R을 이용한 통계 기초와 데이터 분석

Lecture 3

남현진

한성대학교

2020

Section 1

Ggplot2

왜 Ggplot2을 사용하는가

쉬운 코딩을 가지고 좋은 디자인의 시각화를 위해서는 ggplot2를 사용하는 것이 효율적이다. 우선, 예시를 통해 그래프를 기본 패키지로 그리는 코드와 ggplot을 이용한 코드로 비교해 보며 ggplot에 대해 알아보자.

```
head(data)
```

```
##   time variable     value
## 1    1        A 10.747427
## 2    2        A 11.750229
## 3    3        A 12.154331
## 4    4        A 11.152928
## 5    5        A 11.144169
## 6    6        A  9.523301
```

왜 Ggplot2을 사용하는가

Ggplot2을 사용했을 때

```
ggplot(data) +
  geom_line( aes(x = time, y = value, color = variable))
```

Ggplot2을 사용하지 않았을 때

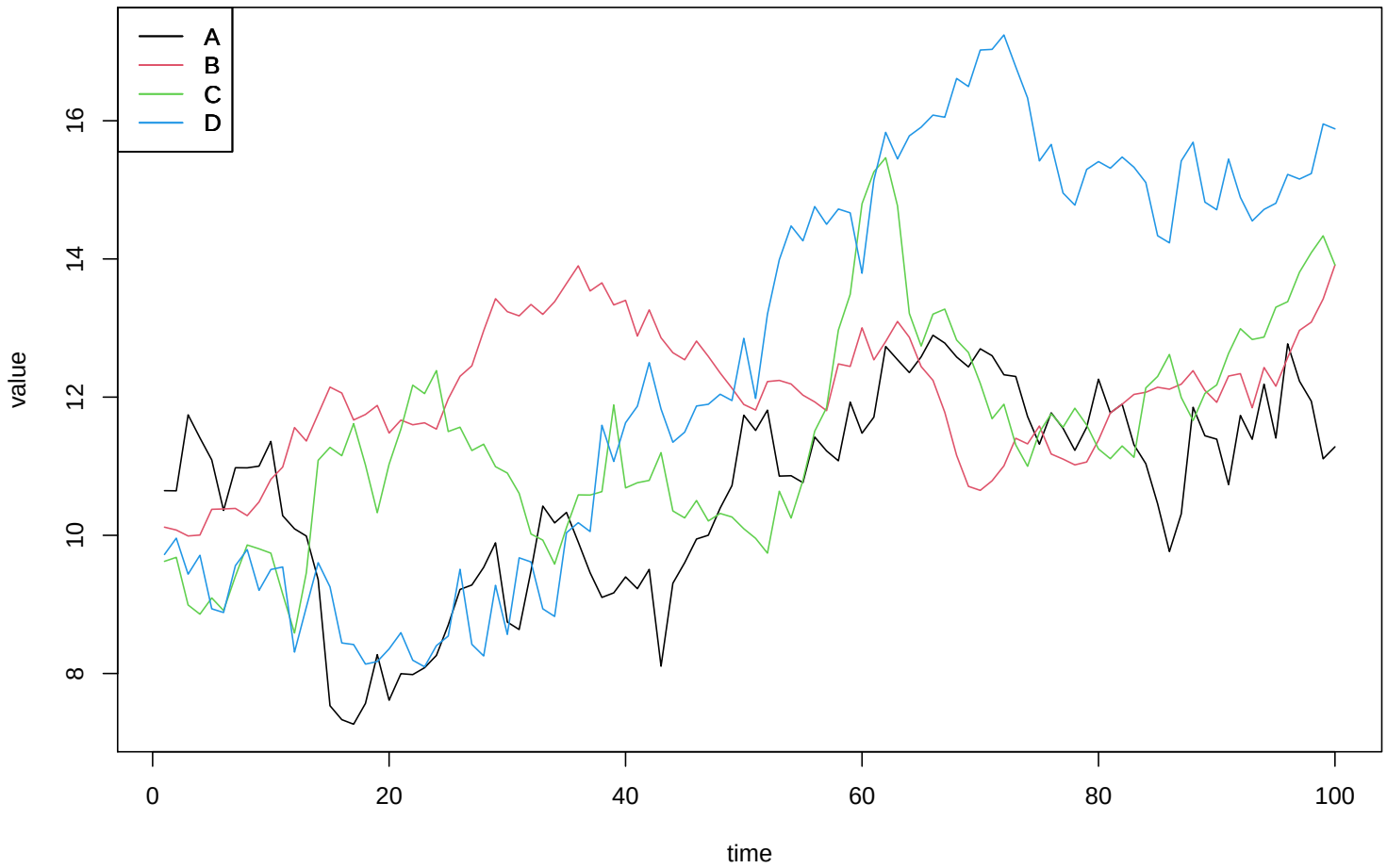
```
# Using base graphics
plot(data$time[data$variable == "A"], data$value[data$variable == "A"],
     type = "l", col = 1, ylim = c(min(data$value), max(data$value)),
     ylab = "value", xlab = "time")
for (i in 2:4) {
lines(data$time[data$variable == unique(data$variable)[i]],
        data$value[data$variable == unique(data$variable)[i]], col = i)
legend("topleft", legend = c("A", "B", "C", "D"), col = 1:4, lty = 1)}
```

왜 Ggplot2을 사용하는가

Ggplot2을 사용했을 때

왜 Ggplot2을 사용하는가

Ggplot2을 사용하지 않았을 때



Introduction

Ggplot2는 ”grammar of graphics”이다. Ggplot2는 빈 캔버스에 geoms을 추가하고, 다른 요소들을 레이어로 더해서 플롯을 완성하는 방식이다. 레이어링 할 수 있는 요소들은 다음과 같다(이외에도 더 많은 요소들이 있다. cheatsheet 참조).

- Data: 시각화하려는 데이터를 의미한다.
- Geometrics: 데이터를 표현하는 플랏의 종류를 의미한다. 산점도의 점, 그래프의 막대나 선 같이 데이터를 매핑하는 모양이라 할 수 있다.
- Aesthetics : 축의 스케일, 색상, 채우기 등 미학적 속성을 의미한다.
- Labs: x축의 이름, y축의 이름, 제목 등 플랏의 설명이 들어가는 요소이다.

Introduction

- Geometric objects
  - geom_point: 산점도
  - geom_bars: 막대 그래프
  - geom_histograms: 히스토그램
  - geom_density: 확률 분포도
  - geom_boxplots: 상자그림
  - geom_lines: 선 그래프
- Aesthetics
  - x: x축
  - y: y축
  - color: 선의 색상
  - fill: 채우기 색상
  - shape: 포인트의 모양
  - size: 크기
  - linetypes: 선 종류
  - alpha: 투명도

Introduction

```
ggplot(data = <DATA>) +
  <Geometric objects>(mapping = aes(<Aesthetics>)) +
  labs(y = <The text for y axis>,
       x= <The text for x axis>,
       title=<The text for the title.>,
       subtitle = < The text for the sub-title.>)
```

- 데이터 DATA 를 사용해서 그래프를 그린다.
- Geometric objects에 해당하는 종류의 그래프를 그린다.
- 원하는 표현 방식을 Aesthetics 적는다.

