

VADeaths

20173204 광명빈

Tidy Data

깔끔한(tidy) 데이터를 만드는 방법에 대하여 알아본다. 사용되는 데이터는 R 에 내장되어 있는 VADeaths 이다. 이 데이터의 구조는 5세 간격의 연령대를 행의 이름으로 하고, 장소(Rural, Urban)와 성별(Male, Female)의 조합을 열의 이름으로 갖는 행렬임을 알 수 있다.

VADeaths

##	Rural Male	Rural Female	Urban Male	Urban Female
## 50-54	11.7	8.7	15.4	8.4
## 55-59	18.1	11.7	24.3	13.6
## 60-64	26.9	20.3	37.0	19.3
## 65-69	41.0	30.9	54.6	35.1
## 70-74	66.0	54.3	71.1	50.0

str(VADeaths)

```
## num [1:5, 1:4] 11.7 18.1 26.9 41 66 8.7 11.7 20.3 30.9 54.3 ...
## - attr(*, "dimnames")=List of 2
## ..$ : chr [1:5] "50-54" "55-59" "60-64" "65-69" ...
## ..$ : chr [1:4] "Rural Male" "Rural Female" "Urban Male" "Urban Female"
```

Base R 의 도구 활용

왜 이 데이터가 깔끔하지(tidy) 않은지 생각해 보자. 데이터를 어떻게 표현해야 깔끔한 것인지 최종 결과물과 비교한다.

c() 는 행렬 구조로 표현한 VADeaths 를 기다란 하나의 벡터로 나타낸다. 이렇게 만든 한 줄의 벡터를 Rates 에 옮겨 놓는다.

보통 ordered() 가 아닌 factor() 를 사용하는 경우가 많은데 연령이라는 변수의 특성을 감안하면 단순히 명목형이 아니고 엄연히 순서가 있기 때문에 ordered() 를 사용하는 것이 적절하다.

```
Rates <- c(VADeaths) ## 행렬을 한 줄의 벡터로 변환
N <- length(Rates) ## 'Rates'의 크기를 'N'으로 저장.
Age <- ordered(rownames(VADeaths)) # 행 이름으로 주어진 글자 벡터, 연령대를 순서형 범주로 변환.
Age <- rep(ordered(rownames(VADeaths)), # 전체 관찰 수효 만큼 반복. 'length.out = '의 용례에 유의.
          length.out = N)
Place <- gl(2, 10, N, # 농촌, 도시의 두 수준을 10번씩 반복하는 'factor' 설정
          labels = c("Rural", "Urban"))
Gender <- gl(2, 5, N, # 성별은 5번씩 반복
          labels = c("Male", "Female"))
data.frame(Age, Place, Gender, Rates) # 각 벡터를 데이터 프레임의 요소로 편성
```

```
##      Age Place Gender Rates
## 1  50-54 Rural   Male  11.7
## 2  55-59 Rural   Male  18.1
## 3  60-64 Rural   Male  26.9
## 4  65-69 Rural   Male  41.0
## 5  70-74 Rural   Male  66.0
## 6  50-54 Rural Female   8.7
## 7  55-59 Rural Female  11.7
## 8  60-64 Rural Female  20.3
## 9  65-69 Rural Female  30.9
## 10 70-74 Rural Female  54.3
## 11 50-54 Urban   Male  15.4
## 12 55-59 Urban   Male  24.3
## 13 60-64 Urban   Male  37.0
## 14 65-69 Urban   Male  54.6
## 15 70-74 Urban   Male  71.1
## 16 50-54 Urban Female   8.4
## 17 55-59 Urban Female  13.6
## 18 60-64 Urban Female  19.3
## 19 65-69 Urban Female  35.1
## 20 70-74 Urban Female  50.0
```

```
VADeaths_df <- data.frame(Age, Place, Gender, Rates) # 데이터 프레임을 새로운 R 객체로 지정
VADeaths_df # 데이터 프레임 출력
```

```
##      Age Place Gender Rates
## 1  50-54 Rural   Male  11.7
## 2  55-59 Rural   Male  18.1
## 3  60-64 Rural   Male  26.9
## 4  65-69 Rural   Male  41.0
## 5  70-74 Rural   Male  66.0
## 6  50-54 Rural Female   8.7
## 7  55-59 Rural Female  11.7
## 8  60-64 Rural Female  20.3
## 9  65-69 Rural Female  30.9
## 10 70-74 Rural Female  54.3
## 11 50-54 Urban   Male  15.4
## 12 55-59 Urban   Male  24.3
## 13 60-64 Urban   Male  37.0
## 14 65-69 Urban   Male  54.6
## 15 70-74 Urban   Male  71.1
## 16 50-54 Urban Female   8.4
## 17 55-59 Urban Female  13.6
## 18 60-64 Urban Female  19.3
## 19 65-69 Urban Female  35.1
## 20 70-74 Urban Female  50.0
```

```
str(VADeaths_df) # 데이터 프레임 구조 파악
```

```
## 'data.frame':   20 obs. of  4 variables:
## $ Age      : Ord.factor w/ 5 levels "50-54"<"55-59"<..: 1 2 3 4 5 1 2 3 4 5 ...
## $ Place    : Factor w/ 2 levels "Rural","Urban": 1 1 1 1 1 1 1 1 1 1 ...
## $ Gender   : Factor w/ 2 levels "Male","Female": 1 1 1 1 1 2 2 2 2 2 ...
## $ Rates    : num  11.7 18.1 26.9 41 66 8.7 11.7 20.3 30.9 54.3 ...
```

