한국인의 삶을 파악하라(1) - 나이와 월급의 관계 ¶

- 몇살때 월급을 가장 많이 받을까?
- 대상 변수 : 나이(birth), 월급(income)

- 2006~2015년까지 전국에서 7000여 가구를 선정하여 매년 추적 조사한 자료
- 데이터 셋 : Koweps_hpc10_2015_beta1.sav
  - 2016년도 발간한 복지패널 데이터 6,914가구, 16,664명에 대한 정보

In [45]:

```
# install.packages("foreign")
```

In [46]:

```
library(foreign)
library(dplyr)
library(ggplot2)
library(readxl)
```

In [47]:

```
dat_welfare <- read.spss(file="D:WWdatasetWWR_DoitWWKoweps_hpc10_2015_beta1.sav", to.data.frame=T)
welfare <- dat_welfare
```

```
Warning message in read.spss(file = "D:WWdatasetWWR_DoitWWKoweps_hpc10_2015_beta1.sa
v", :
"D:WdatasetWR_DoitWKoweps_hpc10_2015_beta1.sav: Compression bias (0) is not the usua
l value of 100"
```

In [48]:

```
welfare <- rename(welfare,
                  sex=h10_g3,
                  birth=h10_g4,
                  marriage=h10_g10,
                  religion=h10_g11,
                  income=p1002_8aq1,
                  code_job=h10_eco9,
                  code_region=h10_reg7)
names(welfare)
```

'h10_id' 'h10_ind' 'h10_sn' 'h10_merkey' 'h_new' 'h10_cobf' 'h10_reg5'
'code_region' 'h10_din' 'h10_cin' 'h10_flag' 'p10_wgl' 'p10_wsl' 'p10_wgc' 'p10_wsc'
'h10_hc' 'nh1001_1' 'nh1001_2' 'h1001_1' 'h10_pind' 'h10_pid' 'h10_g1' 'h10_g2'
'sex' 'birth' 'h10_g6' 'h10_g7' 'h10_g8' 'h10_g9' 'marriage' 'religion' 'h10_g12'
'h1001_110' 'h1001_5aq1' 'h1001_5aq2' 'h1001_5aq3' 'h1001_5aq4' 'h10_med1'
'h10_med2' 'h10_med3' 'h10_med4' 'h10_med5' 'h10_med6' 'h10_med7' 'h10_med8'
'h10_g9_1' 'h10_med9' 'h10_med10' 'h10_eco1' 'h10_eco2' 'h10_eco3' 'h10_eco4'
'h10_eco4_1' 'h10_eco5_1' 'h10_eco6' 'h10_eco_7_1' 'h10_eco_7_2' 'h10_eco_7_3'
'h10_eco8' 'code_job' 'h10_eco10' 'h10_eco11' 'h10_soc1' 'h10_soc_2' 'h10_soc_3'
'h10_soc_4' 'h10_soc_5' 'h10_soc_6' 'h10_soc_7' 'h10_soc_8' 'h10_soc_9'
'h10_soc_10' 'h10_soc_11' 'h10_soc8' 'h10_soc9' 'h10_soc11' 'h10_soc10'
'h10_soc_12' 'h10_soc_13' 'h1005_1' 'h1005_3aq1' 'h1005_2' 'h1005_3' 'h1005_4'
'h1005_5' 'h1005_6' 'h1005_7' 'nh1005_8' 'nh1005_9' 'h1005_3aq2' 'h1006_aq1'
'h1006_1' 'h1006_2' 'h1006_4' 'h1006_5' 'h1006_3' 'h1006_6' 'h1006_8' 'h1006_9'
'h1006_aq2' 'h1006_aq3' 'h1006_10' 'h1006_11' 'h1006_12' 'h1006_13' 'h1006_14'
'h1006_21' 'h1006_22' 'h1006_23' 'h1006_24' 'h1006_25' 'h1006_27' 'h1006_30'
'h1006_33' 'h1006_36' 'h1006_39' 'h1006_3aq1' 'h1007_3aq1' 'h1007_3aq2'
'h1007_5aq1' 'h1007_3aq3' 'h1007_3aq4' 'h1007_3aq5' 'h1007_6aq1' 'h1007_3aq6'
'h1007_5aq2' 'h1007_3aq7' 'h1007_3aq8' 'h1007_3aq9' 'h1007_3aq10' 'h1007_3aq11'
'h1007_5aq3' 'h1007_5aq4' 'h1007_3aq13' 'h1007_6aq4' 'h1007_6aq6' 'h1007_3aq14'
'h1007_3aq15' 'h1007_3aq16' 'h1007_3aq17' 'h1007_4' 'h1007_6aq7' 'h1007_6aq8'
'h1007_6aq9' 'h1007_6aq10' 'h1007_6aq11' 'h1007_5' 'h1007_6aq12' 'h1007_6aq13'
'h1007_6aq14' 'h1007_9' 'h1009_9' 'h1009_6aq4' 'h10_inc1' 'h10_inc2_1'
'h10_inc2_2' 'h10_inc3_1' 'h10_inc3_2' 'h10_inc4_1' 'h10_inc4_2' 'h10_inc5_1'
'h10_inc5_2' 'h10_inc6_1' 'h10_inc6_2' 'h10_inc7_1' 'h10_inc7_2' 'h1008_106'
'h1008_107' 'h1008_108' 'h1008_109' 'h1008_110' 'h1008_111' 'h10_inc2_3'
'h10_inc2' 'h10_inc3_6' 'h10_inc3' 'h10_inc4_7' 'h10_inc4' 'h10_inc4_8' 'h10_inc4_9'
'h1008_155' 'h1008_156' 'h1008_157' 'h1008_158' 'h1008_160' 'h1008_159'
'h1008_3aq3' 'h1008_161' 'h1008_162' 'h1008_163' 'h1008_164' 'h1008_166'
'h1008_165' 'h1008_3aq4' 'h1008_167' 'h1008_168' 'h1008_169' 'h1008_170'
'h10_inc7_3' 'h10_inc7' 'h1008_aq9' 'h1008_aq10' 'h1008_aq11' 'h1008_aq12'
'h1008_aq13' 'h1008_aq14' 'h1008_aq15' 'h1008_6aq1' 'h1008_aq16' 'h1008_aq17'
'h1008_10aq1' 'h1008_aq19' 'h1008_aq20' 'h1008_aq21' 'h1008_5aq3' 'h1008_7aq1'
'h1008_aq22' 'h1008_7aq2' 'h1008_aq23' 'h1008_aq24' 'h1008_4aq116'
'h1008_4aq117' 'h1008_5aq1' 'h1008_7aq4' 'h1008_7aq5' 'h1008_7aq6' 'h1008_7aq7'
'h1008_7aq8' 'h1008_7aq9' 'h1008_aq25' 'h1008_7aq10' 'h1008_aq26' 'h1008_aq27'
'h1008_aq28' 'h1008_aq29' 'h1008_3aq5' 'h1008_4aq118' 'h1008_aq30' 'h1008_6aq3'
'h1008_3aq6' 'h1008_3aq7' 'nh1008_3aq1' 'h1008_aq32' 'h1008_aq33' 'h1008_aq34'
'h1008_3aq8' 'h1008_195' 'h1008_7aq11' 'h1009_aq1' 'h1009_aq2' 'h1009_aq3'
'h1009_aq4' 'h1009_aq5' 'h1009_aq6' 'h1009_aq7' 'h1009_aq8' 'h1010_aq1'
'h1010_aq2' 'h1010_aq3' 'h1010_aq4' 'h1010_aq5' 'h1010_aq6' 'h1010_aq7'
'h1010_aq8' 'h1010_aq9' 'h1010_aq10' 'h1010_aq11' 'h1010_aq12' 'h1010_aq13'