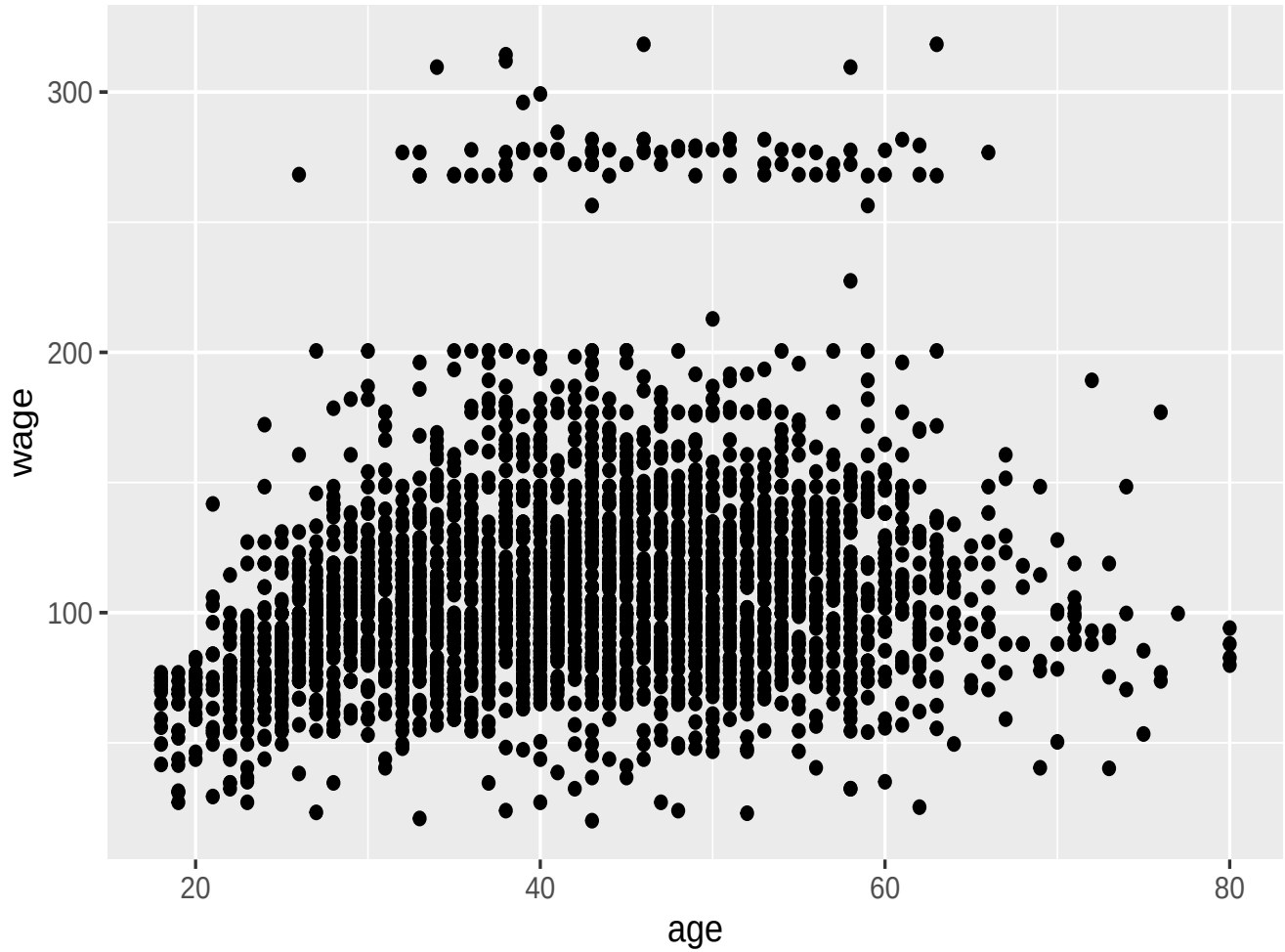
Introduction

Wage 데이터를 사용해서 나이와 임금의 상관관계를 알아보자.

```
Wage %>%
  select(age, wage) %>%
  head(5)
##        age      wage
## 231655  18  75.04315
## 86582   24  70.47602
## 161300  45 130.98218
## 155159  43 154.68529
## 11443   50  75.04315
```

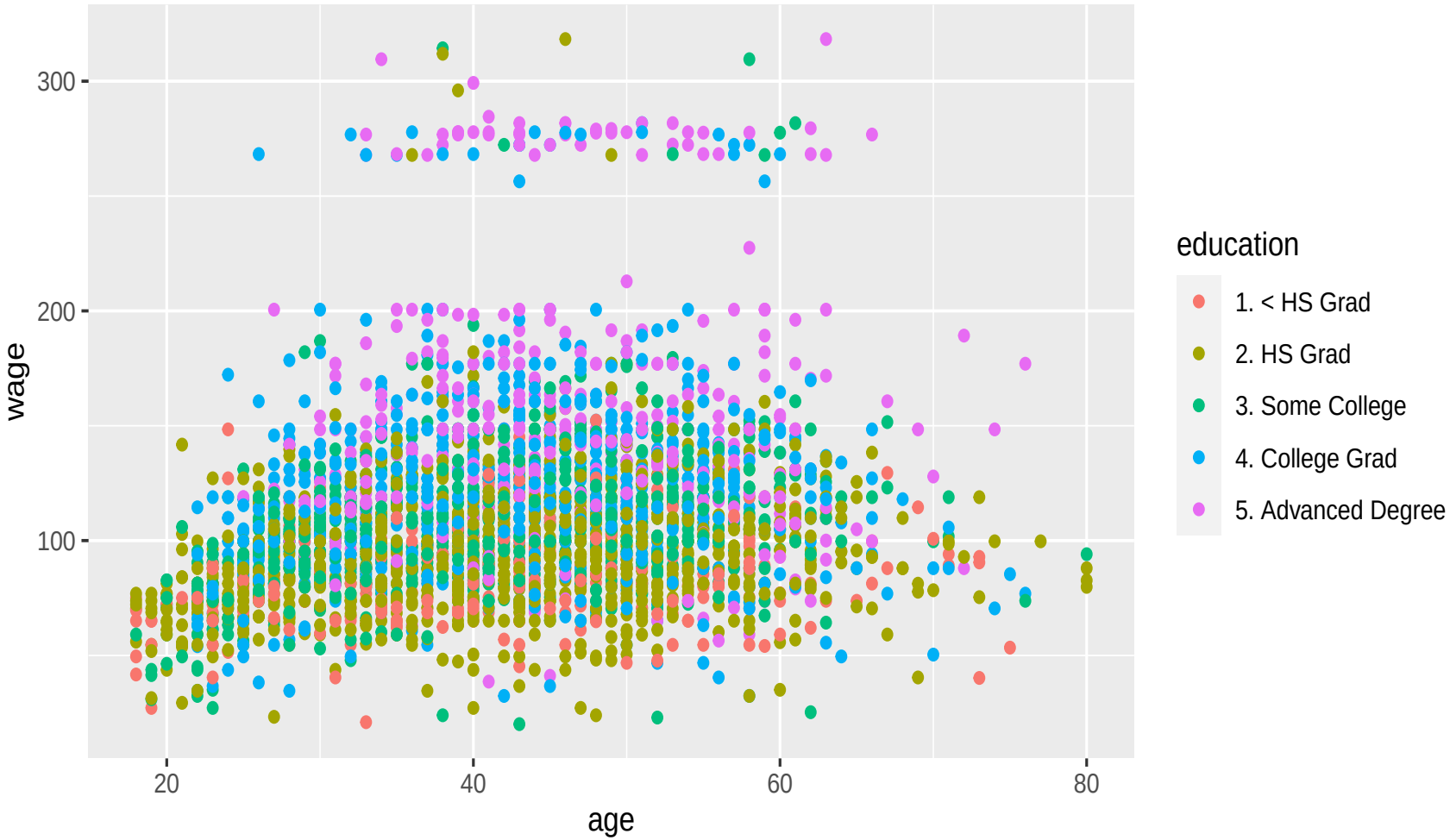
Scatterplots

```
ggplot(data = Wage) +
  geom_point(mapping = aes(x = age, y = wage))
```



Scatterplots

```
ggplot(data = Wage) +
  geom_point(mapping = aes(x = age, y = wage, color = education))
```