VADeaths 를 table 구조로 변환하고, as.data.frame 을 적용할 수도 있으나 Place 와 Gender 를 다시 분리하여야 함.

```
as.data.frame(as.table(VADeaths))
```

```
##      Var1      Var2 Freq
## 1 50-54   Rural Male 11.7
## 2 55-59   Rural Male 18.1
## 3 60-64   Rural Male 26.9
## 4 65-69   Rural Male 41.0
## 5 70-74   Rural Male 66.0
## 6 50-54 Rural Female  8.7
## 7 55-59 Rural Female 11.7
## 8 60-64 Rural Female 20.3
## 9 65-69 Rural Female 30.9
## 10 70-74 Rural Female 54.3
## 11 50-54   Urban Male 15.4
## 12 55-59   Urban Male 24.3
## 13 60-64   Urban Male 37.0
## 14 65-69   Urban Male 54.6
## 15 70-74   Urban Male 71.1
## 16 50-54 Urban Female  8.4
## 17 55-59 Urban Female 13.6
## 18 60-64 Urban Female 19.3
## 19 65-69 Urban Female 35.1
## 20 70-74 Urban Female 50.0
```

혹은 한 번에

```
as.data.frame.table(VADeaths)
```

```
##      Var1      Var2 Freq
## 1 50-54   Rural Male 11.7
## 2 55-59   Rural Male 18.1
## 3 60-64   Rural Male 26.9
## 4 65-69   Rural Male 41.0
## 5 70-74   Rural Male 66.0
## 6 50-54 Rural Female  8.7
## 7 55-59 Rural Female 11.7
## 8 60-64 Rural Female 20.3
## 9 65-69 Rural Female 30.9
## 10 70-74 Rural Female 54.3
## 11 50-54   Urban Male 15.4
## 12 55-59   Urban Male 24.3
## 13 60-64   Urban Male 37.0
## 14 65-69   Urban Male 54.6
## 15 70-74   Urban Male 71.1
## 16 50-54 Urban Female  8.4
## 17 55-59 Urban Female 13.6
## 18 60-64 Urban Female 19.3
## 19 65-69 Urban Female 35.1
## 20 70-74 Urban Female 50.0
```

tidyverse를 이용한 방법

다음 코드를 차례대로 실행하면서 어떤 흐름이 잡히는 지 살펴보세요.

경고문의 Conflicts ... 이하는 R Base 에 있는 filter() 나 lag() 함수를 사용하려면 구체적으로 stats::filter() 나 stats::lag() 라고 하여야 한다는 것을 의미한다.

```
library(tidyverse) # `tidyverse`를 검색 경로에 올려 놓음. 함께 불러들이는 패키지들과 경고문에 유의.
```

```
## -- Attaching packages ----- tidyverse 1.3.0 --
```

```
## √ ggplot2 3.3.2      √ purrr  0.3.4
## √ tibble  3.0.3      √ dplyr  1.0.2
## √ tidyr   1.1.1      √ stringr 1.4.0
## √ readr   1.3.1      √ forcats 0.5.0
```

```
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
VADeaths_tbl <- VADeaths %>% # 최종 결과물을 `tibble` 형식으로 지정.
  as_tibble() %>% # 행렬 구조를 `tibble` 구조로 변환. `tbl_df()`는 더 이상 사용되지 않음.
  mutate(Age = row.names(VADeaths)) %>% # 행 이름으로 주어진 연령대를 글자벡터로 생성
  gather(key = Place_Gender, # `Age`를 제외한 나머지 변수를 `key, value`쌍으로 정리하면서 새로운 변수명 부여.
         value = Rates,
         -Age) %>%
  separate(Place_Gender, c("Place", "Gender"), # `Place_Gender`를 `Place`와 `Gender`로 분리.
           sep = " ") %>%
  mutate(Age = ordered(Age), # `Age`, `Place`, `Gender`를 순서형 범주와 명목형 범주로 변환
         Place = factor(Place),
         Gender = factor(Gender, # `Gender`에서 `level` = `를 설정하지 않으면 알파벳 순에 따라 수준이 정해짐.
                        levels = c("Male", "Female"))) # 즉, `Female`이 1, `Male`이 2가 됨.
VADeaths_tbl # `tibble` 형식으로 출력
```

```
## # A tibble: 20 x 4
##   Age   Place Gender Rates
##   <ord> <fct> <fct>   <dbl>
## 1 50-54 Rural Male    11.7
## 2 55-59 Rural Male    18.1
## 3 60-64 Rural Male    26.9
## 4 65-69 Rural Male     41
## 5 70-74 Rural Male    66
## 6 50-54 Rural Female   8.7
## 7 55-59 Rural Female  11.7
## 8 60-64 Rural Female  20.3
## 9 65-69 Rural Female  30.9
## 10 70-74 Rural Female  54.3
## 11 50-54 Urban Male    15.4
## 12 55-59 Urban Male    24.3
## 13 60-64 Urban Male     37
## 14 65-69 Urban Male    54.6
## 15 70-74 Urban Male    71.1
## 16 50-54 Urban Female   8.4
## 17 55-59 Urban Female  13.6
## 18 60-64 Urban Female  19.3
## 19 65-69 Urban Female  35.1
## 20 70-74 Urban Female   50
```

```
str(VADeaths_tbl) # 구조 파악.
```

```
## tibble [20 x 4] (S3: tbl_df/tbl/data.frame)
## $ Age : Ord.factor w/ 5 levels "50-54"<"55-59"<..: 1 2 3 4 5 1 2 3 4 5 ...
## $ Place : Factor w/ 2 levels "Rural","Urban": 1 1 1 1 1 1 1 1 1 1 ...
## $ Gender: Factor w/ 2 levels "Male","Female": 1 1 1 1 1 2 2 2 2 2 ...
## $ Rates : num [1:20] 11.7 18.1 26.9 41 66 8.7 11.7 20.3 30.9 54.3 ...
```

이 과정을 순서대로 살펴보면, 먼저 행렬 구조를 tibble 형식으로 변환하고,

```
VADeaths %>%
  as_tibble()
```

```
## # A tibble: 5 x 4
##   `Rural Male` `Rural Female` `Urban Male` `Urban Female`
##           <dbl>           <dbl>           <dbl>           <dbl>
## 1          11.7             8.7             15.4             8.4
## 2          18.1            11.7             24.3            13.6
## 3          26.9            20.3              37             19.3
## 4           41             30.9            54.6            35.1
## 5           66             54.3            71.1             50
```