'h1010_aq14' 'h1010_aq15' 'h1010_aq16' 'h1010_aq17' 'h1010_aq18' 'h1010_aq19'
'h1010_aq20' 'h1010_26' 'h1010_27' 'h1010_aq23' 'h1010_aq24' 'h1010_aq25'
'h1010_aq26' 'h1011_2' 'h1011_3' 'h1011_4' 'h1011_5' 'h1011_6' 'h1011_7'
'h1011_8' 'h1011_3aq1' 'h1011_3aq2' 'h1011_3aq3' 'h1011_3aq4' 'h1011_3aq5'
'h1011_3aq6' 'h1011_3aq7' 'h1012_1' 'nh1012_1' 'h1012_2' 'h1012_3' 'h1012_4'
'h1012_5' 'h1012_6' 'h1012_7' 'nh1012_7' 'nh1012_8' 'h1012_9' 'h1012_10'
'h1012_11' 'h1012_12' 'h1012_13' 'h1012_14' 'h1012_15' 'h1012_3aq1' 'h1012_16'
'h1012_17' 'h1012_18' 'h1012_19' 'h1012_1_4aq1' 'h1012_1_5aq1' 'h1012_1_5aq2'
'h1012_1_5aq3' 'h1012_1_5aq4' 'h1012_1_5aq5' 'h1012_1_4aq2' 'h1012_1_4aq3'
'h1013_2' 'h1013_6' 'h1013_10' 'h1013_14' 'h1013_18' 'h1013_22' 'h1013_26'
'h1013_8aq1' 'h1013_5aq1' 'h1013_8aq2' 'h1013_4aq1' 'h1013_4aq2' 'h1013_4aq4'
'h1013_4aq6' 'h1013_4aq8' 'h1013_4aq10' 'h1013_5aq4' 'h1013_5aq6' 'h1013_5aq8'
'h1013_6aq1' 'h1013_4aq14' 'h1013_4aq15' 'h1013_4aq16' 'h1013_4aq17'
'h1013_4aq18' 'h1013_4aq20' 'h1013_4aq22' 'h1013_4aq24' 'h1013_4aq26'
'h1013_4aq28' 'h1013_4aq30' 'h1013_4aq32' 'h1014_4' 'h1014_8' 'h1014_12'
'h1014_16' 'h1014_20' 'h1014_24' 'h1014_28' 'h1014_32' 'h1014_36' 'h1014_3aq1'
'h1014_4aq1' 'h1015_4' 'h1015_8' 'h1015_12' 'h1015_20' 'h1015_25' 'h1015_29'
'h1015_33' 'h1015_37' 'h1015_4aq1' 'h1015_7aq1' 'h1015_aq1' 'h1015_40'
'h1015_41' 'h1015_42' 'h1015_43' 'h1015_44' 'h1015_45' 'h1015_46' 'h1015_47'
'h1015_48' 'h1015_49' 'h1015_50' 'h1015_51' 'h1015_52' 'h1015_53' 'h1015_54'
'h1015_55' 'h1015_56' 'h1015_57' 'h1015_60' 'h1015_aq2' 'h1015_61' 'h1015_62'
'h1015_63' 'h1015_66' 'h1015_67' 'h1015_68' 'h1015_aq3' 'h1015_69' 'h1015_70'
'h1015_71' 'h1015_74' 'h1015_75' 'h1015_76' 'h1015_aq4' 'h1015_77' 'h1015_78'
'h1015_79' 'h1015_82' 'h1015_83' 'h1015_84' 'h1015_aq5' 'h1015_85' 'h1015_86'
'h1015_87' 'h1015_90' 'h1015_91' 'h1015_92' 'h1015_aq6' 'h1015_93' 'h1015_94'
'h1015_95' 'h1015_98' 'h1015_99' 'h1015_100' 'h1015_aq7' 'h1015_101' 'h1015_102'
'h1015_103' 'h1015_106' 'h1015_107' 'h1016_6aq1' 'h1016_6aq4' 'h1016_8'
'h1016_24' 'h1016_4aq1' 'h1016_4aq4' 'h1016_32' 'h1016_36' 'h1016_40'
'h1016_4aq7' 'h1016_56' 'h1016_60' 'h1016_64' 'h1017_1' 'h1017_2' 'h1017_3'
'h1017_4' 'h1017_5' 'h1017_6' 'h1017_7' 'p10_fnum' 'p10_tq' 'p10_cp' 'p1001_1'
'p1001_2' 'p1001_aq1' 'p1001_3' 'p1001_4' 'p1001_5' 'p1001_6' 'p1001_10'
'p1001_11' 'p1001_12' 'p1001_13' 'p1001_14' 'p1001_7' 'p1001_8' 'p1001_9'
'p1001_15' 'p1001_16' 'p1001_17' 'p1001_18' 'p1001_19' 'p1001_20' 'p1001_21'
'p1001_22' 'p1001_23' 'p1001_24' 'p1001_25' 'p1001_27' 'p1001_28' 'p1001_29'
'p1001_30' 'p1001_31' 'p1001_32' 'p1001_33' 'p1001_34' 'p1002_1' 'p1002_2'
'p1002_3' 'p1002_4' 'p1002_5' 'p1002_6' 'p1002_7' 'p1002_8' 'income' 'p1002_9'
'p1002_8aq2' 'p1002_4aq1' 'p1002_10' 'p1002_11' 'p1002_12' 'p1002_8aq3'
'p1002_8aq4' 'p1002_31' 'p1002_4aq2' 'p1002_4aq3' 'p1002_4aq4' 'p1002_8aq5'
'p1002_8aq6' 'p1002_8aq7' 'p1002_8aq8' 'p1002_8aq9' 'p1002_8aq10' 'p1002_8aq11'
'p1002_8aq12' 'p1002_8aq13' 'p1002_8aq14' 'p1002_8aq15' 'p1002_aq2' 'p1002_aq3'
'p1002_aq4' 'p1002_aq5' 'p1003_1' 'p1003_2' 'p1003_5' 'p1003_6' 'p1003_7'
'p1003_8' 'p1003_9' 'p1003_10' 'p1003_11' 'p1003_12' 'p1004_1' 'p1004_2'
'p1004_3' 'p1004_4' 'p1004_5' 'p1004_6' 'p1004_7' 'p1004_8' 'p1004_9' 'p1004_10'
'p1004_11' 'p1004_12' 'p1004_13' 'p1004_3aq1' 'p1004_3aq2' 'p1004_3aq3'
'p1004_3aq4' 'p1004_3aq5' 'p1004_3aq6' 'p1004_3aq7' 'p1004_3aq8' 'p1005_3aq1'
'p1005_3aq2' 'p1005_3aq3' 'p1005_3aq4' 'p1005_3aq5' 'p1005_3aq6' 'p1005_3aq7'
'p1005_3aq8' 'p1005_3aq9' 'p1005_3aq10' 'p1005_2' 'p1005_3' 'p1005_4aq1'
'p1005_4aq2' 'p1005_4aq3' 'p1005_4aq4' 'p1005_4aq5' 'p1005_4aq6' 'p1005_4aq7'
'p1005_4aq8' 'p1005_5' 'p1005_6' 'p1005_7' 'p1005_8' 'p1005_3aq11' 'p1005_9'
'p1005_10' 'p1005_11' 'p1005_12' 'p1005_13' 'p1005_14' 'p1005_15' 'p1005_16'
'p1005_17' 'p1005_18' 'p1005_19' 'p1005_20' 'p1005_21' 'p1005_22' 'p1005_23'
'p1005_24' 'p1005_25' 'p1005_26' 'p1005_27' 'p1005_28' 'p1005_29' 'p1005_4aq9'

