Carseats - report (Advertising vs Sales)

learningSpoonsDS

2018-10-21

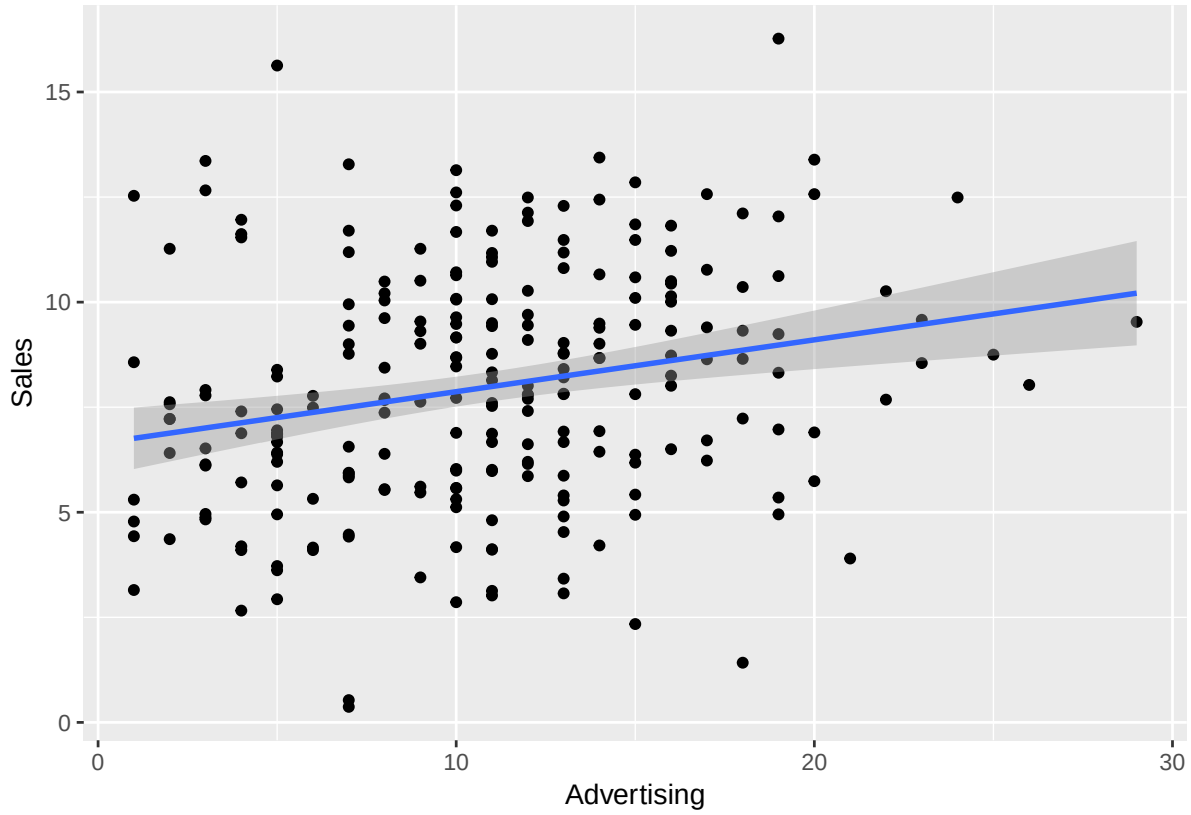
0. Data Import

```
library(dplyr)
library(ggplot2)
library(ISLR)
```

```
# str(Carseats)
Carseats <- Carseats %>%
  filter(US == "Yes") %>%
  filter(Advertising != 0)
```

1. Basic Plot + Purpose

```
xyPlot <- ggplot(data = Carseats, aes(x = Advertising, y = Sales)) +
  geom_point() + stat_smooth(method="lm")
print(xyPlot)
```




```
# colnames(Carseats)
setdiff(colnames(Carseats), c("Sales", "Advertising"))
```

```
## [1] "CompPrice"  "Income"     "Population" "Price"      "ShelveLoc"
## [6] "Age"        "Education"  "Urban"      "US"
```

- 도시 변수에 대해서 광고효과를 알아보는 리포트 입니다.