Age 변수 생성

```
VADeaths %>%
  as_tibble() %>%
  mutate(Age = rownames(VADeaths))
```

```
## # A tibble: 5 x 5
##   `Rural Male` `Rural Female` `Urban Male` `Urban Female` Age
##   <dbl>         <dbl>         <dbl>         <dbl> <chr>
## 1      11.7         8.7         15.4         8.4 50-54
## 2      18.1        11.7         24.3        13.6 55-59
## 3      26.9        20.3         37          19.3 60-64
## 4       41         30.9        54.6        35.1 65-69
## 5       66         54.3        71.1         50 70-74
```

Age 를 제외한 변수를 key, value 쌍으로 정리하면서 새로운 변수명 부여, Age 의 새로운 위치에 유의

```
VADeaths %>%
  as_tibble() %>%
  mutate(Age = rownames(VADeaths)) %>%
  gather(key = Place_Gender,
         value = Rates,
         -Age)
```

```
## # A tibble: 20 x 3
##   Age Place_Gender Rates
##   <chr> <chr>         <dbl>
## 1 50-54 Rural Male    11.7
## 2 55-59 Rural Male    18.1
## 3 60-64 Rural Male    26.9
## 4 65-69 Rural Male    41
## 5 70-74 Rural Male    66
## 6 50-54 Rural Female  8.7
## 7 55-59 Rural Female 11.7
## 8 60-64 Rural Female 20.3
## 9 65-69 Rural Female 30.9
## 10 70-74 Rural Female 54.3
## 11 50-54 Urban Male   15.4
## 12 55-59 Urban Male   24.3
## 13 60-64 Urban Male   37
## 14 65-69 Urban Male   54.6
## 15 70-74 Urban Male   71.1
## 16 50-54 Urban Female  8.4
## 17 55-59 Urban Female 13.6
## 18 60-64 Urban Female 19.3
## 19 65-69 Urban Female 35.1
## 20 70-74 Urban Female 50
```

Place_Gender 를 Place 와 Gender 로 분리. sep = 의 사용 방법에 유의.

```
VADeaths %>%
  as_tibble() %>%
  mutate(Age = rownames(VADeaths)) %>%
  gather(key = Place_Gender,
         value = Rates,
         -Age) %>%
  separate(Place_Gender, c("Place", "Gender"),
         sep = " ")
```

```
## # A tibble: 20 x 4
##   Age   Place Gender Rates
##   <chr> <chr> <chr>   <dbl>
## 1 50-54 Rural Male    11.7
## 2 55-59 Rural Male    18.1
## 3 60-64 Rural Male    26.9
## 4 65-69 Rural Male     41
## 5 70-74 Rural Male    66
## 6 50-54 Rural Female   8.7
## 7 55-59 Rural Female  11.7
## 8 60-64 Rural Female  20.3
## 9 65-69 Rural Female  30.9
## 10 70-74 Rural Female  54.3
## 11 50-54 Urban Male    15.4
## 12 55-59 Urban Male    24.3
## 13 60-64 Urban Male     37
## 14 65-69 Urban Male    54.6
## 15 70-74 Urban Male    71.1
## 16 50-54 Urban Female   8.4
## 17 55-59 Urban Female  13.6
## 18 60-64 Urban Female  19.3
## 19 65-69 Urban Female  35.1
## 20 70-74 Urban Female   50
```

각 구성요소를 특성에 맞게 변환. Gender 의 경우 levels = 를 설정하는 이유에 대하여 생각해 볼 것.

```
VADeaths %>%
  as_tibble() %>%
  mutate(Age = rownames(VADeaths)) %>%
  gather(key = Place_Gender,
         value = Rates,
         -Age) %>%
  separate(Place_Gender, c("Place", "Gender"),
         sep = " ") %>%
  mutate(Age = ordered(Age),
         Place = factor(Place),
         Gender = factor(Gender,
                        levels = c("Male", "Female")))
```

```
## # A tibble: 20 x 4
##   Age   Place Gender Rates
##   <ord> <fct> <fct>   <dbl>
## 1 50-54 Rural Male    11.7
## 2 55-59 Rural Male    18.1
## 3 60-64 Rural Male    26.9
## 4 65-69 Rural Male    41
## 5 70-74 Rural Male    66
## 6 50-54 Rural Female   8.7
## 7 55-59 Rural Female  11.7
## 8 60-64 Rural Female  20.3
## 9 65-69 Rural Female  30.9
## 10 70-74 Rural Female  54.3
## 11 50-54 Urban Male    15.4
## 12 55-59 Urban Male    24.3
## 13 60-64 Urban Male    37
## 14 65-69 Urban Male    54.6
## 15 70-74 Urban Male    71.1
## 16 50-54 Urban Female   8.4
## 17 55-59 Urban Female  13.6
## 18 60-64 Urban Female  19.3
## 19 65-69 Urban Female  35.1
## 20 70-74 Urban Female  50
```