'p1005_4aq10' 'p1005_4aq11' 'p1005_aq1' 'p1005_aq2' 'p1005_aq3' 'p1005_aq4'
'p1005_6aq1' 'p1005_6aq2' 'p1005_6aq3' 'p1005_6aq4' 'p1005_6aq5' 'p1005_6aq6'
'p1005_6aq7' 'p1005_6aq8' 'p1005_6aq9' 'p1005_7aq1' 'p1005_7aq2' 'p1005_7aq3'
'p1007_3aq1' 'p1007_3aq2' 'p1007_3aq3' 'p1007_3aq4' 'p1007_3aq5' 'p1007_3aq6'
'p1007_3aq7' 'np1006_1' 'np1006_2' 'np1006_3' 'np1006_4' 'np1006_5' 'np1006_6'
'np1006_7' 'np1006_8' 'np1006_9' 'np1006_10' 'np1006_11' 'np1006_12' 'np1006_13'
'np1006_14' 'np1006_15' 'np1006_16' 'np1006_17' 'np1006_18' 'np1006_19'
'np1006_20' 'np1006_21' 'np1006_22' 'np1006_23' 'np1006_24' 'np1006_25'
'np1006_26' 'np1006_27' 'np1006_28' 'np1006_29' 'np1006_30' 'np1006_31'
'np1006_32' 'np1006_33' 'np1006_34' 'np1006_35' 'np1006_36' 'np1006_37'
'np1006_38' 'np1006_39' 'np1006_40' 'np1006_41' 'np1006_42' 'np1006_43'
'np1006_44' 'p1006_3aq1' 'c10_fnum' 'c10_cp' 'c10_grade' 'c1001_1' 'c1001_2'
'c1001_3' 'c1001_4' 'c1001_5' 'c1001_6' 'c1001_7' 'c1001_8' 'c1001_9' 'c1001_10'
'c1001_11' 'c1001_12' 'c1001_13' 'c1001_7aq5' 'c1001_7aq6' 'c1001_7aq7'
'c1001_7aq8' 'c1001_4aq1' 'c1001_4aq2' 'c1001_4aq3' 'c1001_4aq4' 'c1001_4aq5'
'c1001_4aq6' 'c1002_1' 'c1002_2' 'c1002_3' 'c1002_4' 'c1002_5' 'c1002_6'
'c1002_7' 'c1002_8' 'c1002_9' 'c1002_10' 'c1002_11' 'c1002_12' 'c1002_13'
'c1002_14' 'c1002_15' 'c1002_16' 'c1002_17' 'c1002_18' 'c1002_19' 'c1002_20'
'c1002_21' 'c1002_22' 'c1002_23' 'c1002_24' 'c1002_25' 'c1002_26' 'c1002_4aq1'
'c1002_27' 'c1002_28' 'c1002_29' 'c1002_30' 'c1002_31' 'c1002_32' 'c1002_33'
'c1002_34' 'c1002_35' 'c1002_36' 'c1002_37' 'c1002_38' 'c1002_39' 'c1002_40'
'c1002_41' 'c1002_42' 'c1002_43' 'c1002_44' 'c1002_45' 'c1002_46' 'c1002_47'
'c1002_48' 'c1002_49' 'c1002_50' 'c1002_51' 'c1002_52' 'c1002_53' 'c1002_54'
'c1002_55' 'c1002_56' 'c1002_57' 'c1002_58' 'c1002_4aq2' 'c1002_59' 'c1002_60'
'c1002_61' 'c1002_62' 'c1002_63' 'c1002_64' 'c1002_65' 'c1002_66' 'c1002_67'
'c1002_68' 'c1002_69' 'c1002_70' 'c1002_71' 'c1002_72' 'c1002_73' 'c1002_74'
'c1002_75' 'c1002_76' 'c1002_77' 'c1002_7aq1' 'c1002_7aq2' 'c1002_7aq3'
'c1002_7aq4' 'c1002_7aq5' 'c1002_7aq6' 'c1002_7aq7' 'c1002_7aq8' 'c1002_7aq9'
'c1002_7aq10' 'c1002_7aq11' 'c1002_7aq12' 'c1002_7aq13' 'c1002_7aq14' 'c1002_78'
'c1002_79' 'c1002_80' 'c1002_81' 'c1002_82' 'c1002_83' 'c1002_84' 'c1002_85'
'c1002_86' 'c1002_87' 'c1002_88' 'c1002_89' 'c1002_90' 'c1002_91' 'c1002_92'
'c1002_93' 'c1002_94' 'c1002_95' 'c1002_96' 'c1002_97' 'c1002_98' 'c1002_4aq3'
'c1002_4aq4' 'c1002_4aq5' 'c1002_4aq6' 'c1002_4aq7' 'c1002_4aq8' 'c1002_7aq15'
'c1003_4aq1' 'c1003_4aq2' 'c1003_4aq3' 'c1003_4aq4' 'c1003_4aq5' 'c1003_4aq6'
'c1003_4aq7' 'c1003_4aq8' 'c1003_4aq9' 'c1003_6' 'c1003_7' 'c1003_8' 'c1003_9'
'c1003_13' 'c1003_14' 'c1003_15' 'c1004_4aq1' 'c1004_4aq2' 'c1004_4aq3'
'c1004_4aq4' 'c1004_4aq5' 'c1004_4aq6' 'c1004_1' 'c1004_2' 'c1004_3' 'c1004_4'
'c1004_5' 'c1004_6' 'c1004_7' 'c1004_8' 'c1004_9' 'c1004_10' 'c1004_4aq7'
'c1004_4aq8' 'c1005_12' 'c1005_13' 'c1005_14' 'c1005_15' 'c1005_16' 'c1005_17'
'c1005_4aq1' 'c1005_18' 'c1005_19' 'c1005_7aq1' 'c1005_7aq2' 'c1005_7aq3'
'c1005_7aq4' 'c1005_7aq5' 'c1005_7aq6' 'c1005_7aq7' 'c1005_7aq8' 'c1005_7aq9'
'c1005_7aq10' 'c1005_7aq11' 'c1005_7aq12' 'c1005_7aq13' 'c1005_7aq14'
'c1005_7aq15' 'c1005_7aq16' 'c1005_7aq17' 'c1005_7aq18' 'c1005_7aq19'
'c1005_7aq20' 'c1005_20' 'c1005_21' 'c1005_22' 'c1005_23' 'c1005_24' 'c1005_25'
'c1005_26' 'c1005_27' 'c1005_28' 'c1005_29' 'c1005_30' 'c1005_31' 'c1005_32'
'c1005_36' 'c1005_4aq2' 'c1005_38' 'c1005_4aq3' 'c1005_40' 'c1005_10aq1'
'c1005_42' 'c1005_4aq4' 'c1005_44' 'c1005_4aq5' 'c1005_46' 'c1005_4aq6'
'c1005_48' 'c1005_4aq7' 'c1005_4aq8' 'c1005_4aq9' 'c1005_4aq10' 'c1005_4aq11'
'c1005_4aq12' 'c1005_4aq13' 'c1005_4aq14' 'c1005_4aq15' 'c1007_4aq1'
'c1007_4aq2' 'c1007_7aq1' 'c1007_7aq2' 'c1007_7aq3' 'c1007_4aq3' 'c1007_4aq4'
'c1007_4aq5' 'c1007_4aq6' 'c1007_4aq7' 'c1007_4aq8' 'c1007_4aq9' 'c1007_4aq10'
'c1007_4aq11' 'c1007_4aq12' 'c1007_4aq13' 'c1007_4aq14' 'c1007_4aq15'