- 이 보고서는 `Advertising`이 `Sales`에 미치는 영향을 분석합니다.
- Income, Population, Age, Education, Urban, US

1. Factor 변수: Urban, US
2. non-Factor 변수: Income, Population, Age, Education

- 후보 추가 변수들은 다음과 같습니다. `setdiff(colnames(Carseats), c("Sales", "Adverting"))`: CompPrice, Income, Advertising, Population, Price, ShelveLoc, Age, Education, Urban, US.
- 각각의 데이터는 각각의 지역 (Spatial Data)를 의미합니다. 그리고 `Advertising`와 `Sales`를 제외한 다른 변수들은 1) 해당 지역의 특성 혹은 2) 해당 지역에서의 business practice의 특성을 반영합니다. (factor 변수는 bold로 표기하였습니다.)

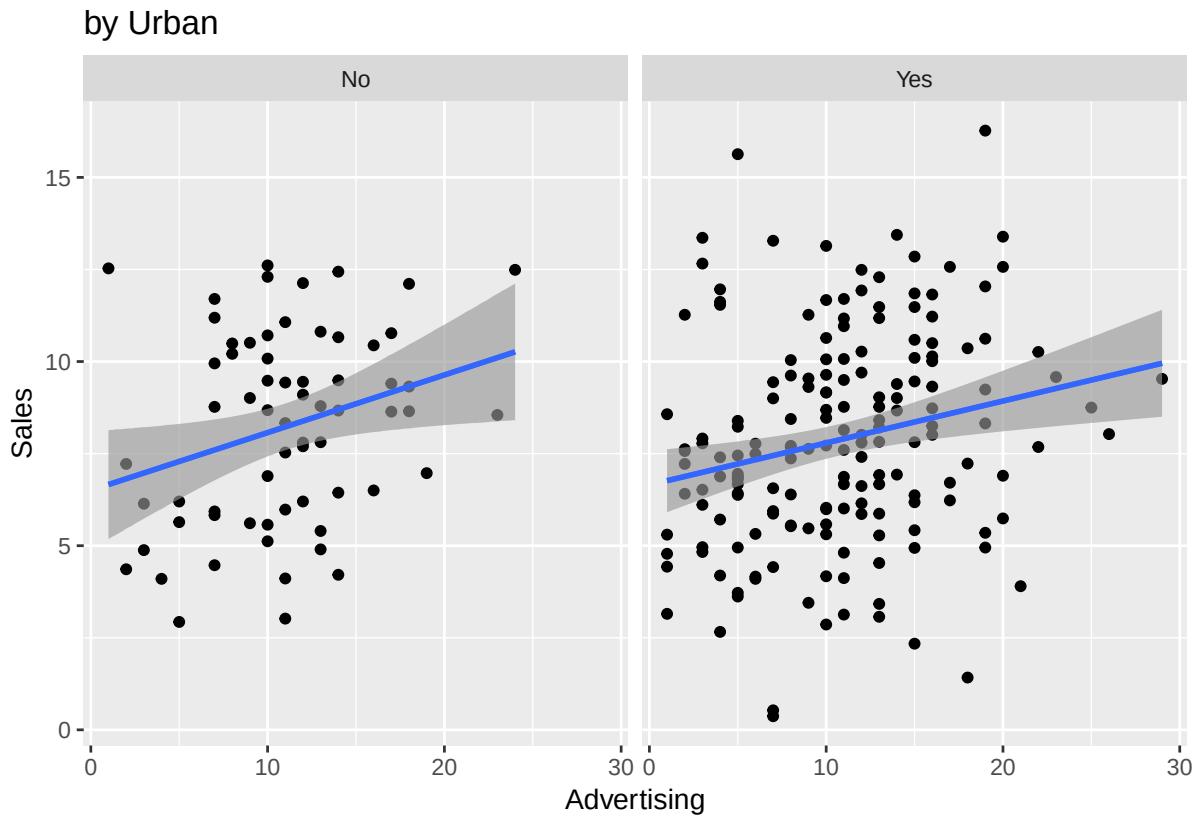
1. 해당 지역의 특성: `Income`, `Population`, `Age`, `Education`, **`Urban`**, **`US`**
2. 해당 지역의 biz의 특성: `CompPrice`, `Price`, **`ShelveLoc`**

- 해당 지역의 특성에 따라서 광고 효과를 살펴봅니다.

