-  FineDust_1 %>%
                filter(!is.na(T1+T7+T13+T18)) %>%
                select(Location, T1, T7, T13, T18)
FineDust_f2 <-  FineDust_1 %>% filter(T18 > 30 & T18 < 40) %>%
                select(Location, T18)
```

데이터 정렬

dplyr 패키지의 arrange 함수 이용

```
# arrange table with increasing order
FineDust_fa <-  FineDust_1 %>%
                filter(T18 > 30 & T18 <40) %>%
                arrange(T18) %>%
                select(Location, T18)
# arrange table with decreasing order
FineDust_fa_desc <-  FineDust_1 %>%
                filter(T18 > 30 & T18 <40) %>%
                arrange(desc(T18)) %>%
                select(Location, T18)
# arrange table with increasing order of two variables
FineDust_fa_multi <- FineDust_1 %>%
                     filter(T18 > 30 & T18 <40) %>%
                     arrange(T1, T18) %>%
                     select(Location, T1, T18)
```

파생 변수 (새로운 열) 추가하기

dplyr 패키지의 mutate 함수를 이용하여 열을 추가할 수 있다.

```
FineDust_rush <- FineDust_1 %>%
                 filter(!is.na(T7+T8+T9)) %>%
                 mutate(Sum = T7+T8+T9) %>%
                 select(Location, T7, T8, T9, Sum) %>%
                 arrange(desc(Sum))
```

지금 구한 FineDust_rush는 무엇을 의미할까요?

집단 별 요약

집단 별 요약은 dplyr의 group_by와 summarize를 이용하여 얻을 수 있다.

```
# Get the internal data file from ggplot2
mpg <- ggplot2::mpg
# Check the mpg data using head(mpg); str(mpg)
# From mpg data, we will calculate average city fuel efficiency of ea
# And display them decreasing order
mpg_new <-  mpg %>%
            group_by(manufacturer) %>%
            summarize(mean_cty = mean(cty)) %>%
            arrange(desc(mean_cty))
# Do the same job for highway fuel efficiency (hwy) of each car maker
mpg_count <- mpg %>%
             group_by(manufacturer) %>%