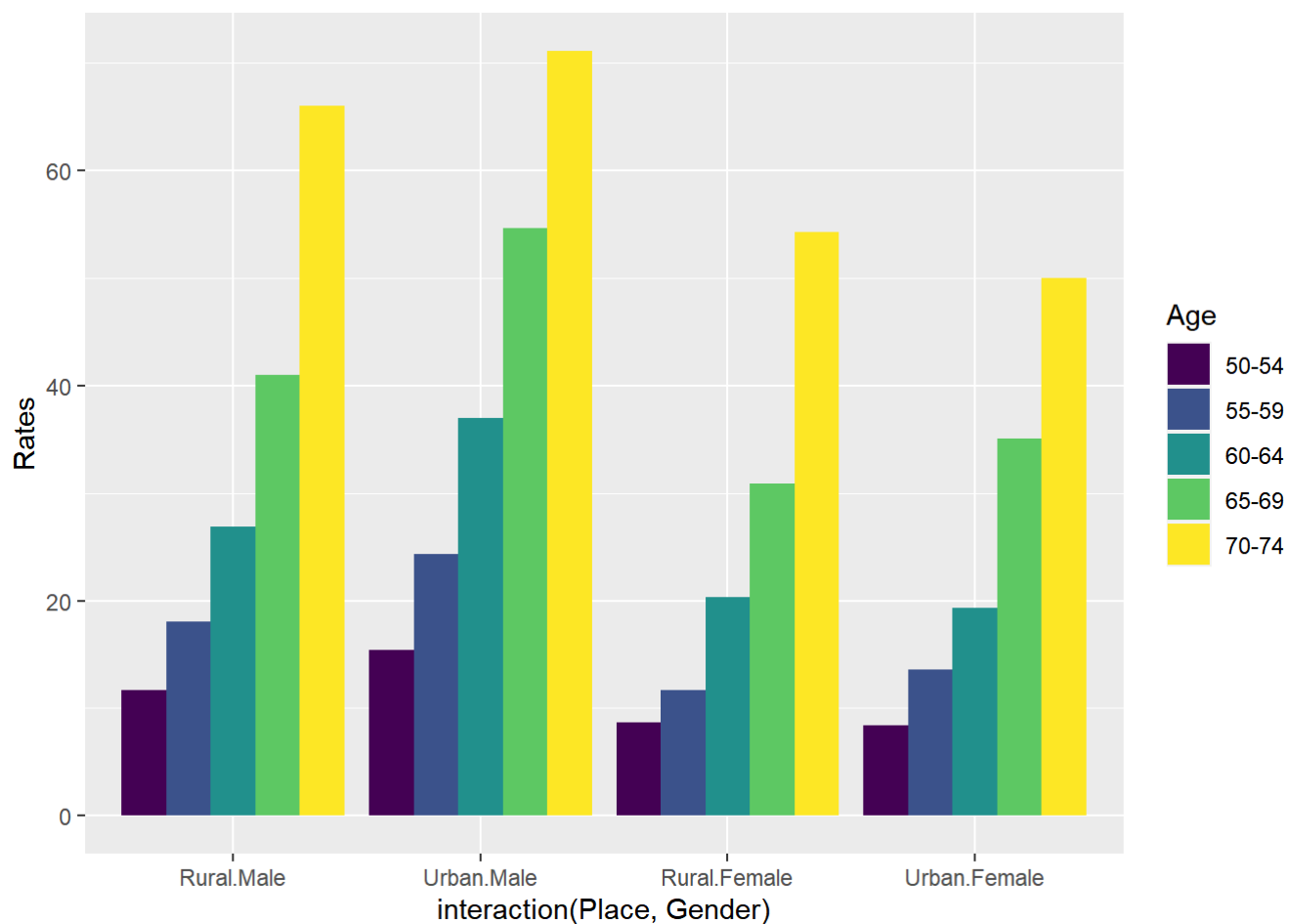
Plots

이 데이터 프레임을 시각적으로 `ggplot()` 을 이용하여 표현하는 방법에 대하여 생각해 보자. 먼저 기본 함수들을 이용하여 생성한 `VADeaths_df` 를 이용하여 그려보면, `data = VADeaths_df` 로 설정하고, `aes()` 의 `x =` 에는 장소(`Place`)와 성별(`Gender`)의 조합인 농촌남성(`Rural.Male`), 도시남성(`Urban.Male`), 농촌여성(`Rural.Female`), 도시여성(`Urban.Female`)을 `interaction(Place, Gender)` 로 나타낸다. `y =` 에는 사망률(`Rates`)을, 각 연령대(`Age`)를 막대의 색깔(`fill =`)로 구분한다.

막대그래프로 표현하기 위하여 `geom_bar()` 를 사용하였는데, 가장 간단한 형식으로 나타내었다. 추가 정보나 보다 세부적인 표현은 다음에 다루기로 한다.

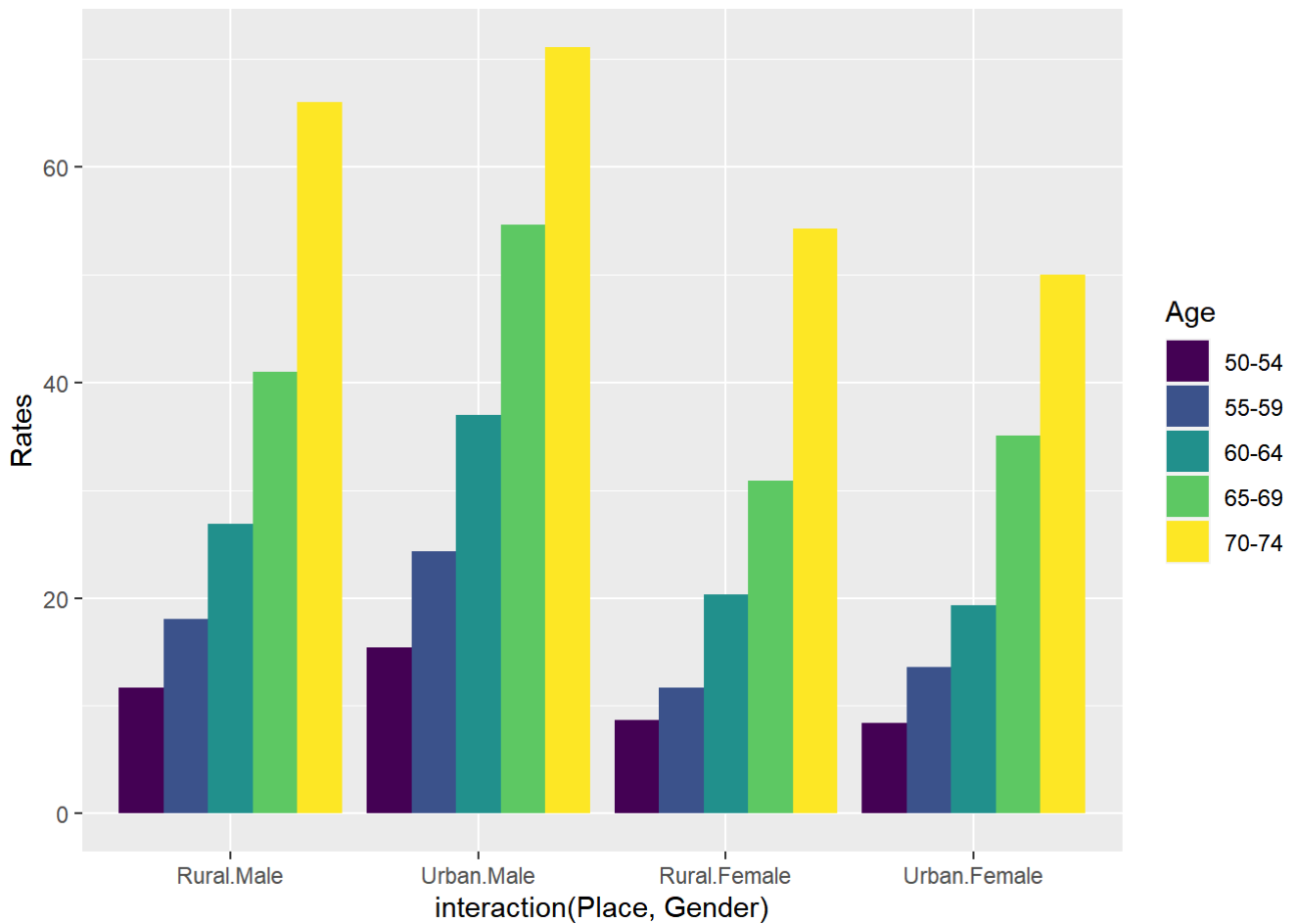
도시남성들의 사망률이 전 연령대에서 고르게 가장 높게 나타나는 반면, 도시 여성들은 대부분의 연령대에서 사망률이 낮게 나타나고 있다. 도시에 사는 남성들 ...

```
ggplot(data = VADeaths_df,
       mapping = aes(x = interaction(Place, Gender),
                     y = Rates,
                     fill = Age)) +
geom_bar(stat = "identity",
        position = position_dodge())
```