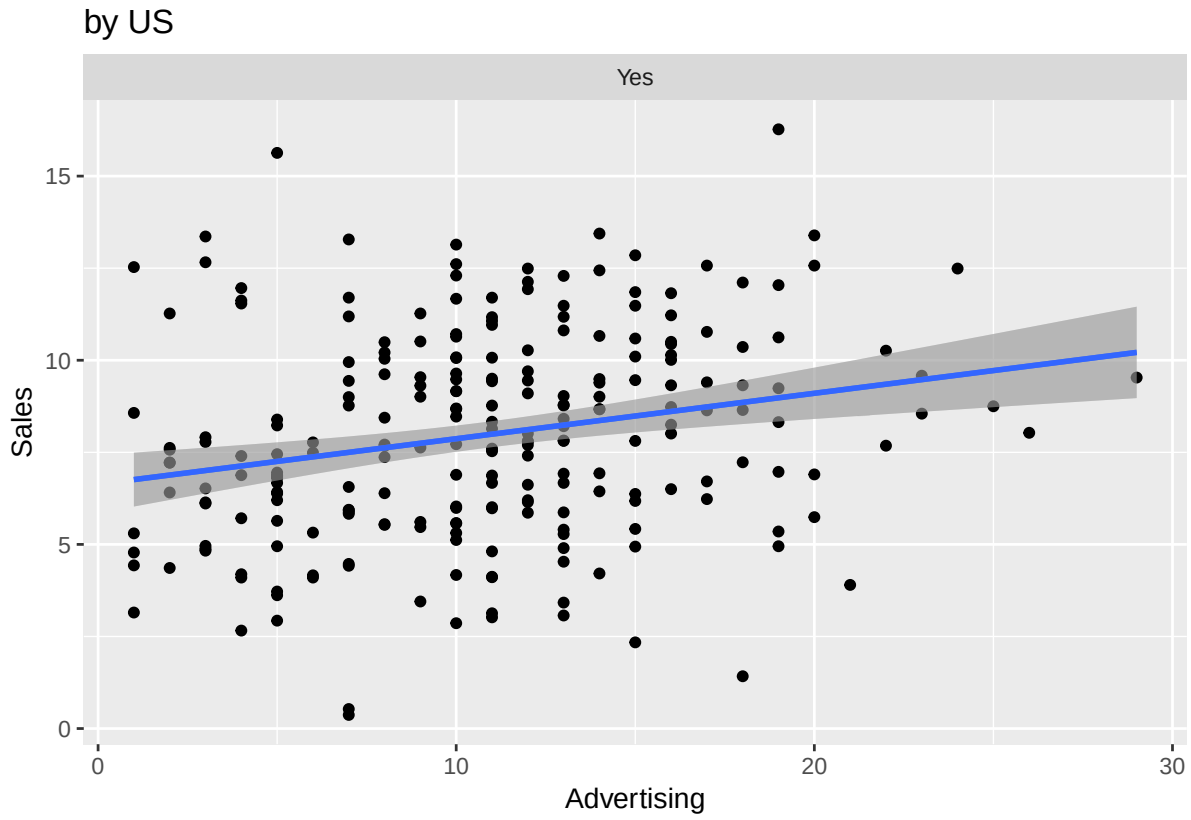
1. Factor 변수에 따라서 광고 효과가 달라지는지 살펴봅니다.
2. Factor 변수가 아닌 경우에는 4분위 값을 기준으로 Factor 변수로 변환하여 살펴봅니다.