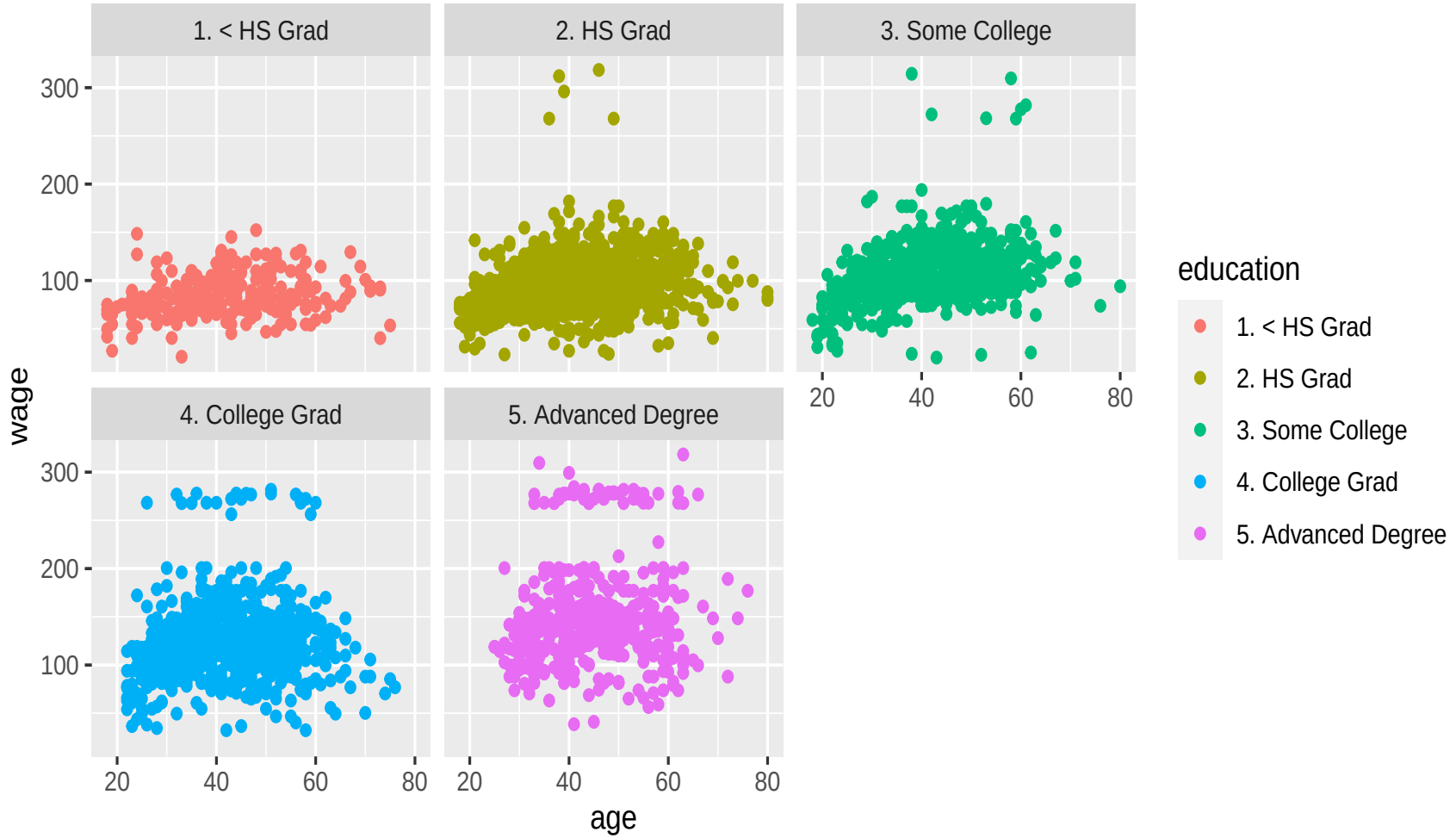


Scatterplots

```
ggplot(data = Wage) +
  geom_point(mapping = aes(x = age, y = wage, color = education, shape = education))
```

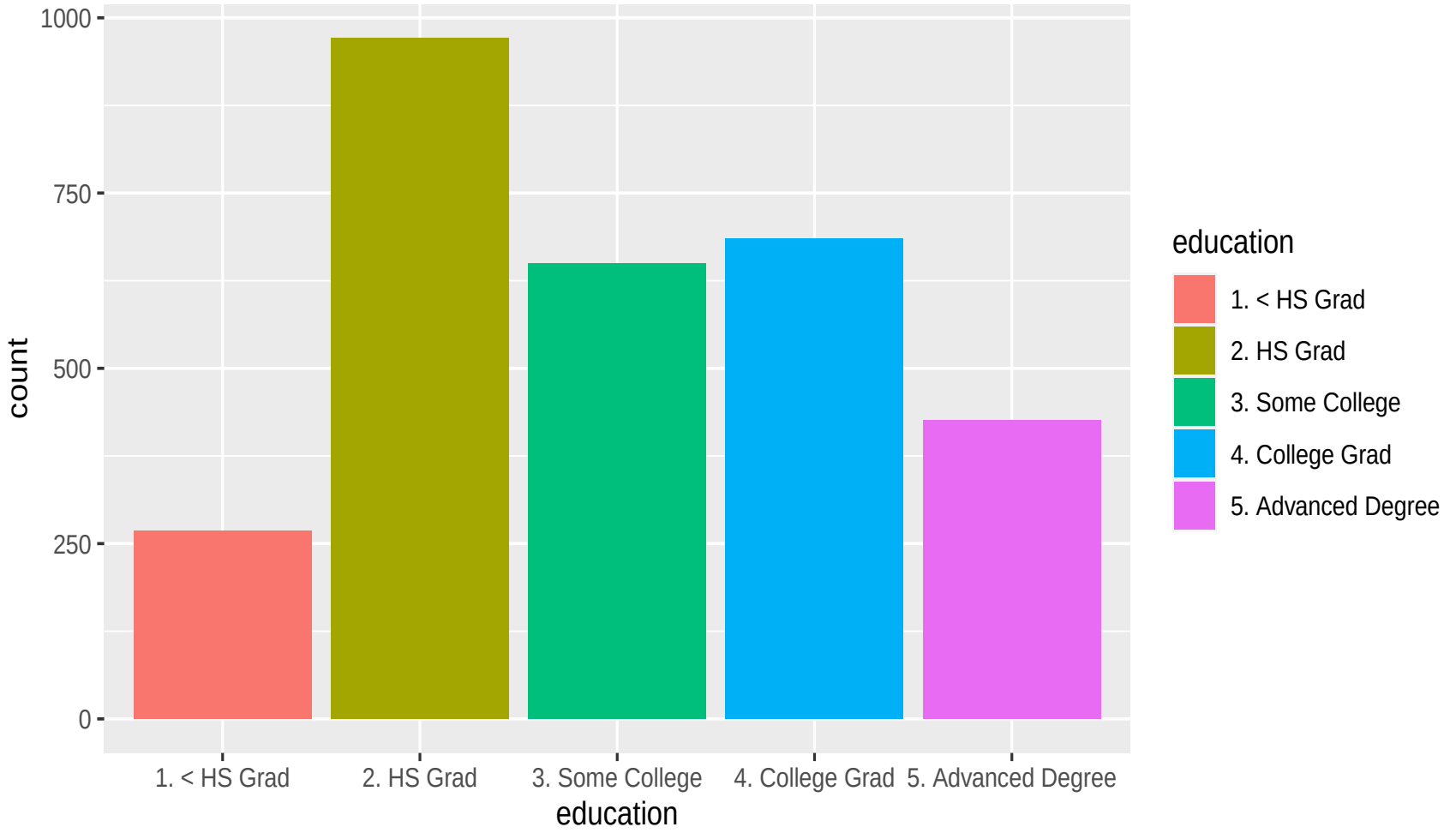
Multiple layer

```
ggplot(data = Wage) +
  geom_point(mapping = aes(x = age, y = wage, col = education)) +
  facet_wrap(~ education)
```



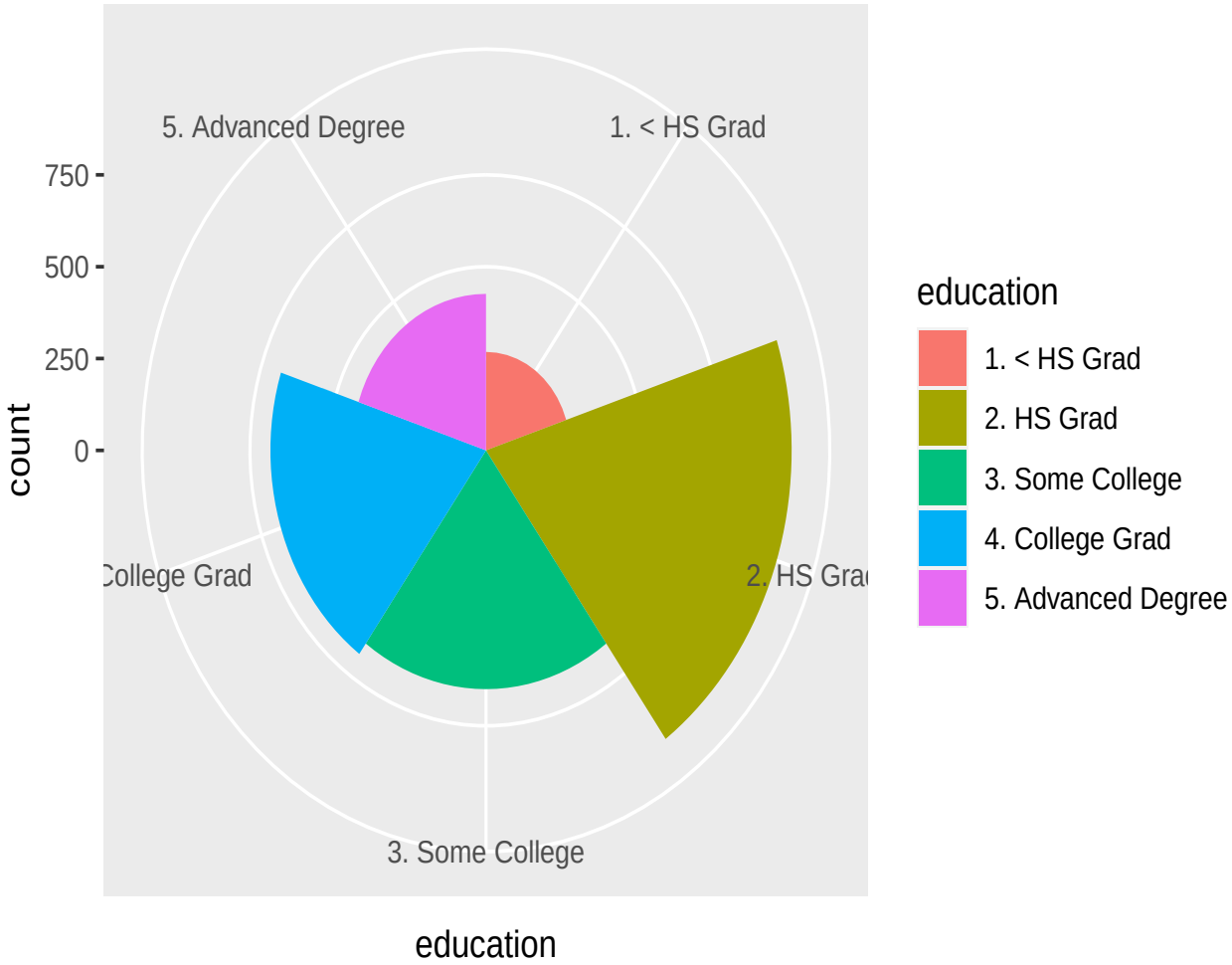
Bar chart

```
ggplot(data = Wage) +
  geom_bar(mapping = aes(education, fill=education))
```