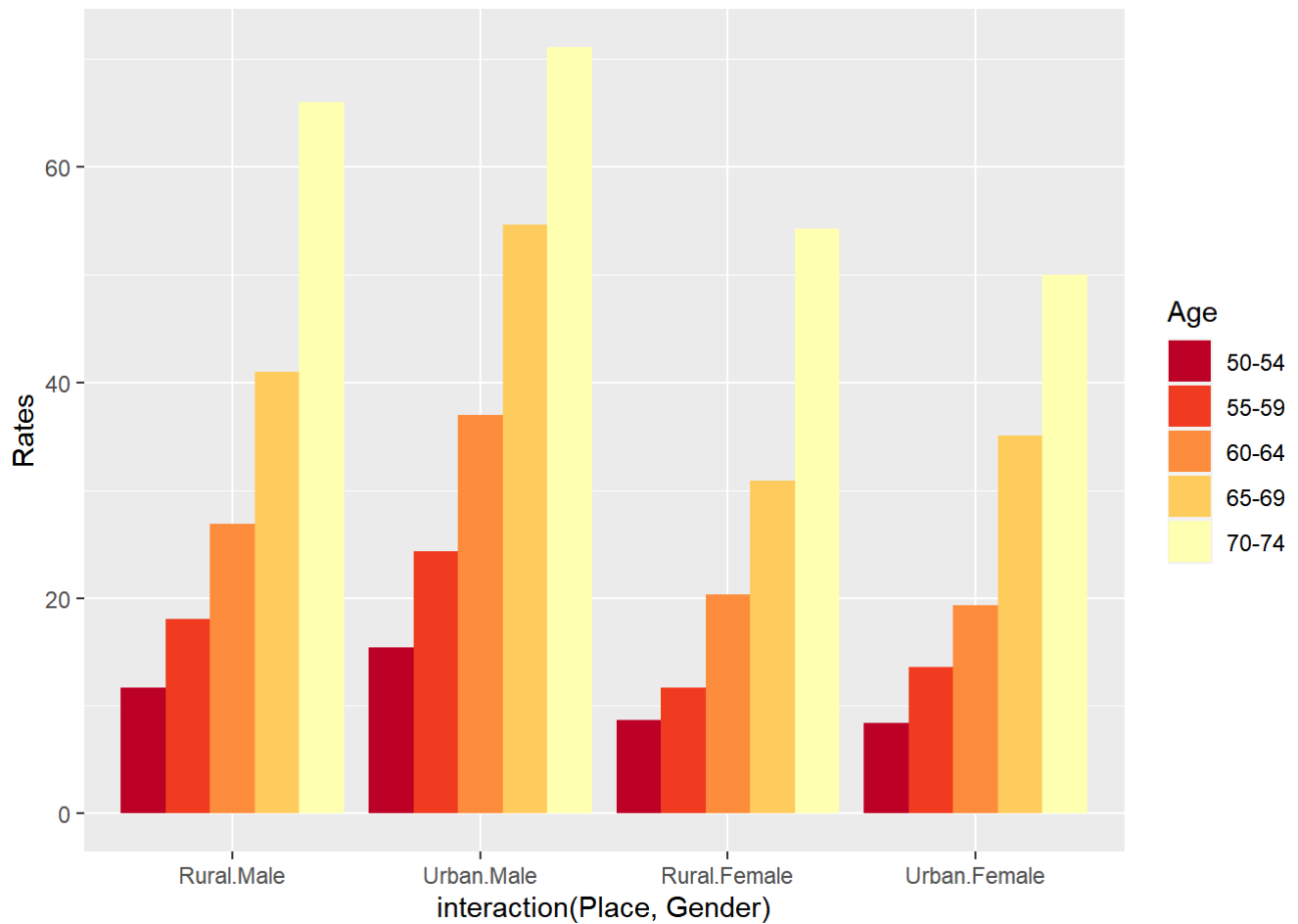
동일한 내용을 VADeaths_tbl 로 그리면,

```
ggplot(data = VADeaths_tbl,
       mapping = aes(x = interaction(Place, Gender),
                     y = Rates,
                     fill = Age)) +
geom_bar(stat = "identity",
        position = position_dodge())
```