2. Factor 변수 분석

```
xyPlot + facet_wrap(~ Urban) + ggtitle("by Urban") + stat_smooth(method="lm")
```




```
xyPlot + facet_wrap(~ US) + ggtitle("by US") + stat_smooth(method="lm")
```



3. Non-Factor 변수 분석

1. Mutate Factors

Method 1 - basic method

```
Carseats <- Carseats %>%
  mutate(
    IncomeF =
      ifelse(Income < summary(Carseats$Income)[2], "Q1",
             ifelse(Income < summary(Carseats$Income)[3], "Q2",
                    ifelse(Income < summary(Carseats$Income)[5], "Q3", "Q4")))) %>%
  mutate(
    AgeF =
      ifelse(Age < summary(Carseats$Age)[2], "Q1",
             ifelse(Age < summary(Carseats$Age)[3], "Q2",
                    ifelse(Age < summary(Carseats$Age)[5], "Q3", "Q4"))))
```

Method 2 - using function

```
generateQuartile <- function(x) {
  return(summary(x)[c(2,3,5)])
}
Carseats <- Carseats %>%
  mutate(
    IncomeF =
      ifelse(Income < generateQuartile(Carseats$Income)[1], "Q1",
             ifelse(Income < generateQuartile(Carseats$Income)[2], "Q2",
                    ifelse(Income < generateQuartile(Carseats$Income)[3], "Q3", "Q4")))) %>%
  mutate(
    AgeF =
      ifelse(Age < generateQuartile(Carseats$Age)[1], "Q1",
```

```
            ifelse(Age < generateQuartile(Carseats$Age)[2], "Q2",
                   ifelse(Age < generateQuartile(Carseats$Age)[3], "Q3", "Q4")))) %>%
  mutate(
    EducationF =
      ifelse(Education < generateQuartile(Carseats$Education)[1], "Q1",
            ifelse(Education < generateQuartile(Carseats$Education)[2], "Q2",
                   ifelse(Education < generateQuartile(Carseats$Education)[3], "Q3", "Q4")))) %>%
  mutate(
    PopulationF =
      ifelse(Population < generateQuartile(Carseats$Population)[1], "Q1",
            ifelse(Population < generateQuartile(Carseats$Population)[2], "Q2",
                   ifelse(Population < generateQuartile(Carseats$Population)[3], "Q3", "Q4"))))
```

2. Draw Plots

Method 1 - basic method

```
xyPlot <- ggplot(Carseats, aes(x = Advertising, y = Sales)) +
  geom_point() + stat_smooth(method="lm")
  # Let xyPlot to update the dataset
xyPlot + facet_wrap(~ IncomeF) + ggtitle("by Income") +
  labs(subtitle = generateQuartile(Carseats$Income) %>% paste(collapse = ","))
xyPlot + facet_wrap(~ AgeF) + ggtitle("by Age") +
  labs(subtitle = generateQuartile(Carseats$Age) %>% paste(collapse = ","))
xyPlot + facet_wrap(~ EducationF) + ggtitle("by Education") +
  labs(subtitle = generateQuartile(Carseats$Education) %>% paste(collapse = ","))
xyPlot + facet_wrap(~ PopulationF) + ggtitle("by Population") +
  labs(subtitle = generateQuartile(Carseats$Population) %>% paste(collapse = ","))
```