             summarize(number = n())
```

데이터 합치기

dplyr 패키지의 left_join과 bind_rows 함수를 이용하여 두 개의 데이터를 합칠 수 있다.

bind_rows는 rbind와 사용법 및 결과가 동일하다 (주의 변수 명 곧 열 이름이 같아야 한다).

```
# create data frame about car makers and their nationality
UniqueCarMakers <-  mpg %>%
                    mutate(nationality = ifelse(select(., manufacture
                    group_by(manufacturer) %>%
                    summarize(nationality = unique(nationality))
NatOfCar <- data.frame("manufacturer" = UniqueCarMakers$manufacturer,
                       "Nationality" = UniqueCarMakers$nationality, s
mpg_leftjoin <- left_join(mpg, NatOfCar, by="manufacturer") %>%
                select(manufacturer, Nationality)
```

결측치 확인 및 제거

결측치 확인 및 제거를 위해서는 is.na 함수를 사용하며, na.omit함수를 이용하여 결측치 제거를 할 수도 있다.

```
# Check the location of NAs
head(is.na(FineDust_1))
```

```
##       Type Location    T1    T2    T3    T4    T5    T6
## [1,] FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [2,] FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [3,] FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [4,] FALSE    FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE
## [5,] FALSE    FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [6,] FALSE    FALSE  TRUE FALSE FALSE  TRUE  TRUE  TRUE
##         T7    T8    T9   T10   T11   T12   T13   T14   T15
## [1,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [2,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [3,] FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [4,] FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE FALSE
```

결측치의 제거

1. filter와 is.na 조합
2. na.rm = T 옵션 활용
3. na.omit함수 활용

```
# filter + is.na
FineDust_noNA_T1 <- FineDust_1 %>% filter(!is.na(T1))