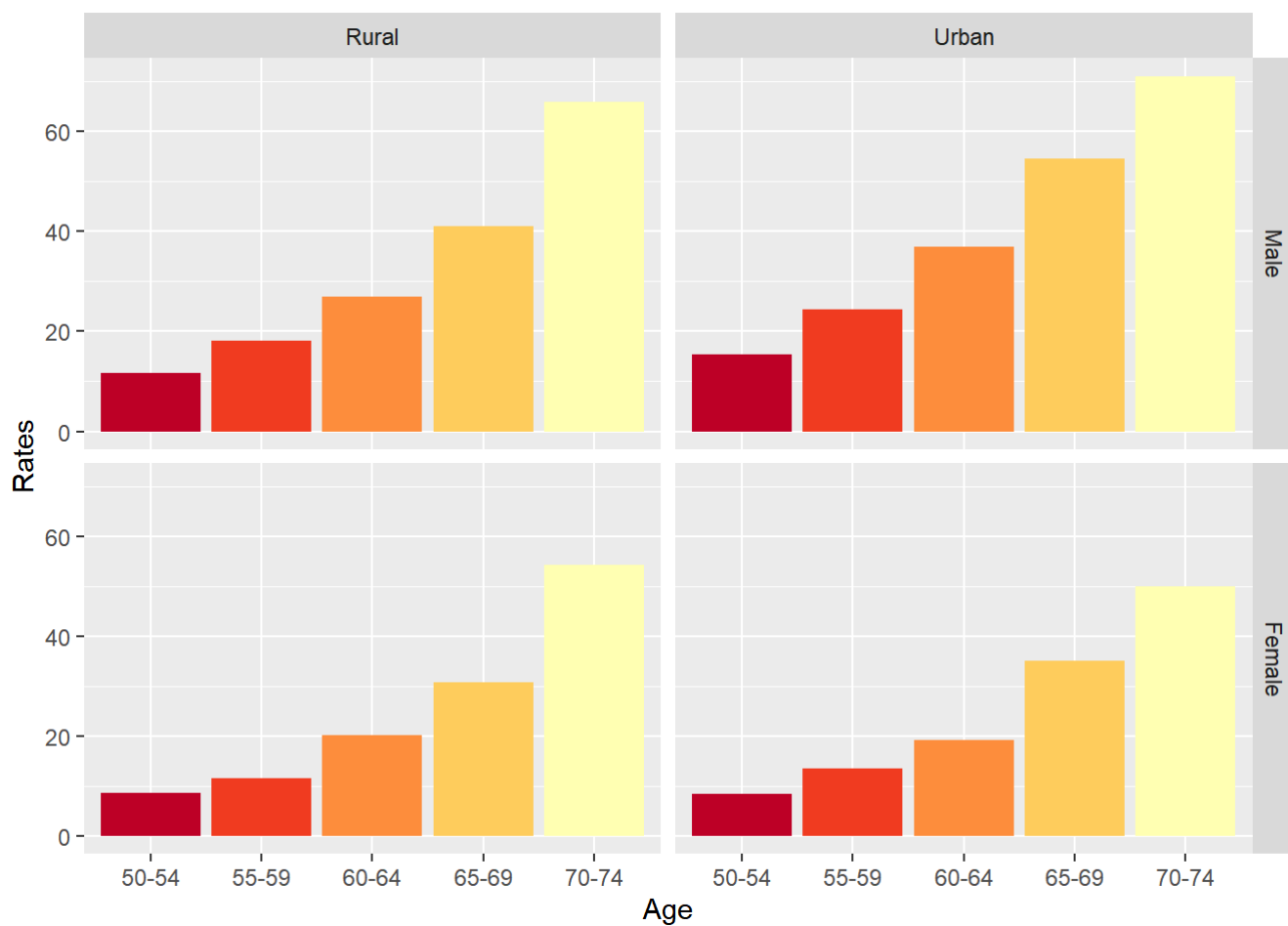
막대의 색깔을 Sequential 팔레트 계열(`scale_fill_brewer` 도움말 참조)의 색깔 중 연령대의 변화에 맞도록 조정하면,

```
ggplot(data = VADeaths_tbl,
       mapping = aes(x = interaction(Place, Gender),
                     y = Rates,
                     fill = Age)) +
  geom_bar(stat = "identity",
          position = position_dodge()) +
  scale_fill_brewer(palette = "YlOrRd",
                   direction = -1)
```