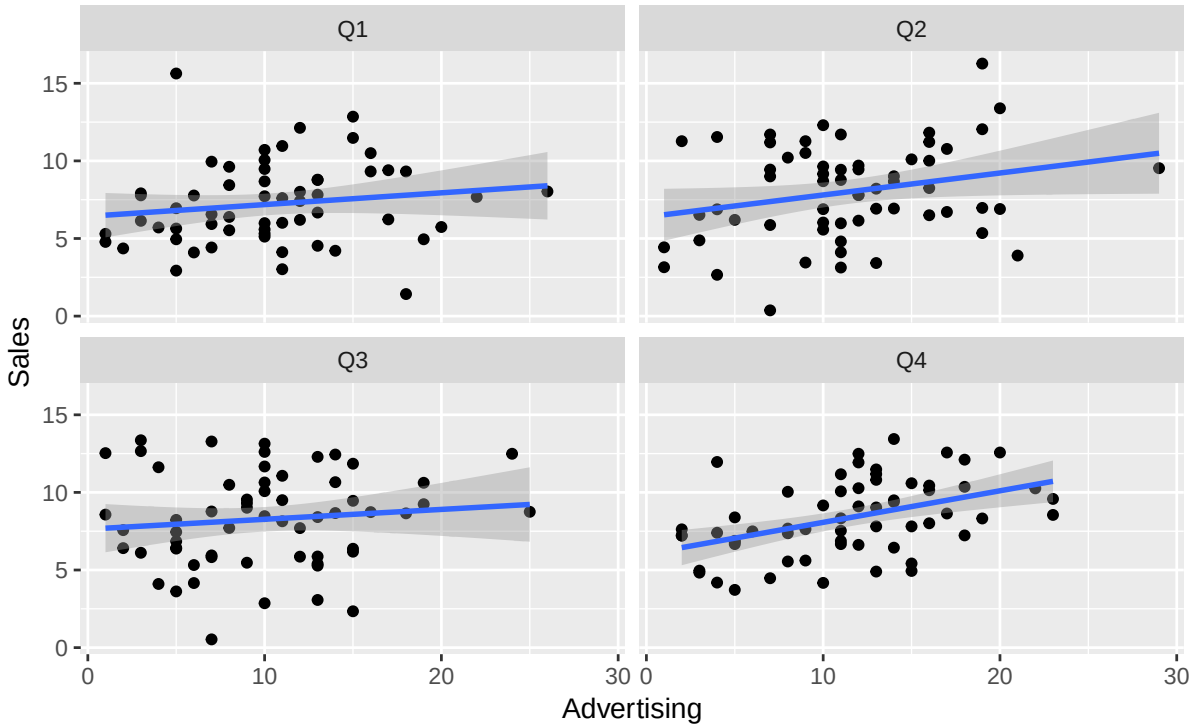
Method 2 - advanced - using for loop

- Variable의 갯수에 많아져도 코드가 길어지지 않도록 처리합니다.
- 반복문 안에서 `facet_wrap(as.formula(paste0("~", var, "F")))`를 사용하여 string 값인 var를 변수 입력으로 사용할 수 있습니다.
- 같은 목적으로 `generateQuartile(eval(parse(text=paste0("Carseats$", var))))`를 사용하여 var를 변수 입력으로 사용할 수 있습니다.

```
xyPlot <-
  ggplot(data = Carseats, aes(x = Advertising, y = Sales)) +
  geom_point() + stat_smooth(method="lm")
  # Let xyPlot to update the dataset
variables <- c("Income", "Population", "Age", "Education")
for (var in variables) {
  a <- xyPlot +
    facet_wrap(as.formula(paste0("~", var, "F"))) +
    labs(title =
           paste0("by ", var, "F"),
         subtitle =
           generateQuartile(eval(parse(text=paste0("Carseats$", var)))) %>%
           as.numeric() %>% paste(collapse = ","))
  print(a)
}
```