# na.rm = T option
mean(FineDust_1$T1, na.rm = T)
```

```
## [1] 31
```

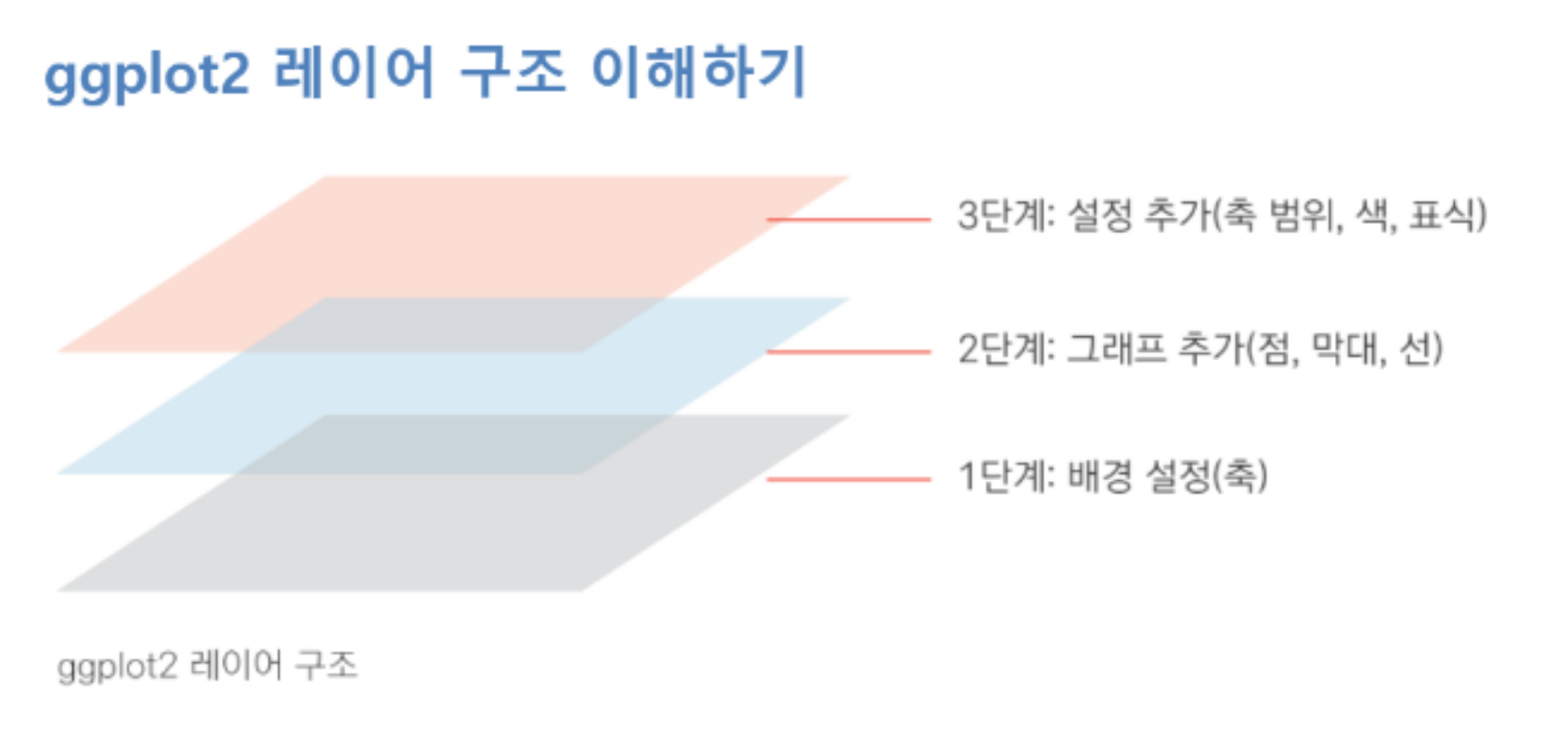
```
# na.omit function
FineDust_noNA_all <- na.omit(FineDust_1)
```

ggplot2를 이용한 그래프 작성

ggplot2

1. 그래프를 만들 때 가장 많이 사용하는 패키지.
2. 쉽고 짧은 문법으로 아름다운 그래프 작성 가능.
3. Layer 구조의 문법
    - 배경 제작
    - 그래프 형태 작성
    - 축 범위, 색, 표식 등 설정

ggplot2의 구조



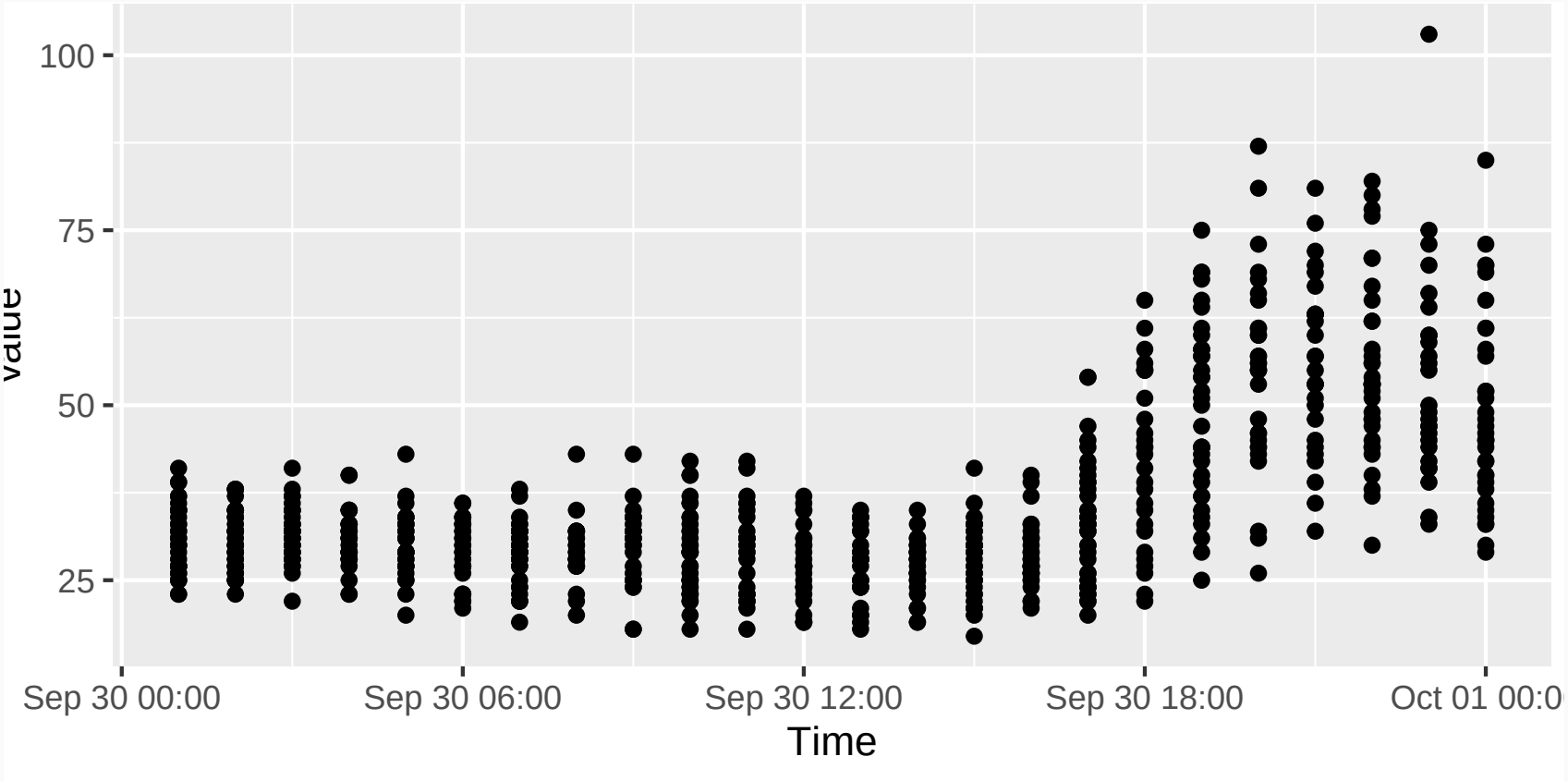
출처: 김영우 (2017) 쉽게 배우는 R 데이터 분석. 이지스퍼블리싱

ggplot2를 이용한 그래프 작성 - 시계열 그래프

```
# install.packages("ggplot2")
# install.packages("reshape2", repos="cran.seoul.go.kr")
# For better format for ggplot2
library(reshape2)
FineDust_melt <- melt(FineDust_1, id=c("Type", "Location"))
FineDust_plot <-
    FineDust_melt %>%
    mutate(Time = paste("2019-09-30 ", gsub("T", "", variable), ":00:
    mutate(Time = as.POSIXct(Time)) %>%
    filter(!is.na(value))
```

산포도 (Scatter plot)

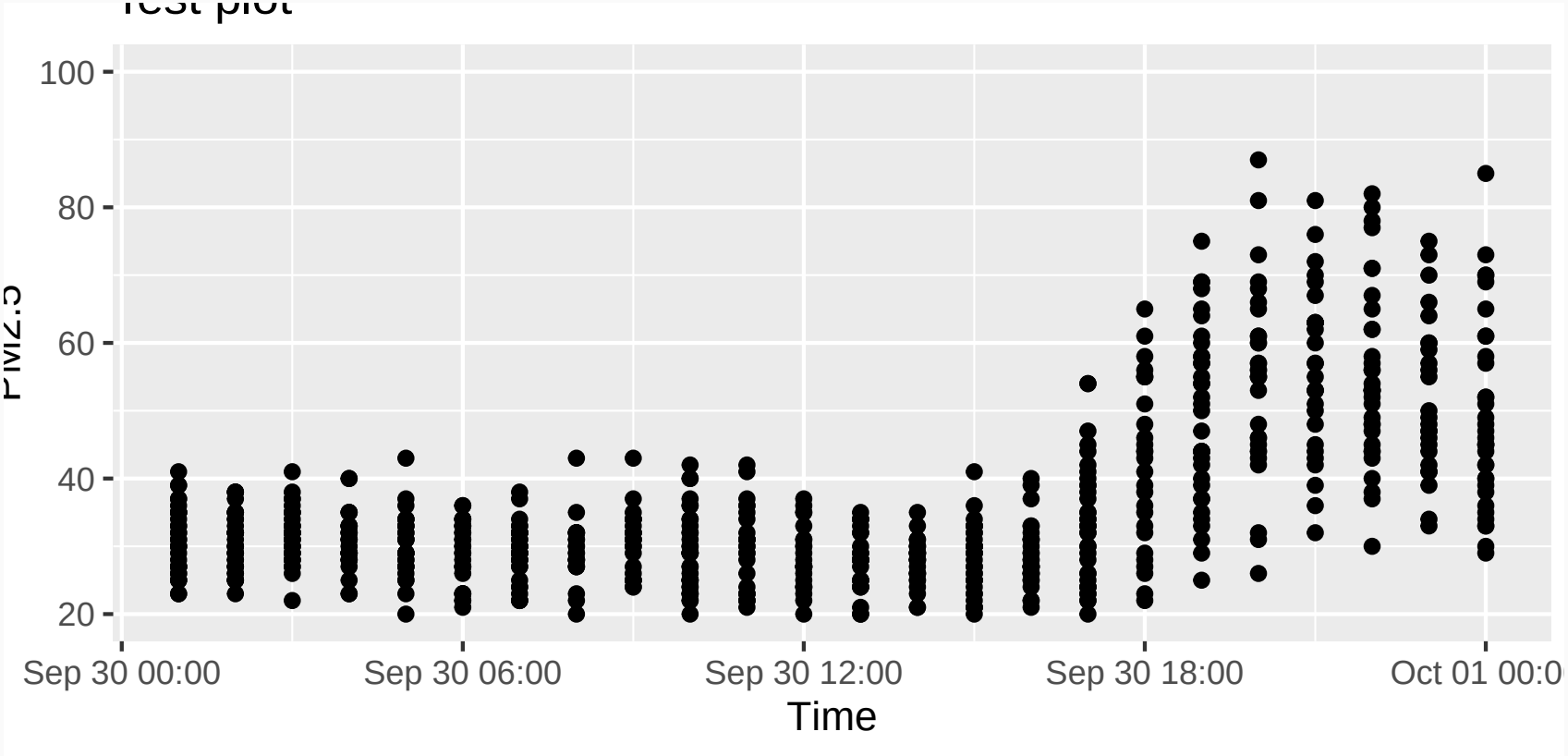