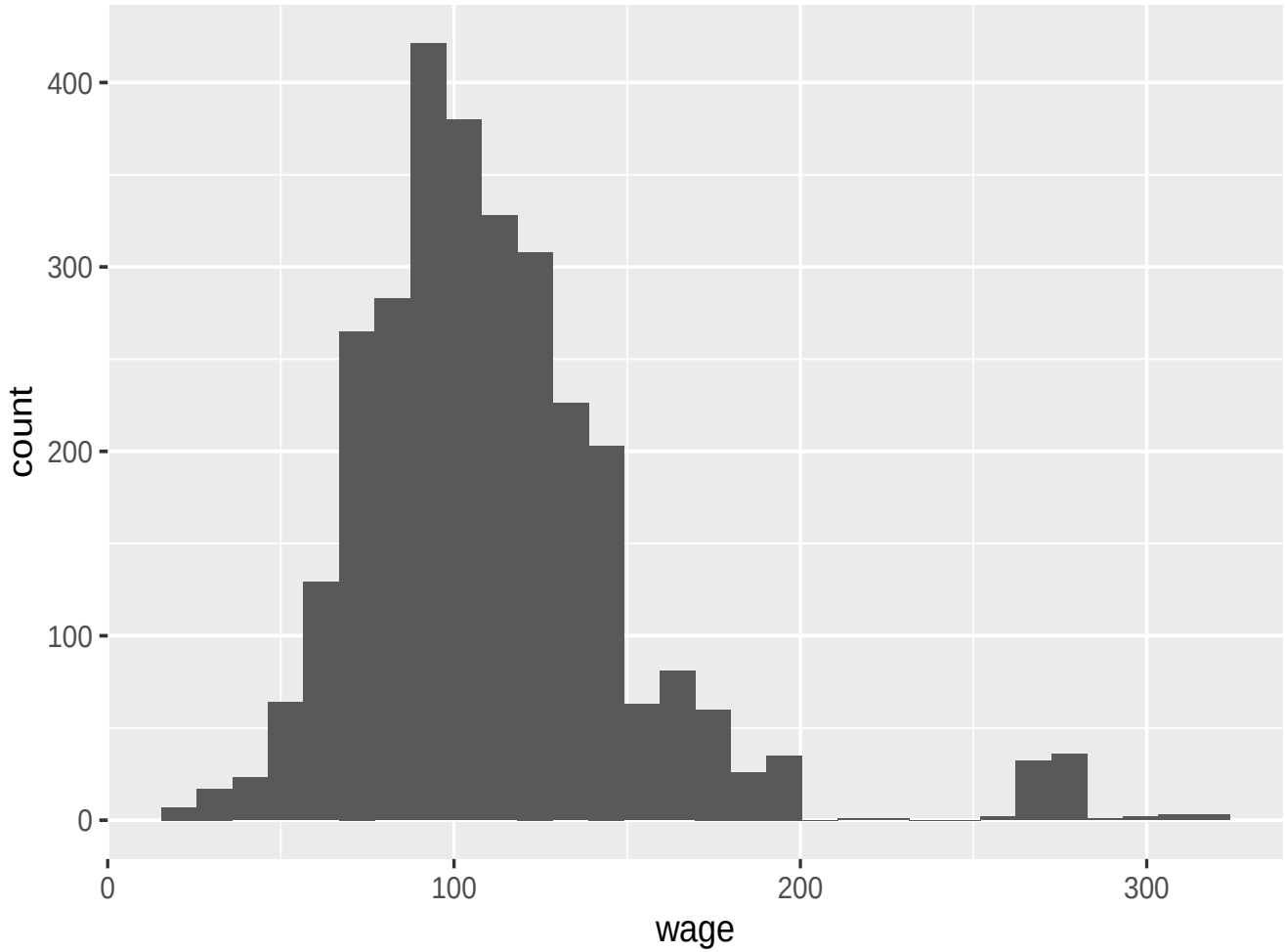
Bar chart

```
ggplot(data = Wage) +
  geom_bar(mapping = aes(education, fill=education), width = 1) +
  coord_polar()
```