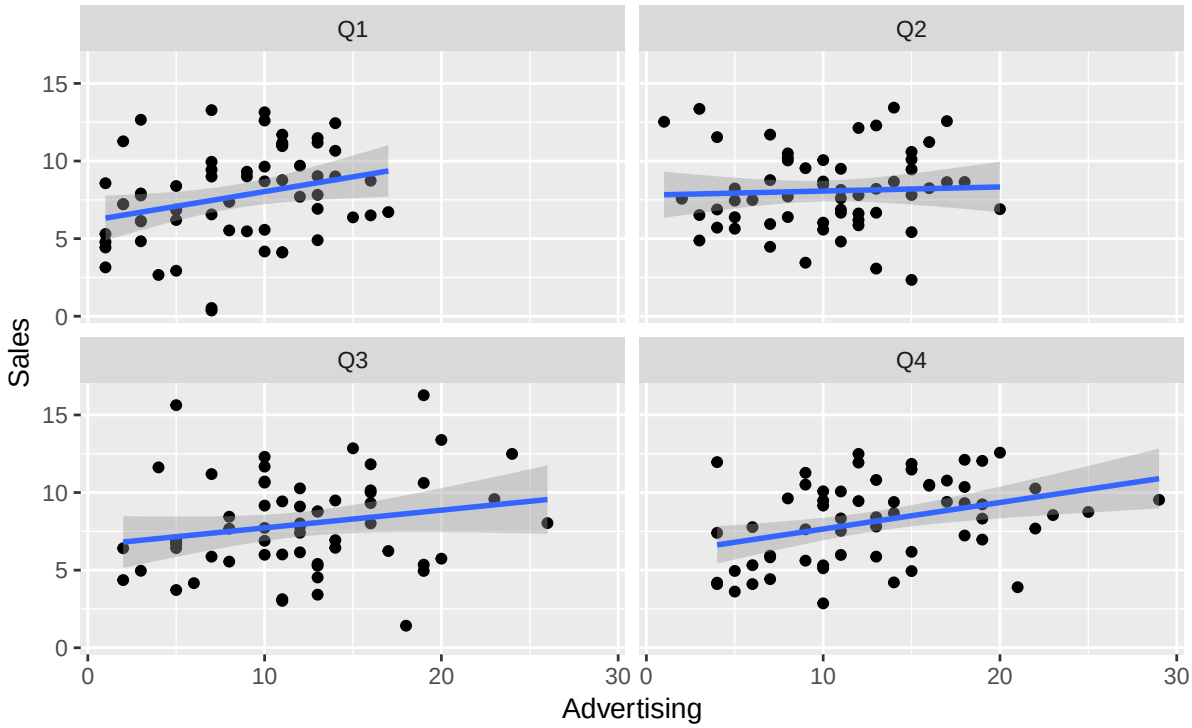
by IncomeF

44,70,93



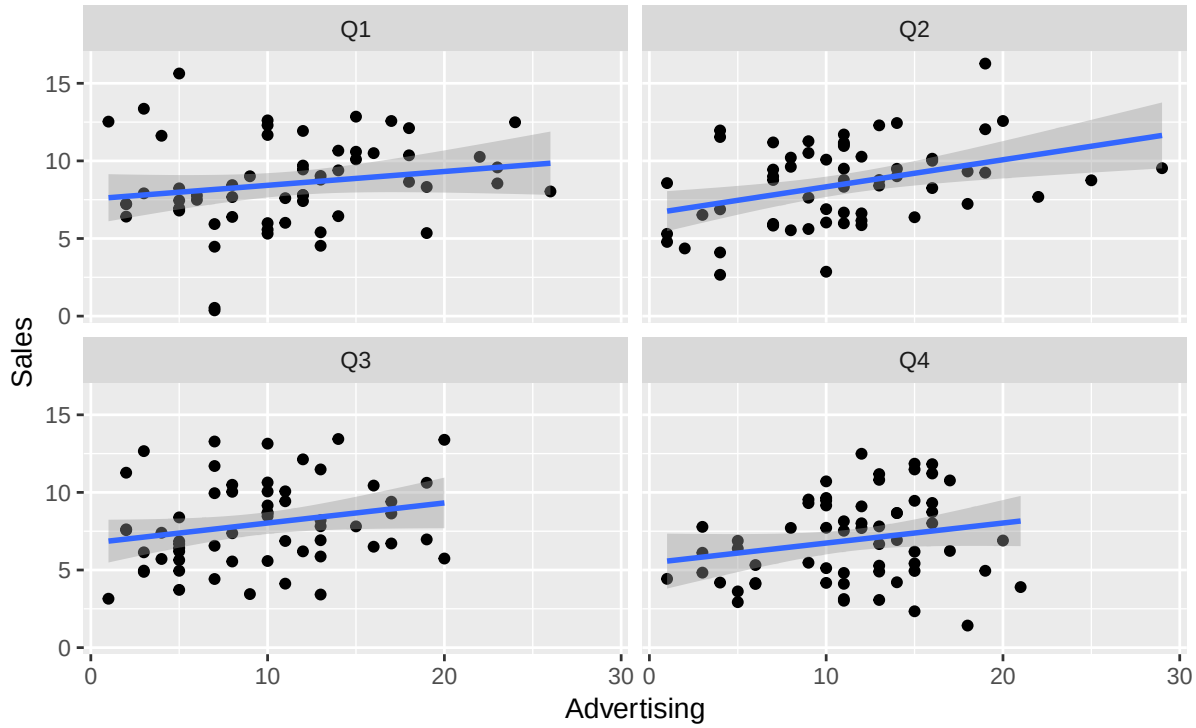
by PopulationF

159,289,401



by AgeF

41,54,65



by EducationF

11,14,16