```r
TimeSeriesPlot <- ggplot(data=FineDust_plot,
                         aes(x = Time, y = value))  # Create backgrou
TimeSeriesPlot + geom_point() # Add plot to the background
```




```r
ggsave("./Data/ggplot2_test_r1.png") # Save the plot
```

산포도 (Scatter plot)

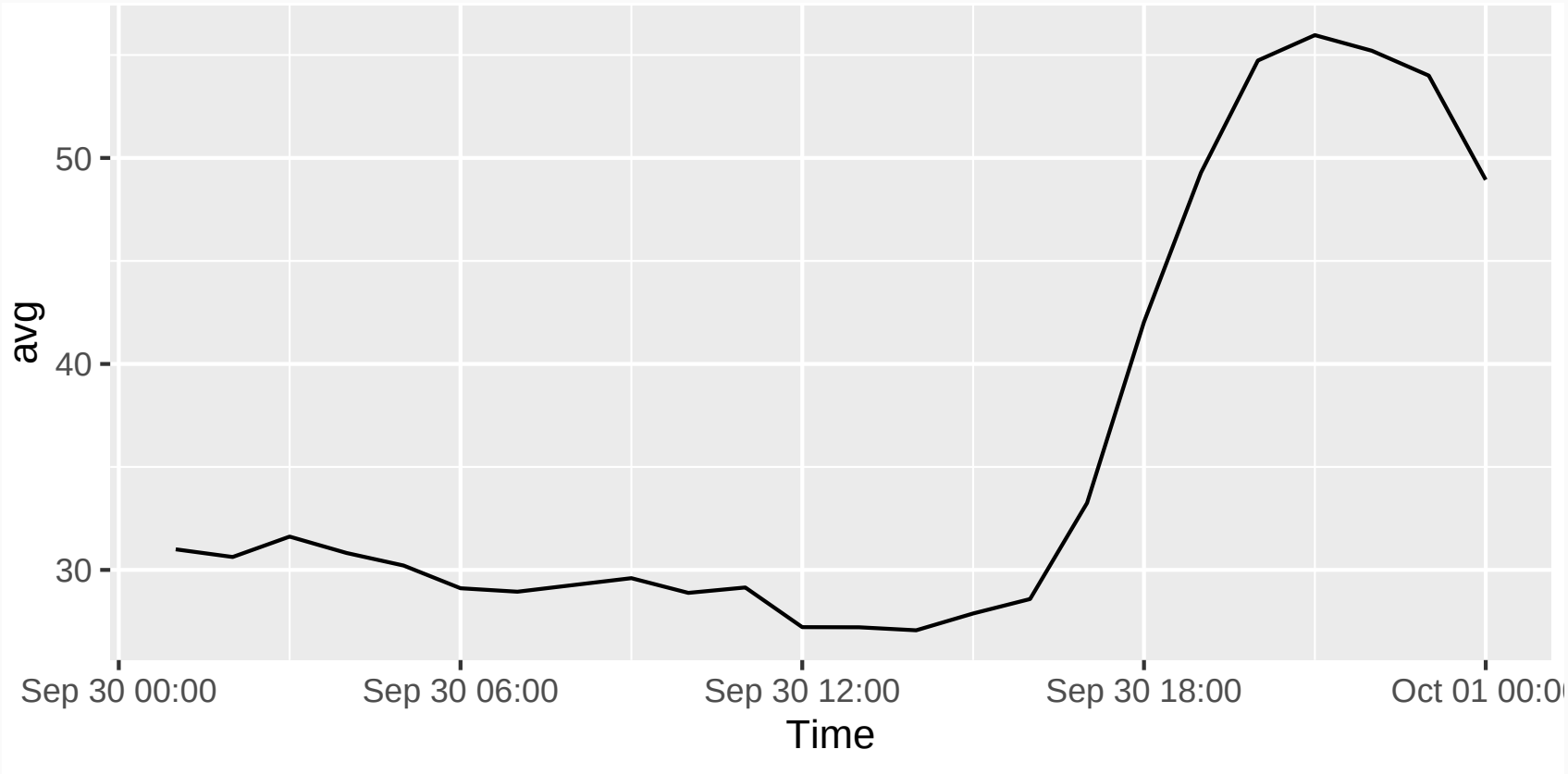
```
# Add additional options to the graph
TimeSeriesPlot + geom_point() + ylim(20, 100) +
    xlab("Time") + ylab("PM2.5") + ggtitle("Test plot")
```




```
# xlab, ylab, and ggtitle are equal to labs(title="", x="", y="")
```

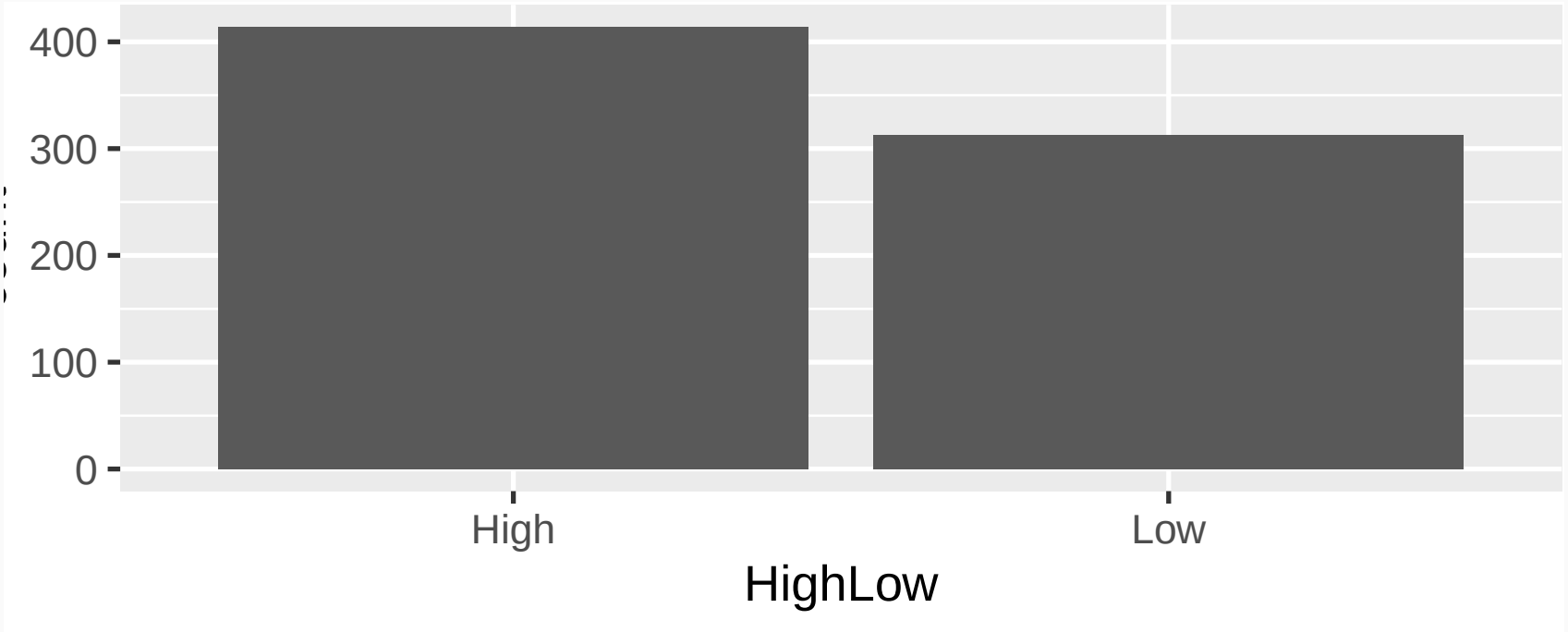
선 그래프 (line graph)

```
FineDust_plot_line <-   FineDust_plot %>% group_by(Time) %>%
                        summarize(avg = mean(value))
ggplot(data=FineDust_plot_line, aes(x=Time, y=avg)) + geom_line()
```



막대 그래프 (Bar chart)

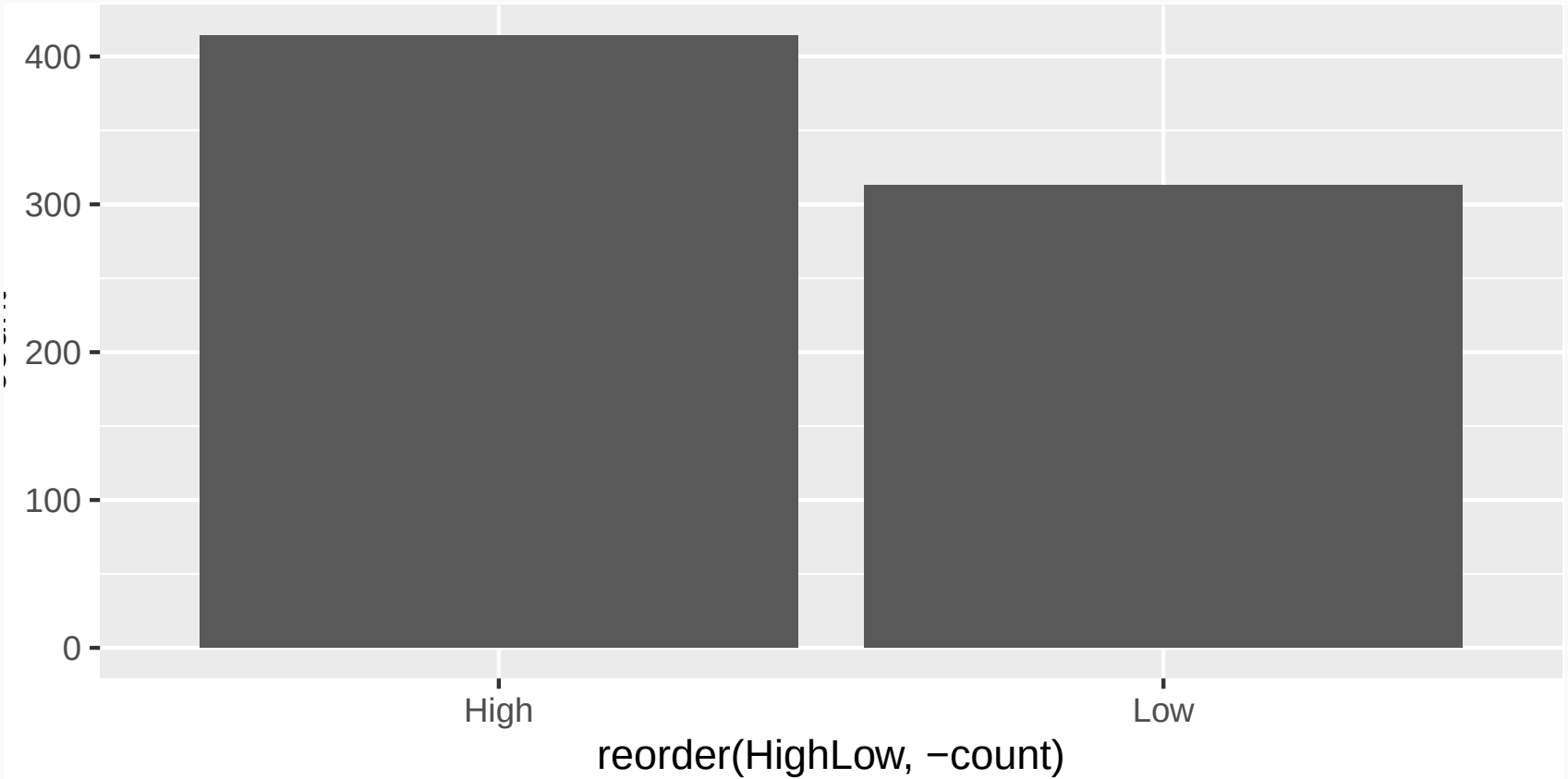
```