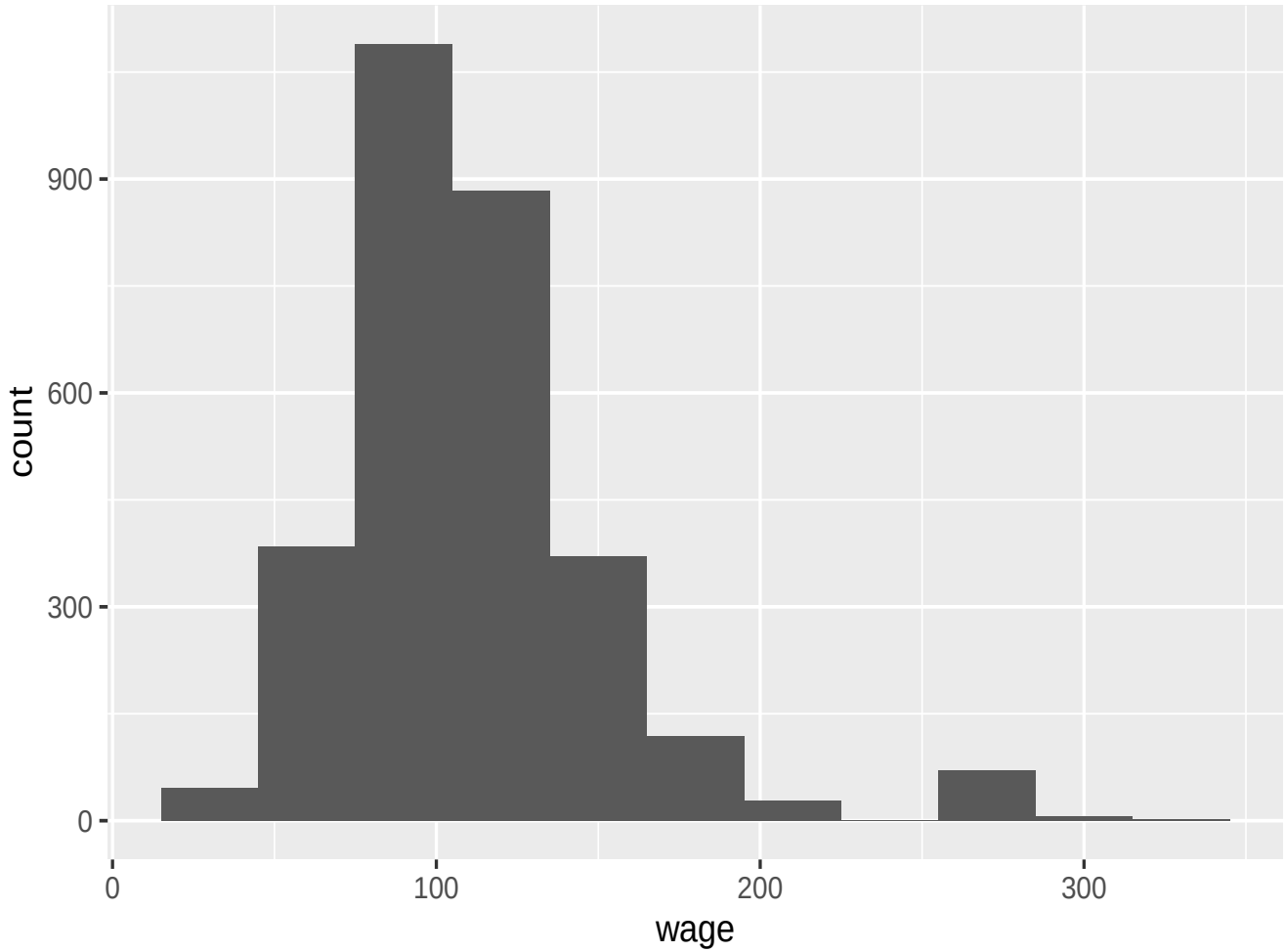
Histogram

```
ggplot(data = Wage) +
  geom_histogram(mapping = aes(wage))
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



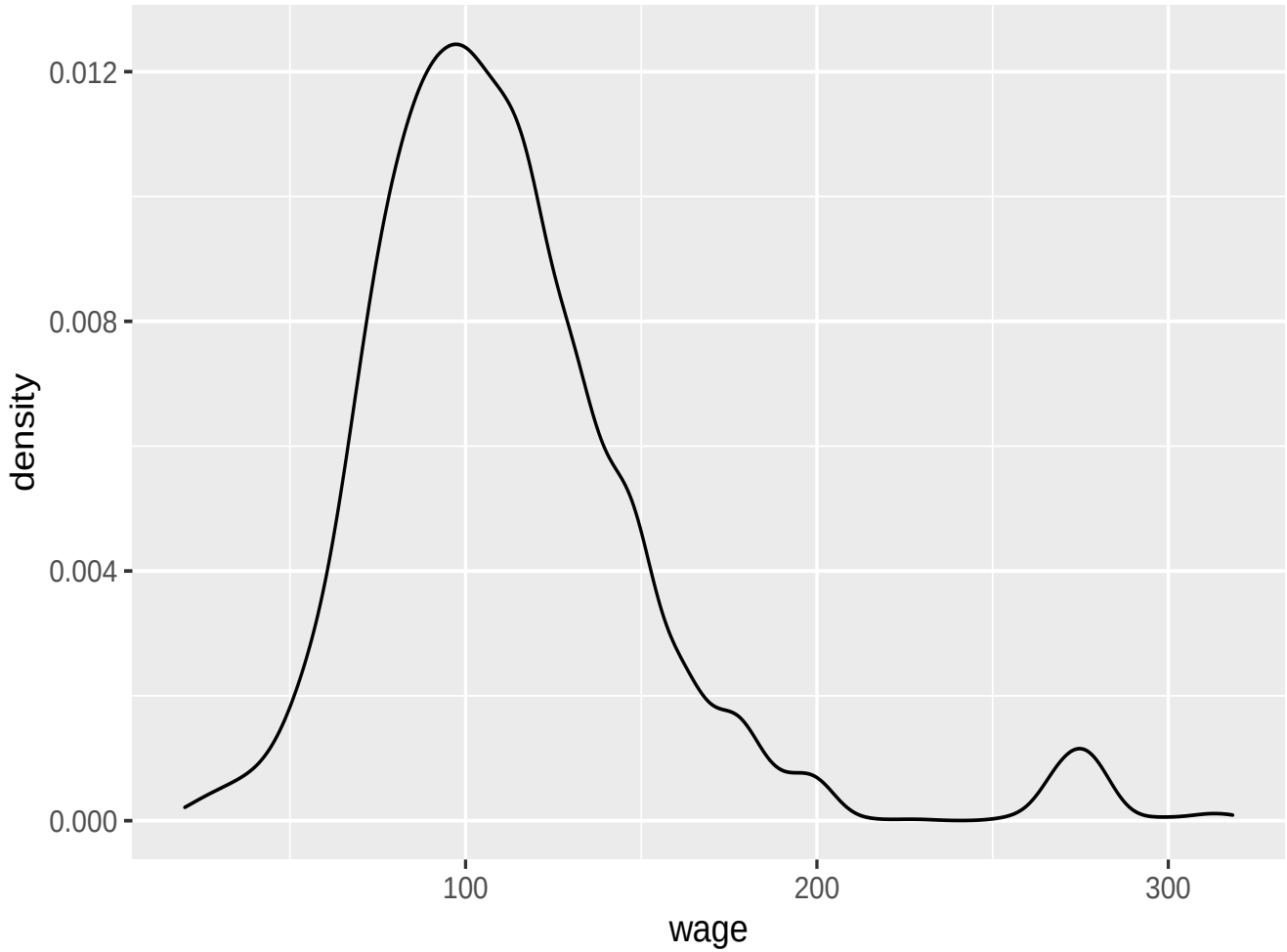
Histogram

```
ggplot(data = Wage) +
  geom_histogram(mapping = aes(wage), binwidth =  30)
```



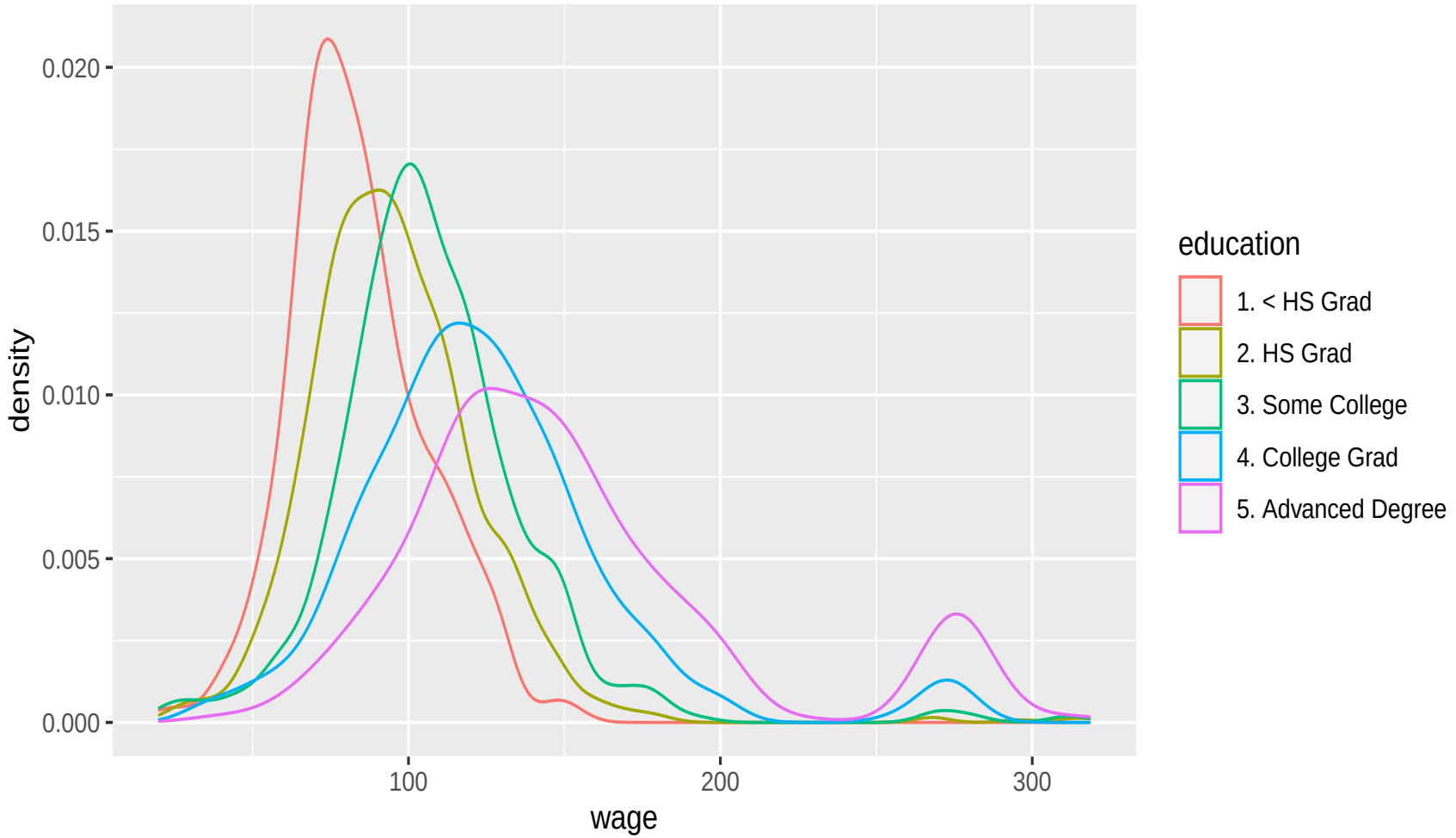
Density