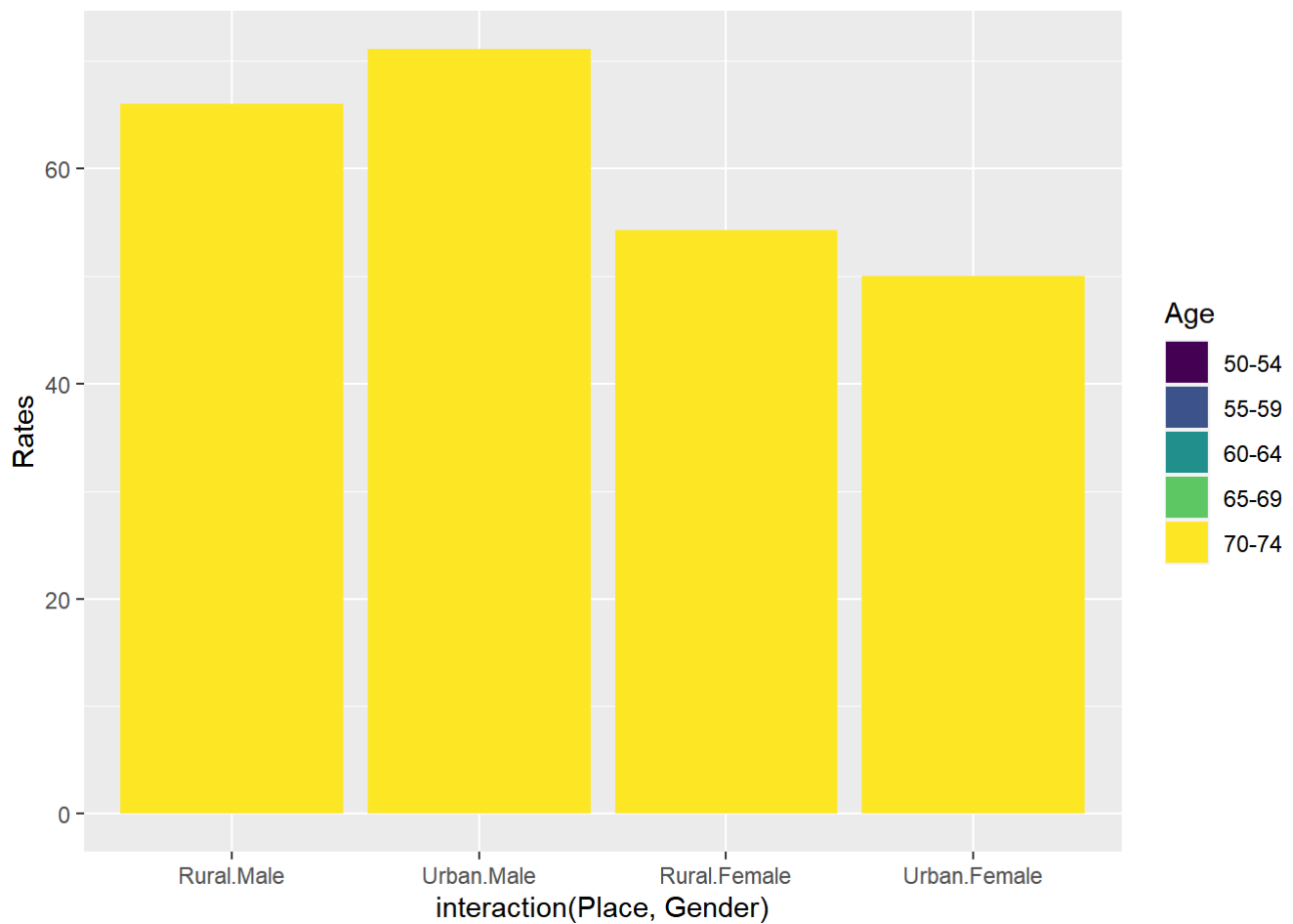
facet_grid 를 이용하여 패널로 구분하여 나타내면,

```
ggplot(data = VADeaths_tbl,
        mapping = aes(x = Age,
                      y = Rates,
                      fill = Age)) +
  geom_bar(stat = "identity",
           position = position_dodge()) +
  scale_fill_brewer(guide = FALSE,
                   palette = "YlOrRd",
                   direction = -1) +
  facet_grid(Gender ~ Place)
```