# Create datasets for bar chart
FineDust_Bar <- FineDust_melt %>% filter(!is.na(value)) %>%
                mutate(HighLow = ifelse(value > 30, "High", "Low")) %
                group_by(HighLow) %>% summarize(count= n())
# This is basic usage of bar chart
BarChart <- ggplot(data=FineDust_Bar, aes(x=HighLow, y=count))
BarChart + geom_col()
```



막대 그래프 결과

reorder 함수를 이용해서 막대 크기 순으로 정렬하여 표현 가능하다.

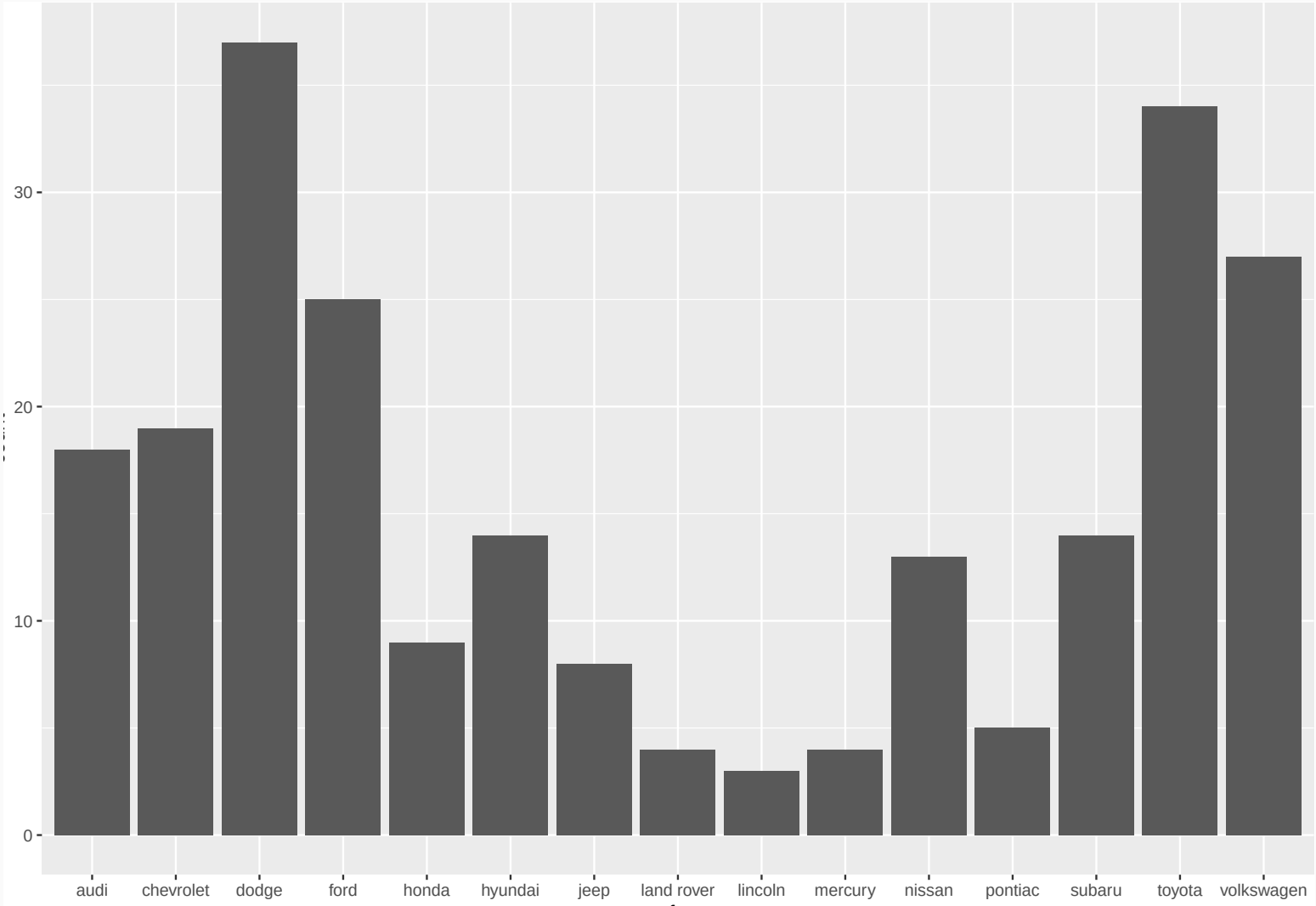
```
# This is basic usage of bar chart with decreasing
BarChart <- ggplot(data=FineDust_Bar,
                   aes(x=reorder(HighLow, -count), y=count))
BarChart + geom_col()
```



히스토그램 그리기

geom_bar() 함수 사용하면 히스토그램을 그릴 수 있다 (mpg data를 이용한 예)

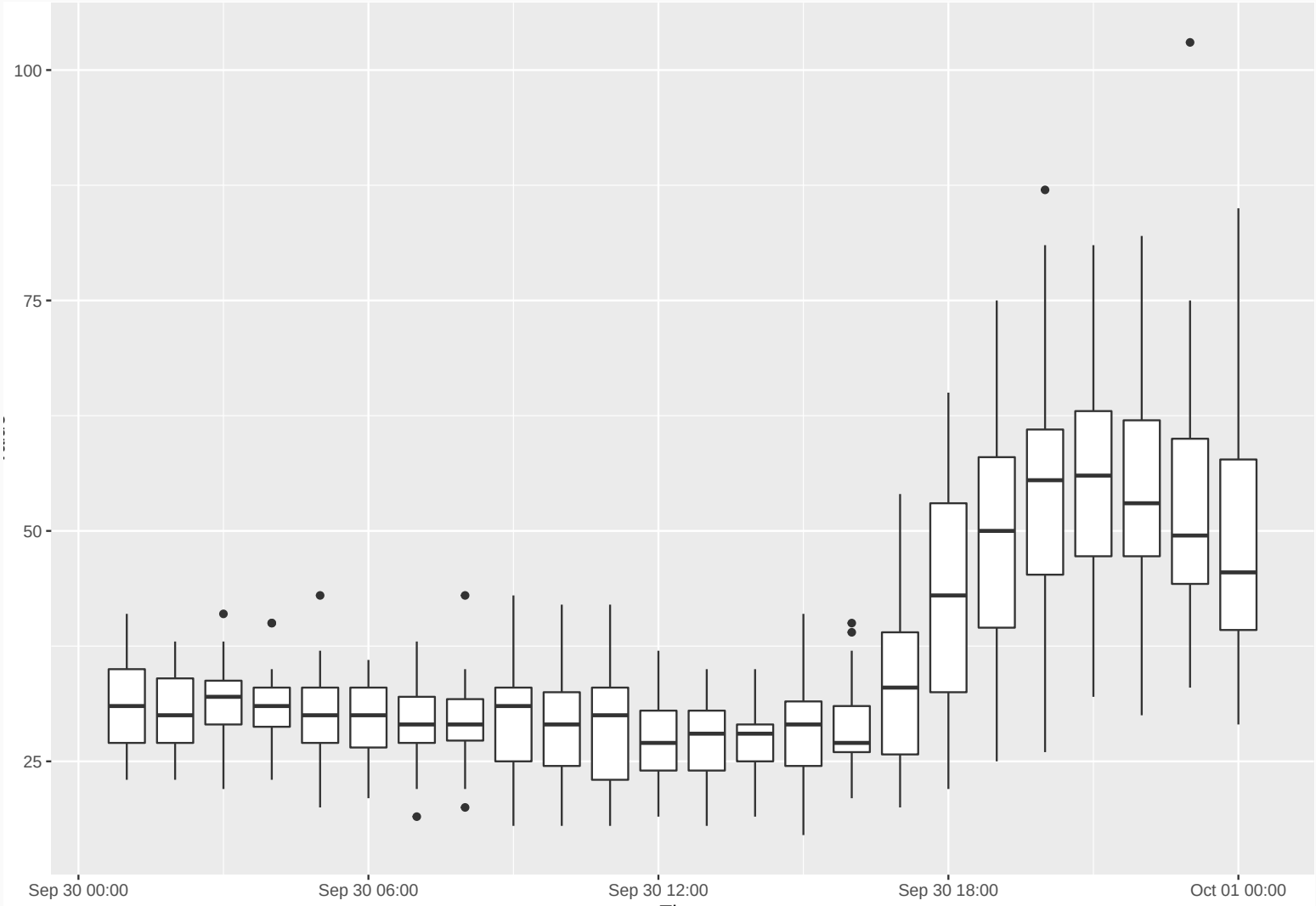
```
ggplot(data=mpg, aes(x = manufacturer)) + geom_bar()
```



상자 그림 (box plot)

집단 간 분포 차이를 표현하기 위해서 자주 쓰이는 그래프 (box_plot)

```