```
ggplot(data = Wage) +
  geom_density(mapping = aes(x = wage, y = ..density..))
```



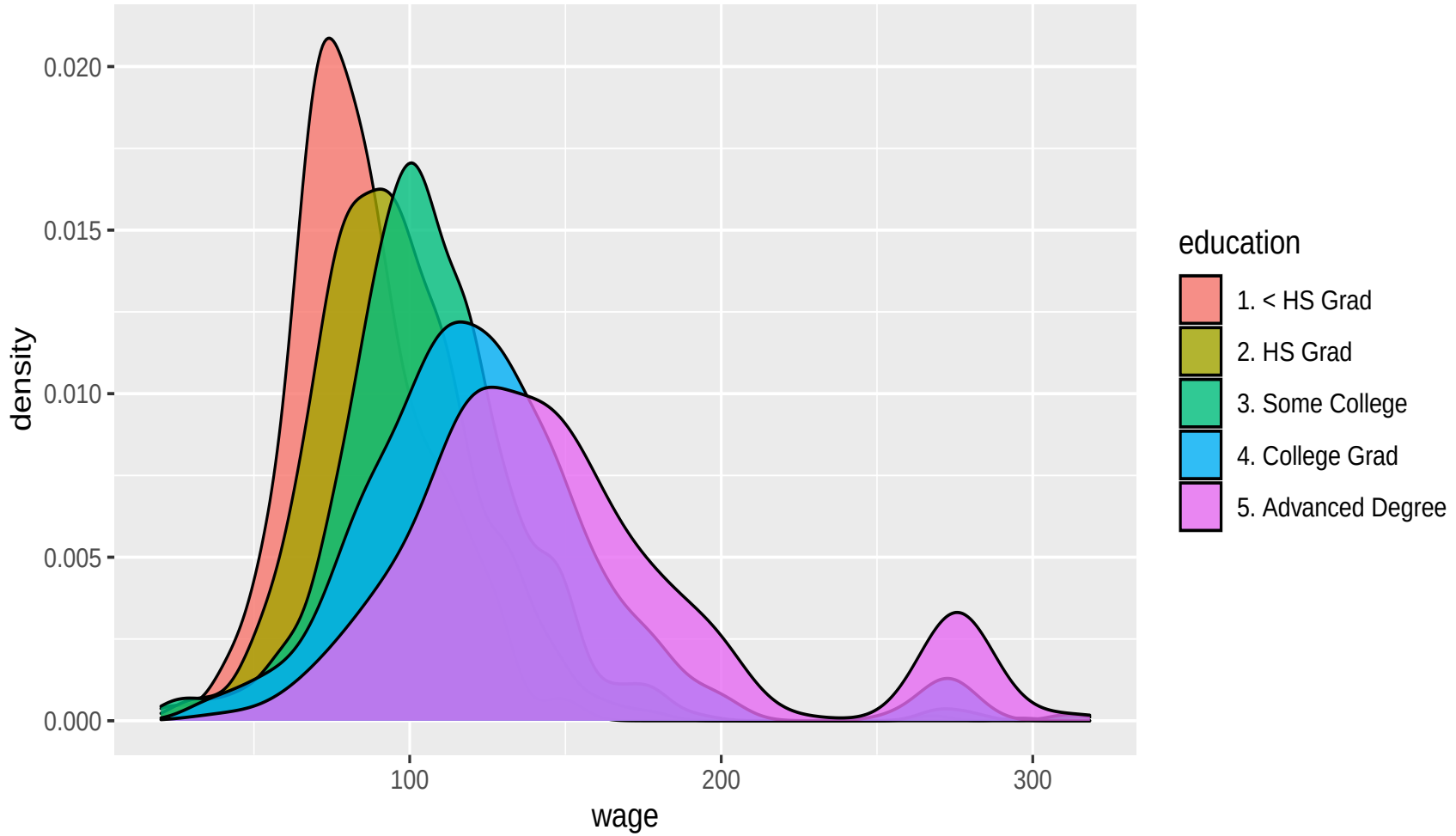
Density

```
ggplot(data = Wage) +
  geom_density(mapping = aes(x = wage, y = ..density.., col = education))
```



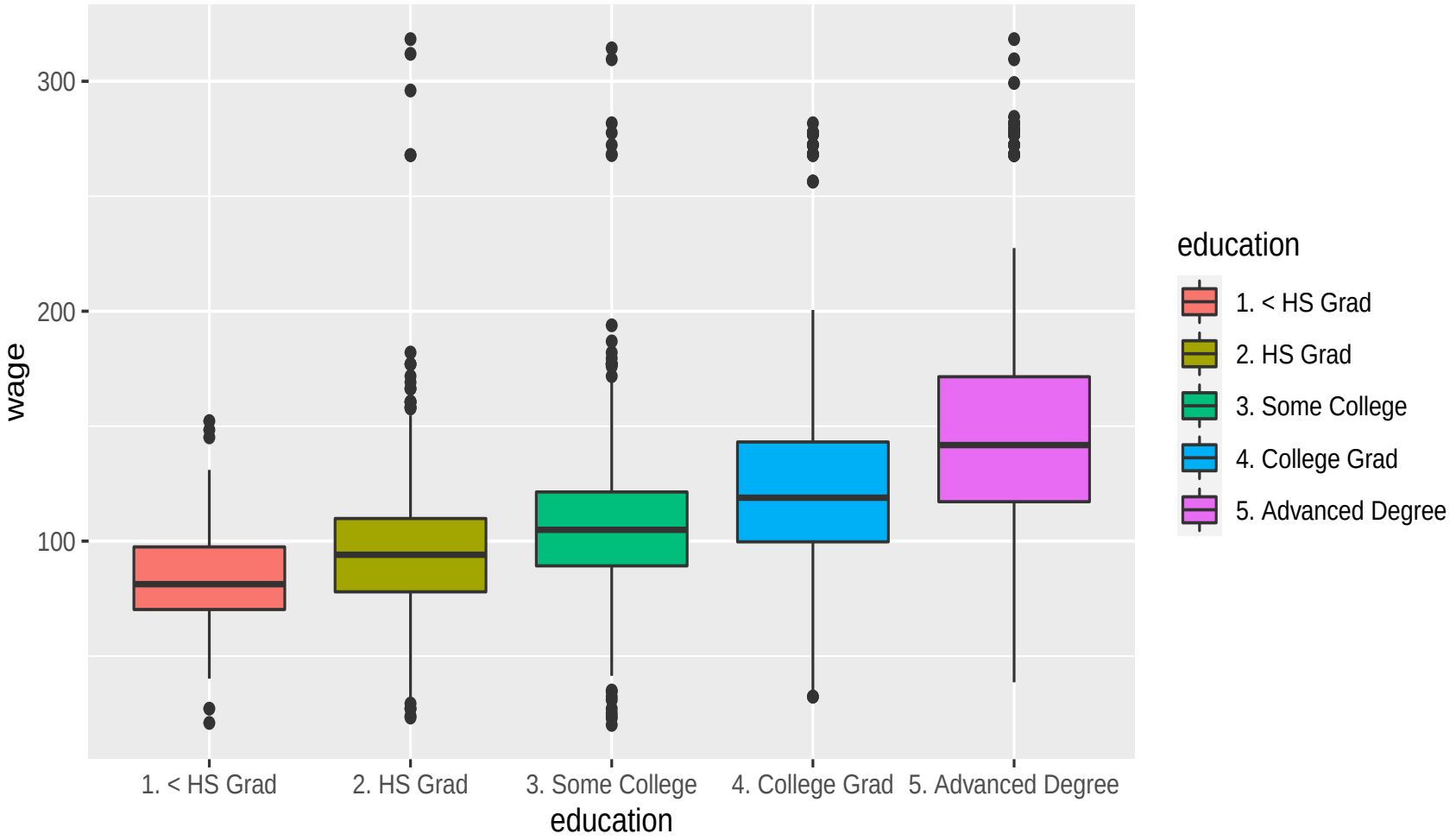
Density