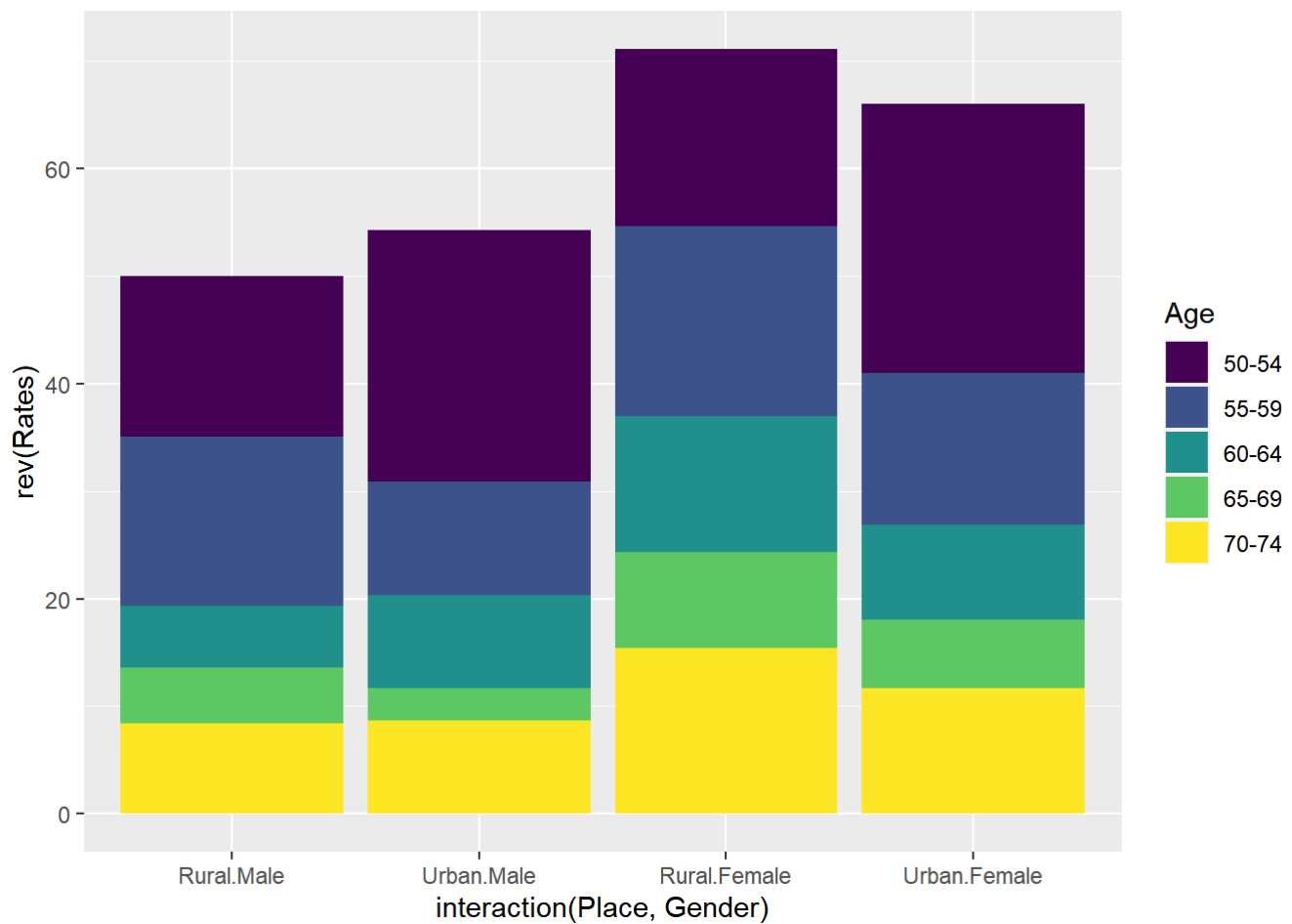


position = "identity"

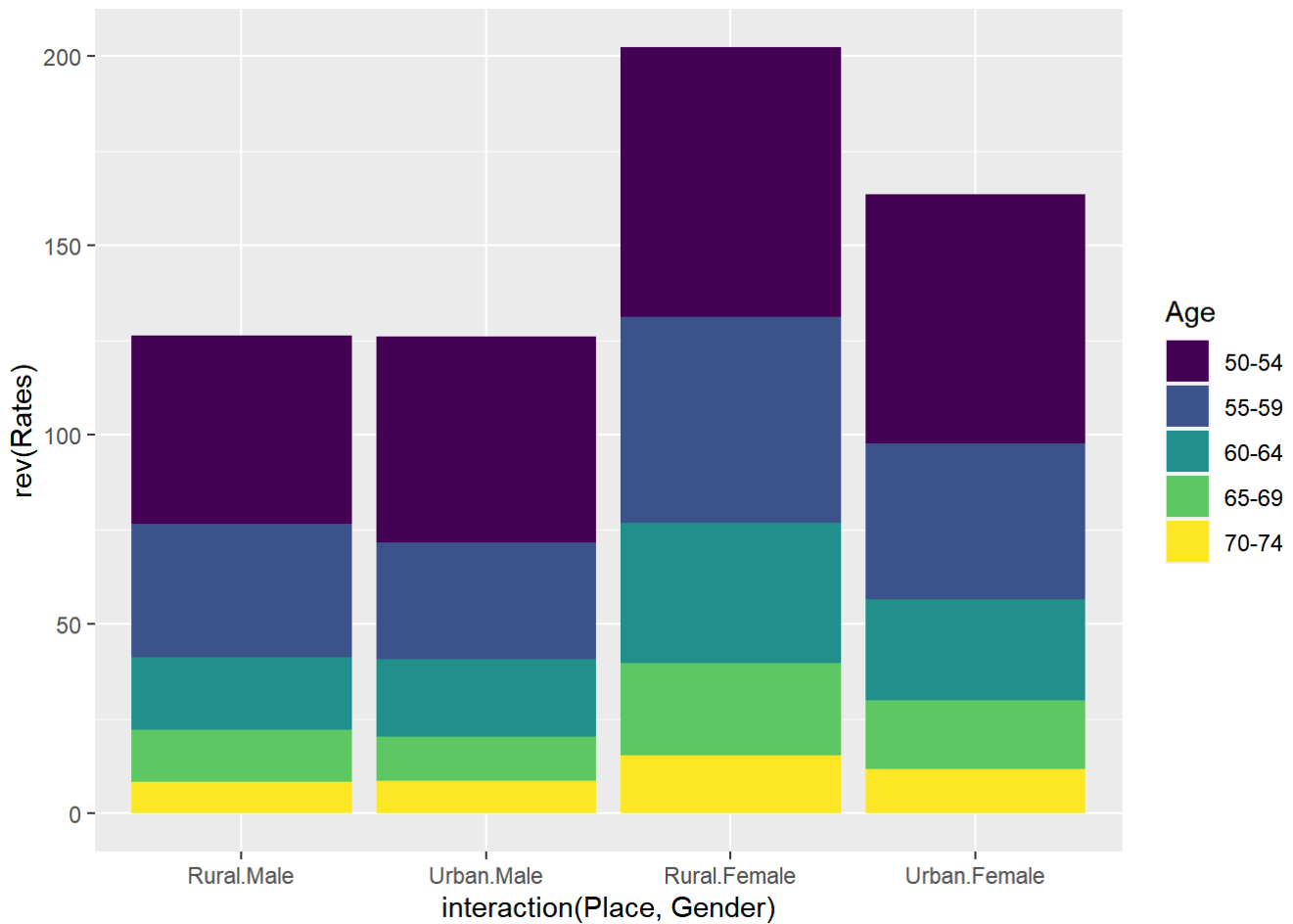
```
ggplot(data = VADeaths_df,
       mapping = aes(x = interaction(Place, Gender),
                     y = Rates,
                     fill = Age)) +
geom_bar(stat = "identity",
        position = "identity")
```



```
ggplot(data = VADeaths_df,  
       mapping = aes(x = interaction(Place, Gender),  
                     y = rev(Rates),  
                     fill = Age)) +  
geom_bar(stat = "identity",  
        position = "identity")
```



```
ggplot(data = VADeaths_df,
       mapping = aes(x = interaction(Place, Gender),
                     y = rev(Rates),
                     fill = Age)) +
geom_bar(stat = "identity",
        position = position_stack())
```



Comments

이번시간에는 VADeaths데이터를 통하여 깔끔하지않은 데이터를 만드는방법을 알 수있었습니다. 연령은 순서가 있기때문에 ordered를 이용하는것이 적절하다는 것을 알 수 있었습니다. %>%는 기존과 달리 구조화되어 있기 때문에 좀더 가독성이 높고 이해하기가 쉽게 설계되었음을 느낄수 있었습니다. 또한 ggplot을통해 데이터를 시각화하여 데이터를 더 직관적으로 이해할 수 있는 시간 이였습니다.