TimeSeriesPlot + geom_boxplot(aes(group=Time))
```



상자 그림 해석

상자 그림	값	설명
상자 아래 세로선	아래 수염	하위 0~25% 내에 해당하는 값
상자 밑면	1사분위수(Q1)	하위 25% 위치 값
상자 내 굵은 선	2사분위수(Q2)	하위 50% 위치 값(중앙값)
상자 윗면	3사분위수(Q3)	하위 75% 위치 값
상자 위 세로선	윗수염	하위 75~100% 내에 해당하는 값
상자 밖 점 표식	극단치	Q1, Q3 밖 1.5 IQR을 벗어난 값

참고 1.5 IQR: 사분위 범위(Q1~Q3간 거리)의 1.5배

출처: 김영우 (2017) 쉽게 배우는 R 데이터 분석. 이지스퍼블리싱

그래프의 중첩

ggplot의 특징

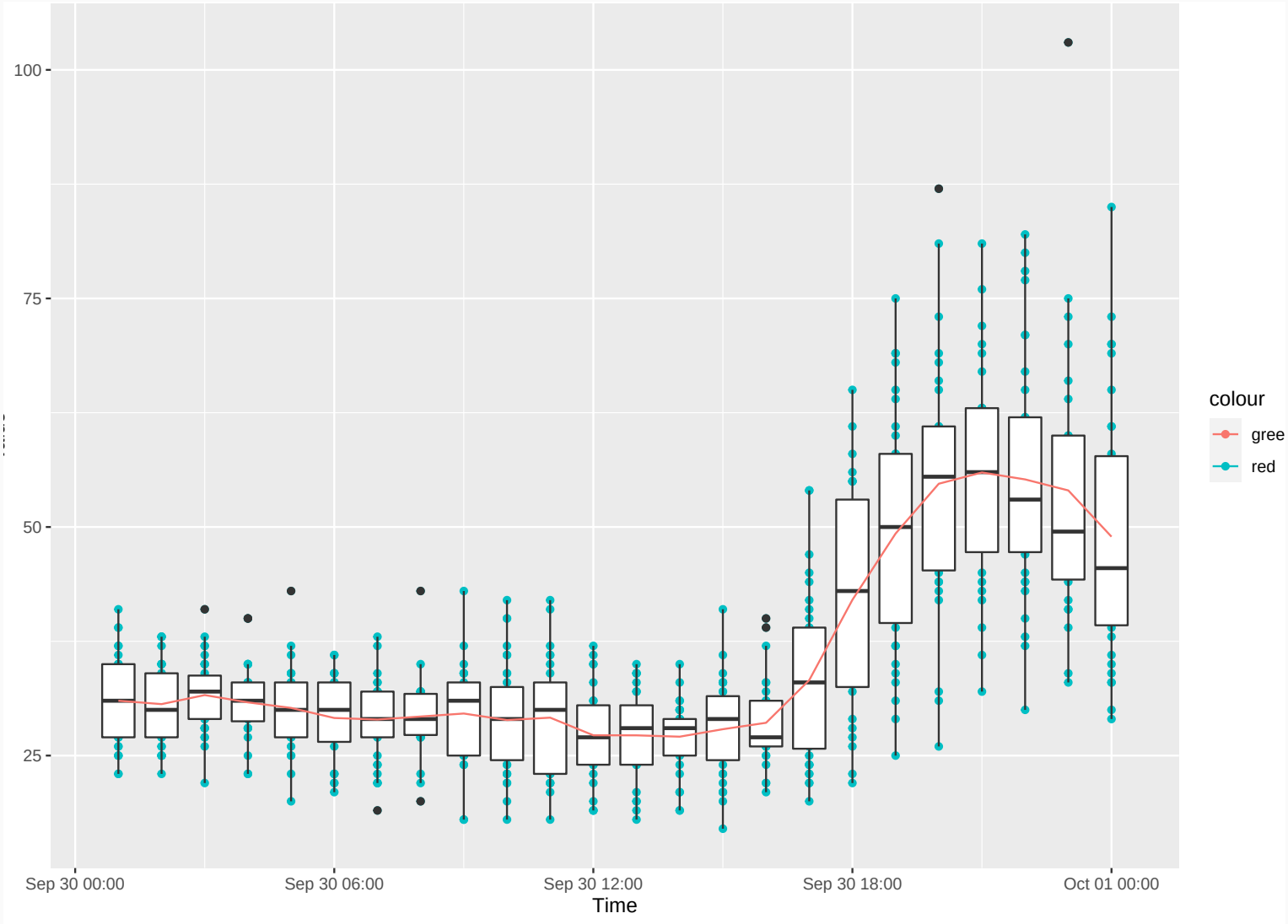
- layer 구조
  - 여러 종류의 그래프 중첩 가능
  - 그래프의 다양한 옵션을 중첩 가능
  - 예) 산포도 + 선 그래프 + 상자 그림

그래프의 중첩

```
OverlappedGraph <-  TimeSeriesPlot +
                    geom_point(aes(colour = "red")) +
                    geom_boxplot(aes(group=Time)) +
                    geom_line(data = FineDust_plot_line,
                              aes(x = Time, y=avg, col="green"))
```

그래프의 중첩

OverlappedGraph



과제

첨부된 자료들 (PM25_20190930.xls와 PM100_20190930.xls)의 산포도와 선그래프들을 중첩시켜 하나의 그래프로 표현하시오.