```
ggplot(data = Wage) +
  geom_density(mapping = aes(x = wage, y = ..density.., fill = education),
               alpha = 0.8)
```



Box plot

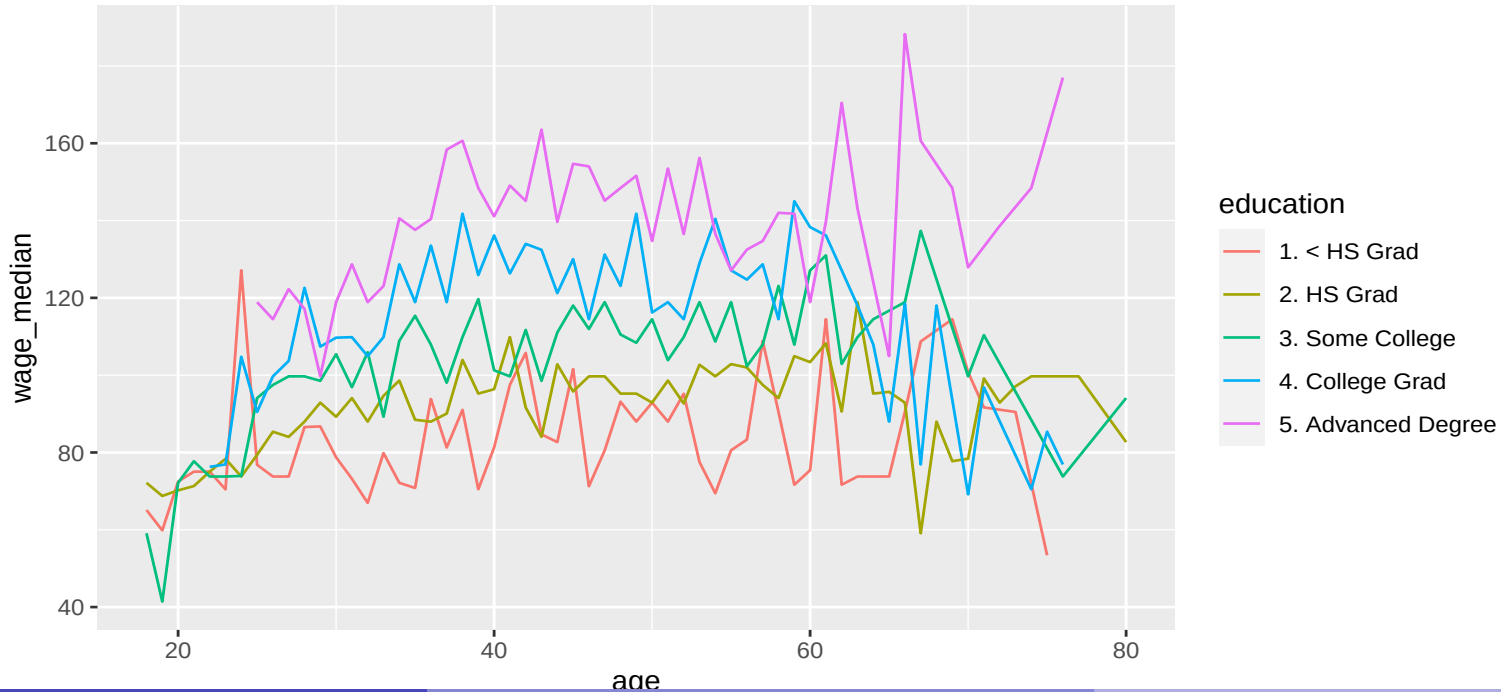
```
ggplot(data = Wage) +
  geom_boxplot(mapping = aes(x = education, y = wage, fill = education))
```



Line

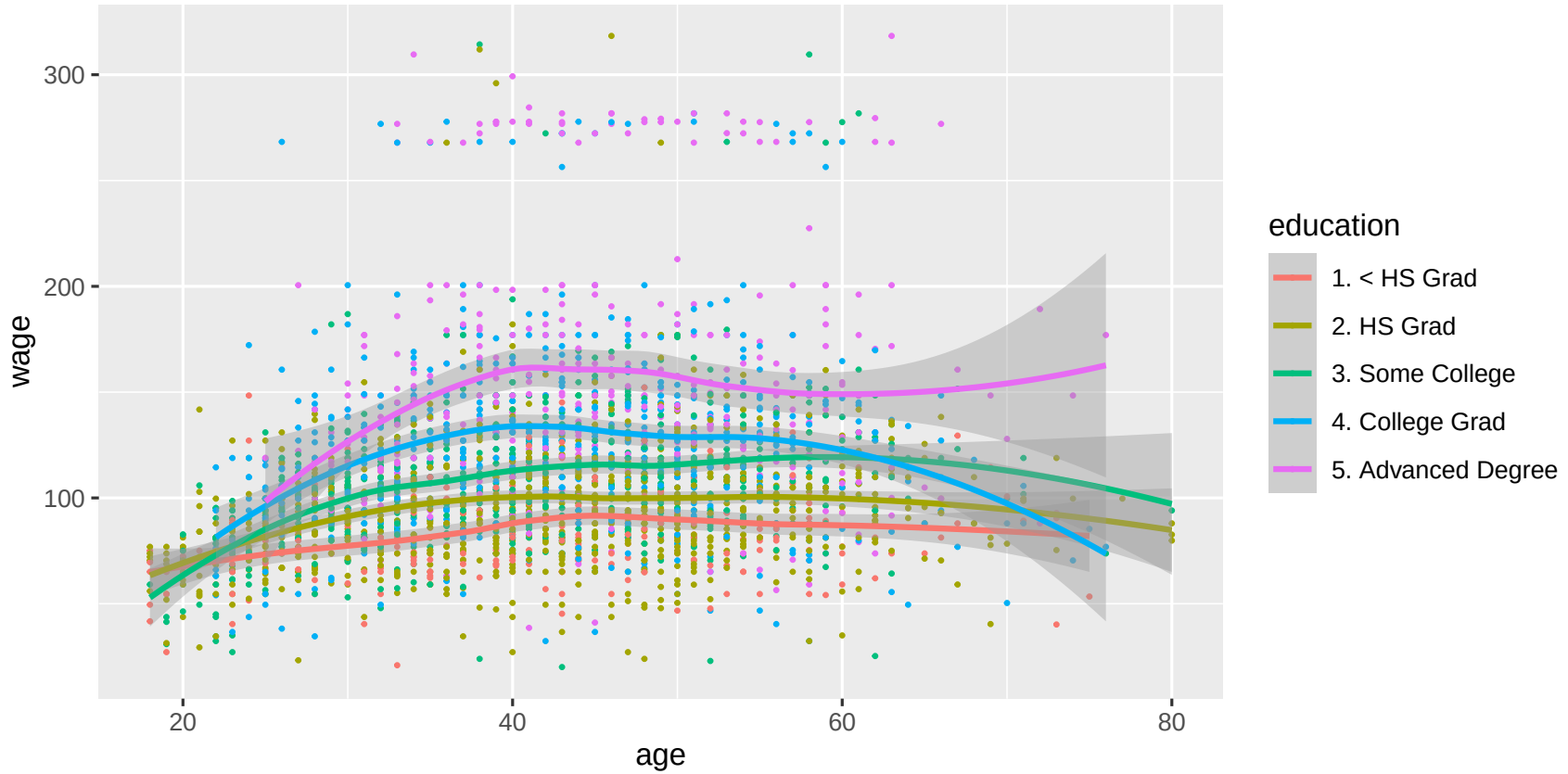
```
Wage1 <- Wage %>%
  group_by(education, age) %>%
  summarise(wage_median = median(wage))
## `summarise()` regrouping output by 'education' (override with `.groups` argument)


ggplot(data = Wage1) +
  geom_line(mapping = aes(x = age, y = wage_median, color = education))
```



scatter

```
ggplot(data = Wage) +
  geom_point(mapping = aes(x = age, y = wage, color = education),  size = 0.4) +
  stat_smooth(aes(x = age, y = wage, color = education), method='loess')
## `geom_smooth()` using formula 'y ~ x'
```



Section 2

실습

Section 3

Covid 데이터

데이터 불러오기

read.csv 함수를 이용해서 데이터를 불러온다. 현재 R이 사용하고 있는 디렉토리는 getwd()로 확인할 수 있으며 디렉토리 변경은 setwd()를 이용하면 된다.

```
getwd()
## [1] "/Users/hyunjinnam/course-statistics/Lecture 3"
#setwd()
covid.data <- read.csv('covid_data.csv')
```

데이터 탐색

오늘 사용하게 될 데이터는 전세계의 나라 별 코로나 확진자 수이다. 변수는 총 3가지로 나라, 첫 확진자 발생일을 기준으로 경과 일, 그리고 확진자 수이다.

```
str(covid.data)
```

```
## 'data.frame':    31258 obs. of  3 variables:
##  $ Country: chr  "Afghanistan" "Afghanistan" "Afghanistan" "Afghanistan" ...
##  $ Day    : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ N      : int  1 1 1 1 1 1 1 1 1 1 ...
```

나라들의 이름의 상위 15개 값을 살펴보면 다음과 같다.

```
unique(covid.data$Country) %>% head(15)
```

```
##  [1] "Afghanistan"          "Albania"              "Algeria"
##  [4] "Andorra"              "Angola"               "Antigua and Barbuda"
##  [7] "Argentina"            "Armenia"              "Australia"
## [10] "Austria"              "Azerbaijan"           "Bahamas"
## [13] "Bahrain"              "Bangladesh"           "Barbados"
```

데이터 탐색

데이터의 상위 7개의 값은 다음과 같다.

```
head(covid.data, 7)
##       Country Day N
## 1 Afghanistan   1 1
## 2 Afghanistan   2 1
## 3 Afghanistan   3 1
## 4 Afghanistan   4 1
## 5 Afghanistan   5 1
## 6 Afghanistan   6 1
## 7 Afghanistan   7 1
```

데이터 탐색

우리나라의 데이터를 뽑아서 살펴보자.

```
country <- 'Korea, South'
country.data <- covid.data %>%
  filter(Country == country)

head(country.data)
##         Country Day N
## 1 Korea, South   1 1
## 2 Korea, South   2 1
## 3 Korea, South   3 2
## 4 Korea, South   4 2
## 5 Korea, South   5 3
## 6 Korea, South   6 4
```

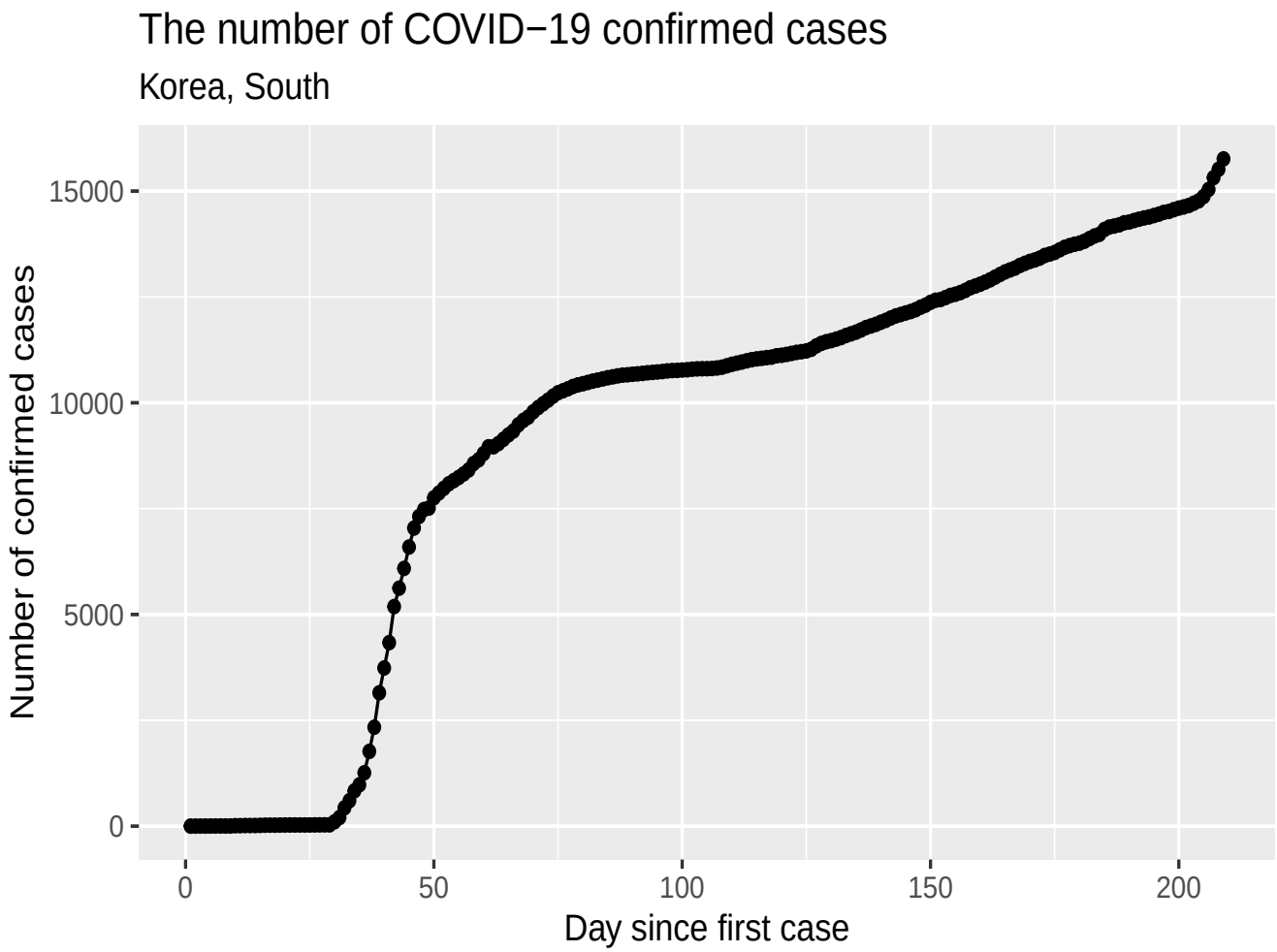
데이터 그리기

우리나라의 코로나 확진자 수를 그래프로 그려보자.

```
covid_korea <- ggplot(country.data) +
  geom_line(aes(x = Day, y = N)) +
  geom_point(aes(x = Day, y = N)) +
   labs(y = 'Number of confirmed cases',x="Day since first case",
        title="The number of COVID-19 confirmed cases",
        subtitle = country)
```

데이터 그리기

```
covid_korea
```



데이터 그리기

png 이용하면 ggplot을 로컬 파일로 저장할 수 있다. 원하는 플랏들을 프린트한 후 dev.off()를 사용하면 저장이 멈춘다.

```
png("covid_korea.png")
print(covid_korea)
dev.off()
```

데이터 그리기

반복문을 사용해서 나라별 코로나 확진자 수를 그래프로 그려보자.

```
country.list <- unique(covid.data$Country)
pdf('covid.pdf')
for(i in 1:length(country.list)){
  country <- country.list[i]
  country.data <- covid.data %>%
    filter(Country == country)
  forecast.plot <- ggplot(country.data) +
    geom_line(aes(x = Day, y = N)) +
    geom_point(aes(x = Day, y = N)) +
    labs(y = 'Number of confirmed cases',x="Day since first case",
         title="The number of COVID-19 confirmed cases",
         subtitle = country)
  print(forecast.plot)
}
dev.off()
```

데이터 그리기

링크 'https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_confirmed_global.csv' 를 이용하면 매일 업데이트되는 코로나 데이터를 가져올 수 있다.

```
url <- 'https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_
link <- getURL(url)
data <- read.csv(text = link)

data <- data %>%
  select(-Province.State,-Lat, -Long) %>%
  group_by(Country.Region) %>%
  summarise_all(sum, na.rm = TRUE)

country.list <- unique(data$Country.Region)
data.final <- data.frame()
forecast.final <- data.frame()
```

데이터 그리기

```
for(i in 1:length(country.list)){
  country <- country.list[i]
  country.data <- data %>%
    filter( Country.Region == country) %>%
    select(-Country.Region)

  first_day <- min(which(country.data > 0))
  last_day <-  ncol(country.data)
  N <- country.data[,first_day:last_day] %>% t() %>% as.vector()
  day <- seq(from = 1, to = last_day - first_day + 1, by = 1)
  country.data <- data.frame( Country = country,Day = day, N = N)
  data.final <- rbind(data.final, country.data)
}

data.final <- as.data.frame(data.final)
write.csv(data.final, 'covid_data.csv', row.names =  F)
```