'c1007_4aq16' 'c1007_4aq17' 'c1007_4aq18' 'c1007_4aq19' 'c1007_4aq20'
'c1007_4aq21' 'c1007_4aq22' 'c1007_4aq23' 'h10_pers_income1' 'h10_pers_income2'
'h10_pers_income3' 'h10_pers_income4' 'h10_pers_income5'

In [49]:

```
welfare <- select(welfare, sex, birth, marriage, religion, income, code_job, code_region)
names(welfare)
```

'sex' 'birth' 'marriage' 'religion' 'income' 'code_job' 'code_region'

01 변수 확인

In [50]:

```
class(welfare$birth)
class(welfare$income)

dim(welfare)
head(welfare)
```

'numeric'

'numeric'

16664 7

sex	birth	marriage	religion	income	code_job	code_region
2	1936	2	2	NA	NA	1
2	1945	2	2	NA	NA	1
1	1948	2	2	120	942	1
1	1942	3	1	200	762	1
2	1923	2	1	NA	NA	1
1	1962	1	1	NA	530	1

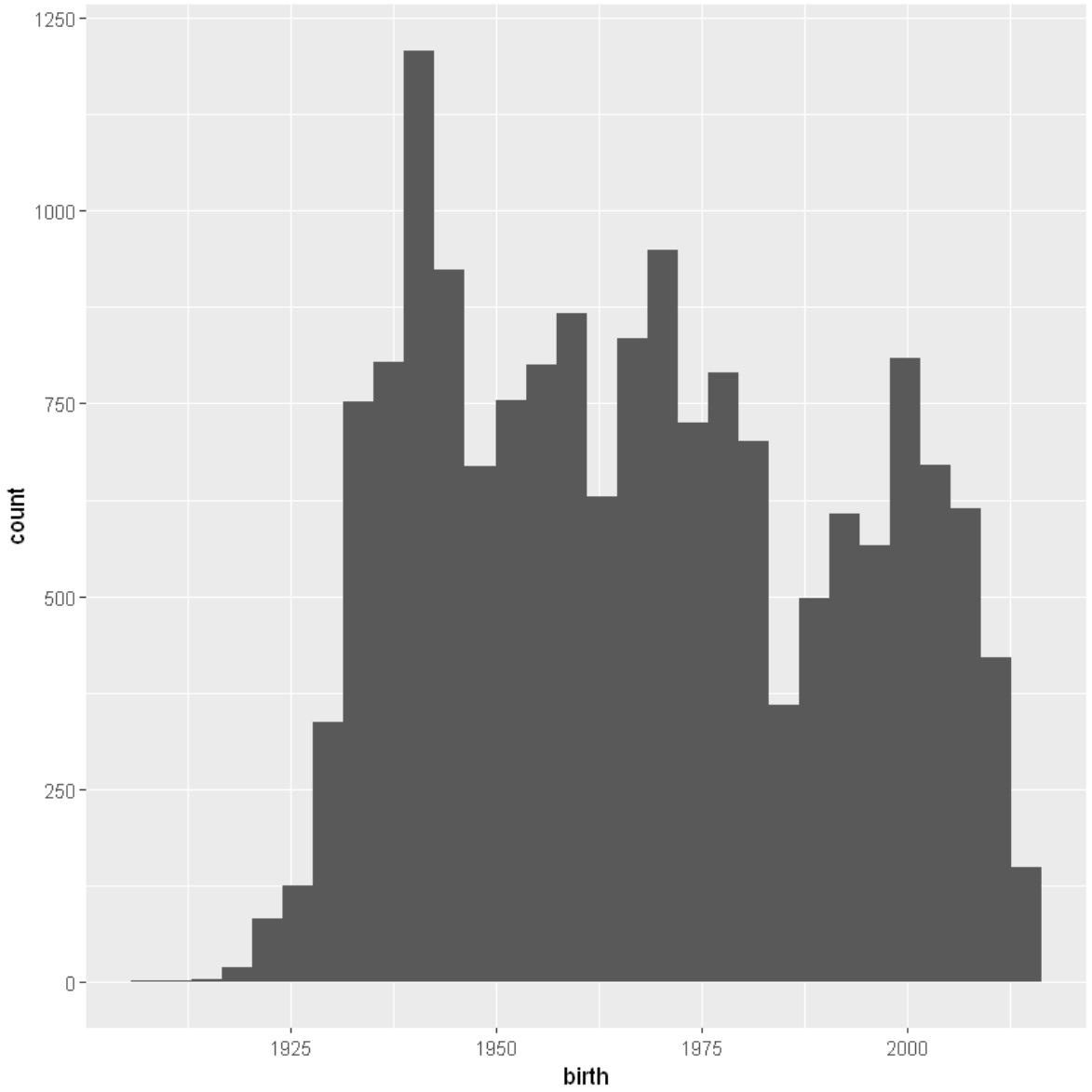
birth 살펴보기

In [51]:

```
summary(welfare$birth)
ggplot(welfare, aes(x=birth)) + geom_histogram()
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1907    1946    1966    1968    1988    2014
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



02 결측치 확인 및 파생변수 생성

- 1900~2014사이의 값을 지니고, 모름/무응답은 9999로 코딩되어 있음.

In [52]:

```
table(is.na(welfare$birth))
```

```
FALSE
16664
```

In [53]:

```
## 만약 결측치가 있다면 다음과 같이 처리 가능
welfare$birth <- ifelse(welfare$birth==9999, NA, welfare$birth)
table(is.na(welfare$birth))
```

```
FALSE
16664
```

나이변수(age) 만들기

- 2015년도 조사 진행. 2015에서 연도를 뺀 후, 1년을 더하면 된다.
- age <- 2015 - welfare$birth + 1

In [54]:

```
welfare$age <- 2015 - welfare$birth + 1
welfare
```

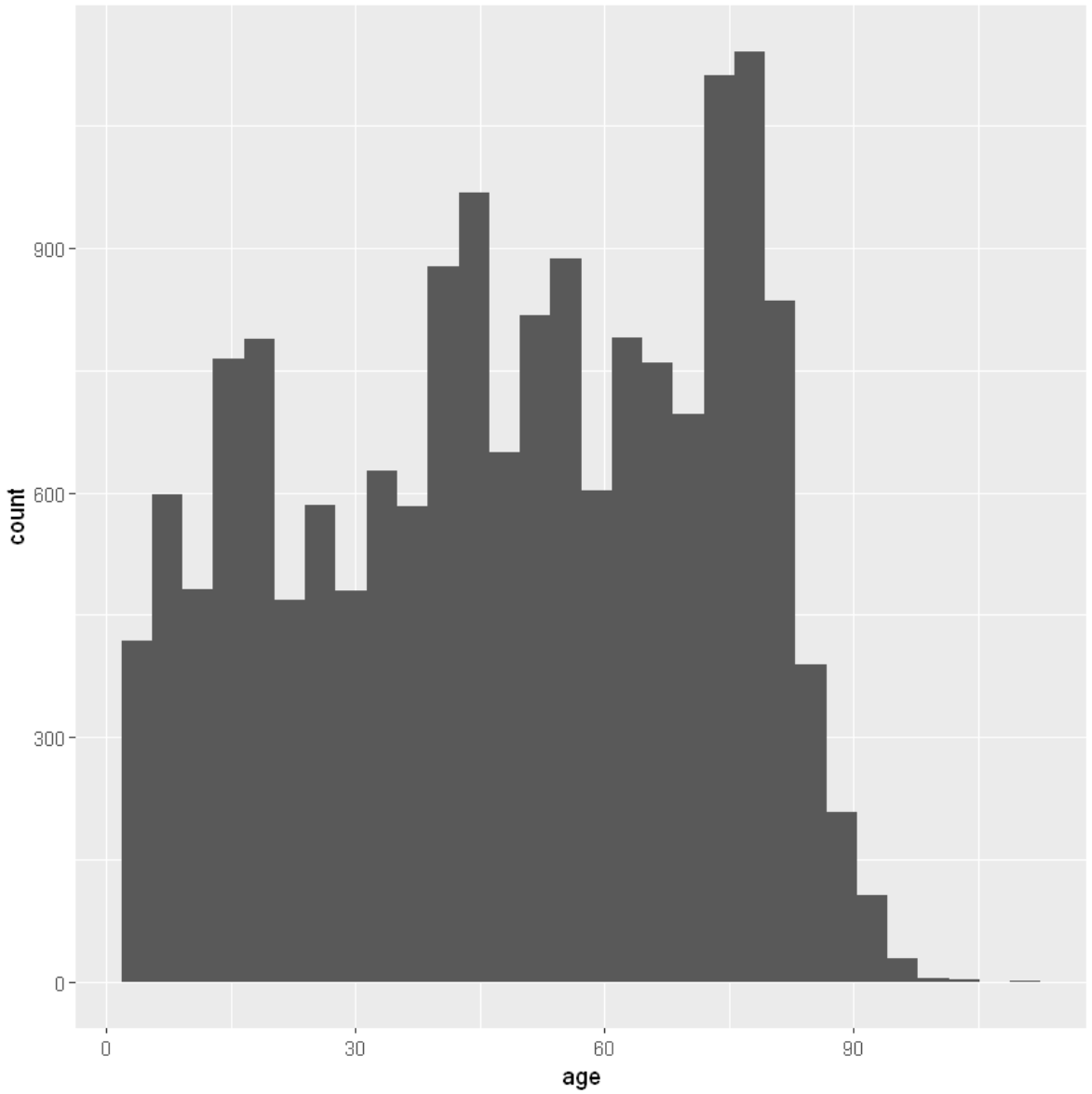
sex	birth	marriage	religion	income	code_job	code_region	age
2	1936	2	2	NA	NA	1	80
2	1945	2	2	NA	NA	1	71
1	1948	2	2	120	942	1	68
1	1942	3	1	200	762	1	74
2	1923	2	1	NA	NA	1	93
1	1962	1	1	NA	530	1	54
2	1963	1	1	NA	NA	1	53
2	2003	0	1	NA	NA	1	13
1	1927	1	1	NA	NA	1	89
2	1934	1	1	NA	NA	1	82
2	1930	2	2	NA	NA	7	86
1	1956	1	2	NA	NA	7	60
2	1960	1	2	NA	NA	7	56
2	1940	2	1	NA	NA	2	76
2	1970	3	1	NA	NA	1	46
2	1992	5	1	NA	NA	1	24
1	1940	1	2	NA	NA	1	76
2	1941	1	2	NA	NA	1	75
1	1962	5	2	40	999	1	54
1	1978	1	2	350	312	1	38
2	1975	1	2	200	254	1	41
2	2004	0	2	NA	NA	1	12
2	2008	0	2	NA	NA	1	8
1	1941	1	1	NA	NA	1	75
2	1942	1	1	150	510	1	74
1	1964	5	2	NA	530	1	52
2	1940	2	2	NA	NA	1	76
1	1975	5	1	NA	286	1	41
1	1978	1	2	NA	521	1	38
2	1961	1	2	NA	NA	1	55
...	...	...	...	...	...	...	...
2	1934	1	2	NA	NA	6	82
1	1942	1	1	75.00	873	6	74
2	1945	1	1	NA	NA	6	71
1	1945	1	2	NA	NA	6	71
2	1948	1	2	NA	NA	6	68

sex	birth	marriage	religion	income	code_job	code_region	age
1	1956	3	1	162.00	873	6	60
1	1941	1	2	NA	611	6	75
2	1939	1	2	NA	611	6	77
1	1966	1	1	188.00	873	6	50
2	1962	1	1	90.00	899	6	54
2	1993	5	1	NA	NA	6	23
1	2000	0	1	NA	NA	6	16
2	2003	0	1	NA	NA	6	13
1	1944	1	2	NA	611	6	72
2	1947	1	2	NA	611	6	69
1	1933	1	2	NA	NA	6	83
2	1947	1	1	NA	NA	6	69
1	1933	1	2	NA	611	6	83
2	1934	1	2	NA	611	6	82
2	1934	2	1	NA	NA	6	82
1	1956	1	2	187.66	743	6	60
2	1958	1	1	NA	611	6	58
1	1932	1	2	NA	NA	6	84
2	1933	1	2	NA	NA	6	83
1	1967	1	1	NA	874	5	49
2	1967	1	1	NA	NA	5	49
2	1992	5	1	302.50	314	5	24
1	1995	5	1	NA	NA	5	21
2	1998	0	1	NA	NA	5	18
1	2001	0	1	NA	NA	5	15

In [55]:

```
ggplot(welfare, aes(x=age)) + geom_histogram()
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



03 나이(age)와 월급(income)의 관계 분석

(가) 나이에 따른 월급 평균표만들기

In [56]:

```
age_income <- welfare %>%
              filter(!is.na(income)) %>%
              group_by(age) %>%
              summarise(mean_income = mean(income))
head(age_income)
```

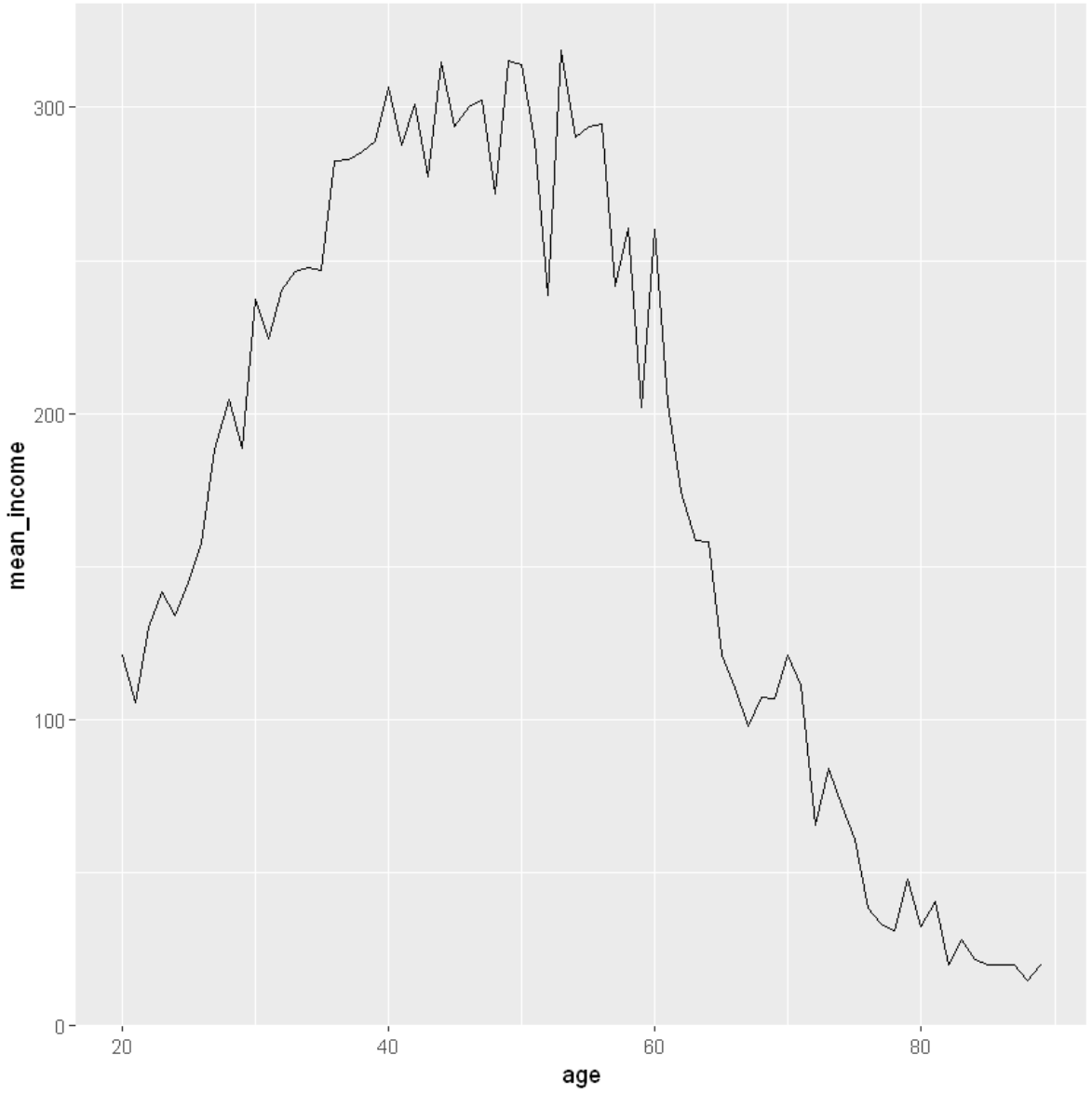
age	mean_income
20	121.3000
21	105.5185
22	130.0923
23	141.7157
24	134.0877
25	144.6559

(나) ggplot를 이용한 그래프 그리기

- x축 : age, y축 : 월급평균(mean_income)

In [57]:

```
ggplot(age_income, aes(x=age, y=mean_income)) + geom_line()
```



한국인의 삶을 파악하라(2) - 나이와 월급의 관계

- 어떤 연령의 월급이 가장 많이 받을까?
- 대상 변수 : 나이(birth)=> 연령대(class), 월급(mean_income)

In [58]:

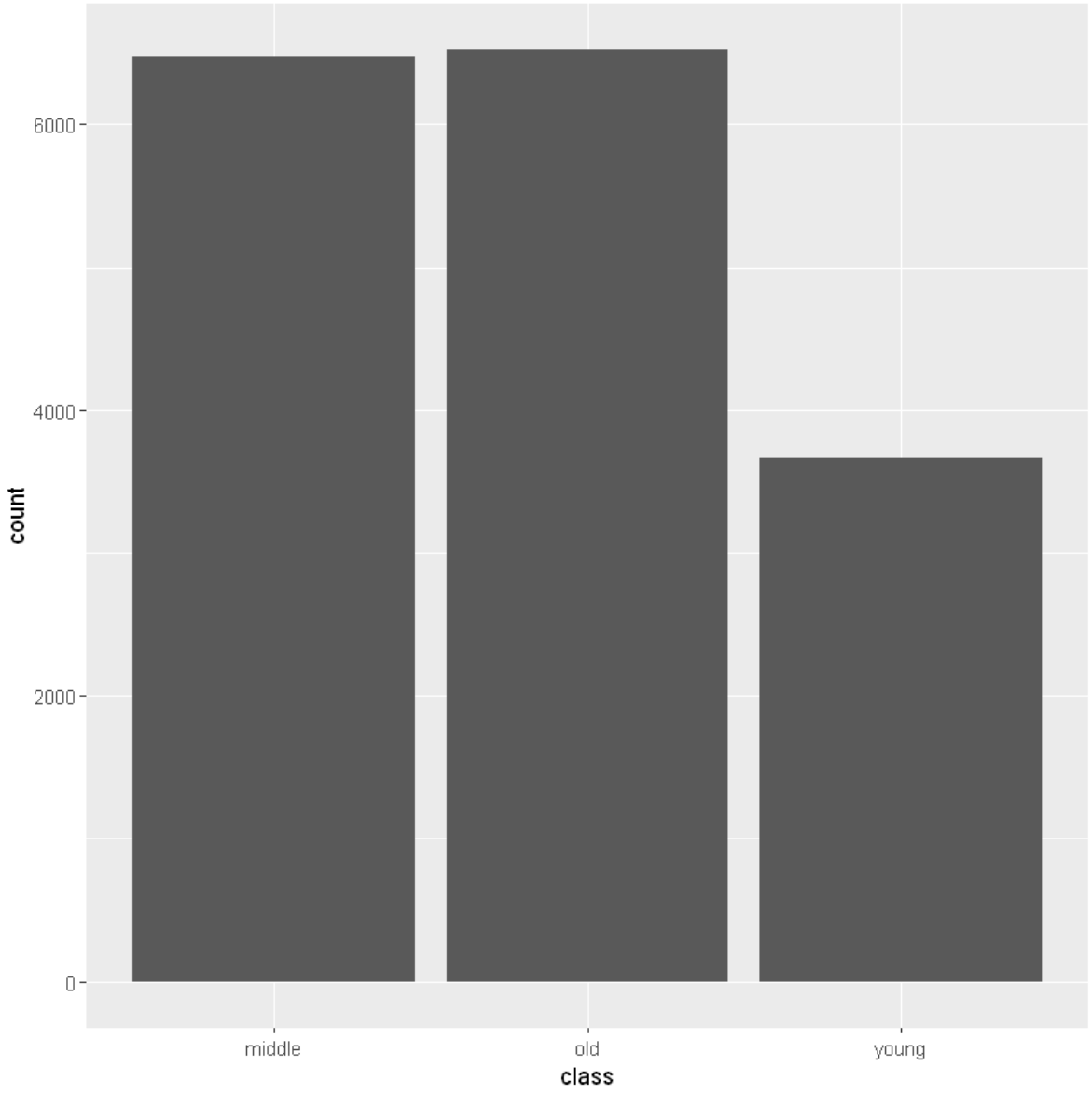
```
welfare <- welfare %>%
           mutate(class=ifelse(age<25, 'young',
                               ifelse(age >= 59, "middle", "old")))
table(welfare$class)
```

```
middle    old  young
  6475   6522   3667
```

막대 그래프

In [59]:

```
ggplot(welfare, aes(x=class)) + geom_bar()
```



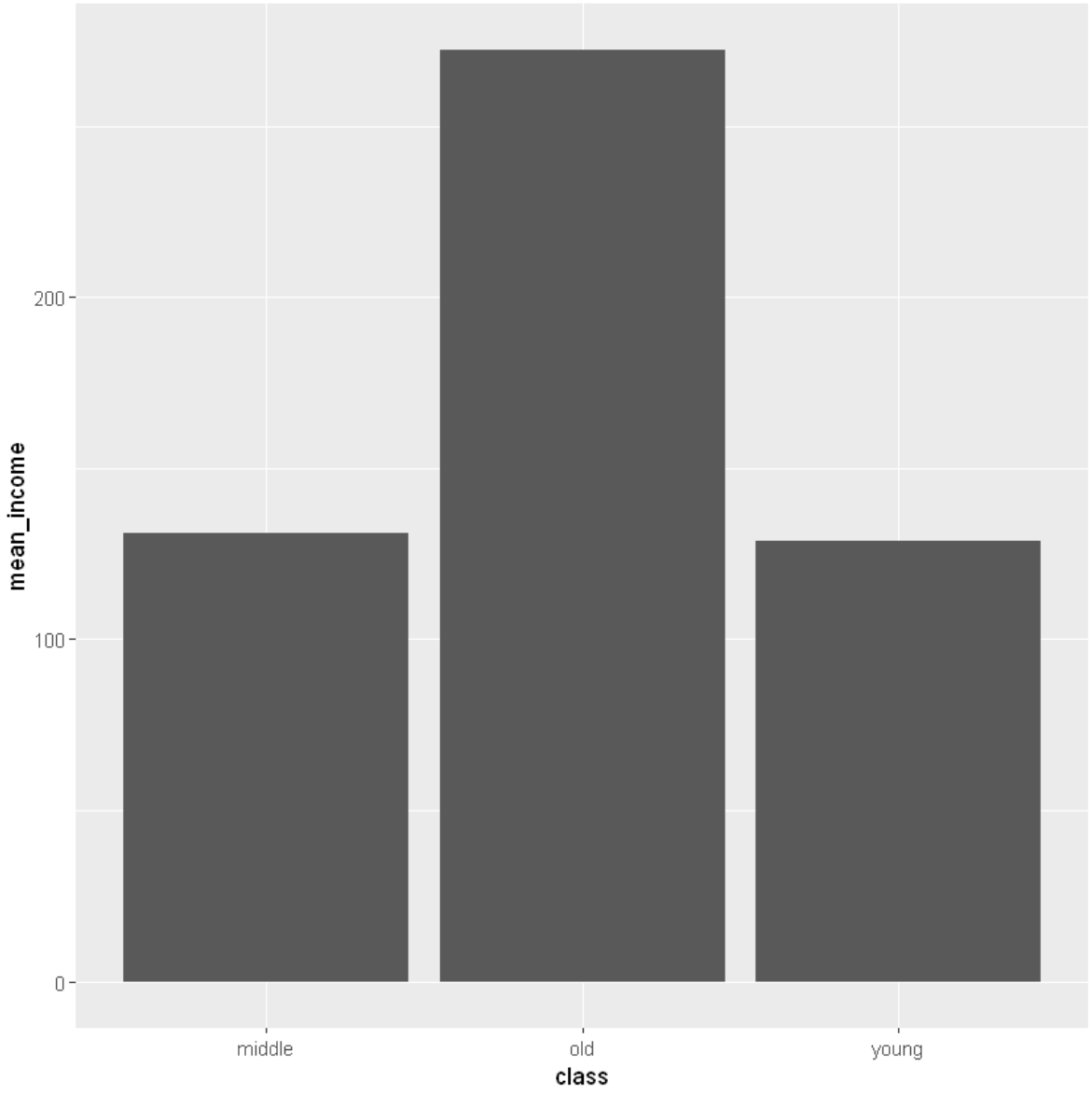
In [60]:

```
### 연령대별 월급 평균표 만들기
class_income <- welfare %>% filter(!is.na(income)) %>%
                           group_by(class) %>%
                           summarise(mean_income = mean(income))
class_income
```

class	mean_income
middle	131.0803
old	272.2487
young	128.5538

In [61]:

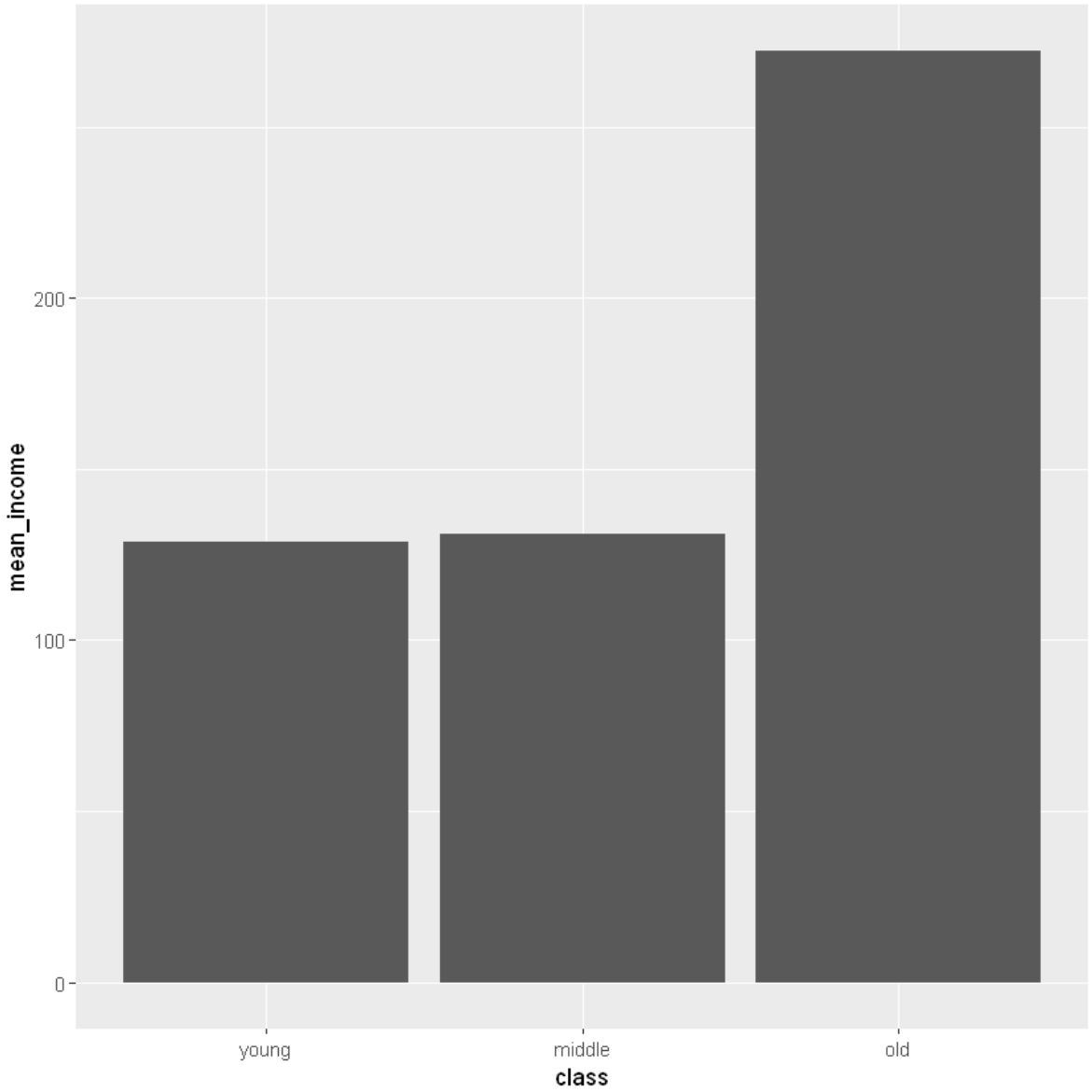
```
ggplot(class_income, aes(x=class, y=mean_income)) + geom_col()
```



x축의 값을 초년, 중년, 노년의 나이 순으로 정렬하도록 설정

In [62]:

```
ggplot(class_income, aes(x=class, y=mean_income)) + geom_col() +
            scale_x_discrete(limits = c('young', 'middle', 'old'))
```



확인

- 중년이 280만원 정도로 가장 높다.
- 노년은(131만) 초년이 받는 것보다 적은 월급(128만원)

(실습과제) P2-1

- (가) 10대, 20대, ...,60대, 기타 연령대로 나누고 가장 수입을 많이 받는 연령대는?
- (나) 10대, 20대, ...,60대, 기타 연령대로 나누고 가장 수입을 적게 받는 연령대는?

한국인의 삶을 파악하라(3) - 연령대 및 성별 월급 차이

- 성별 월급 차이는 연령대별로 다를까?
- 대상 변수 : 나이(birth)=> 연령대(class), 월급(mean_income), 성별(sex)

In [63]:

```
names(welfare)
```

'sex' 'birth' 'marriage' 'religion' 'income' 'code_job' 'code_region' 'age' 'class'

In [64]:

```
summary(welfare)
head(welfare$sex)
table(welfare$sex)
```

```
      sex            birth         marriage        religion
 Min.   :1.000   Min.   :1907   Min.   :0.000   Min.   :1.000
 1st Qu.:1.000   1st Qu.:1946   1st Qu.:1.000   1st Qu.:1.000
 Median :2.000   Median :1966   Median :1.000   Median :2.000
 Mean   :1.545   Mean   :1968   Mean   :1.648   Mean   :1.517
 3rd Qu.:2.000   3rd Qu.:1988   3rd Qu.:2.000   3rd Qu.:2.000
 Max.   :2.000   Max.   :2014   Max.   :6.000   Max.   :2.000

     income          code_job       code_region         age
 Min.   :   0.0   Min.   : 111.0   Min.   :1.000   Min.   :  2.00
 1st Qu.: 122.0   1st Qu.: 314.0   1st Qu.:2.000   1st Qu.: 28.00
 Median : 192.5   Median : 611.0   Median :3.000   Median : 50.00
 Mean   : 241.6   Mean   : 591.2   Mean   :3.705   Mean   : 48.43
 3rd Qu.: 316.6   3rd Qu.: 863.0   3rd Qu.:6.000   3rd Qu.: 70.00
 Max.   :2400.0   Max.   :1012.0   Max.   :7.000   Max.   :109.00
 NA's   :12030    NA's   :9135
    class
 Length:16664
 Class :character
 Mode  :character
```

2 2 1 1 2 1

```
   1    2
7578 9086
```

연령대 및 성별 월급 평균표 만들기

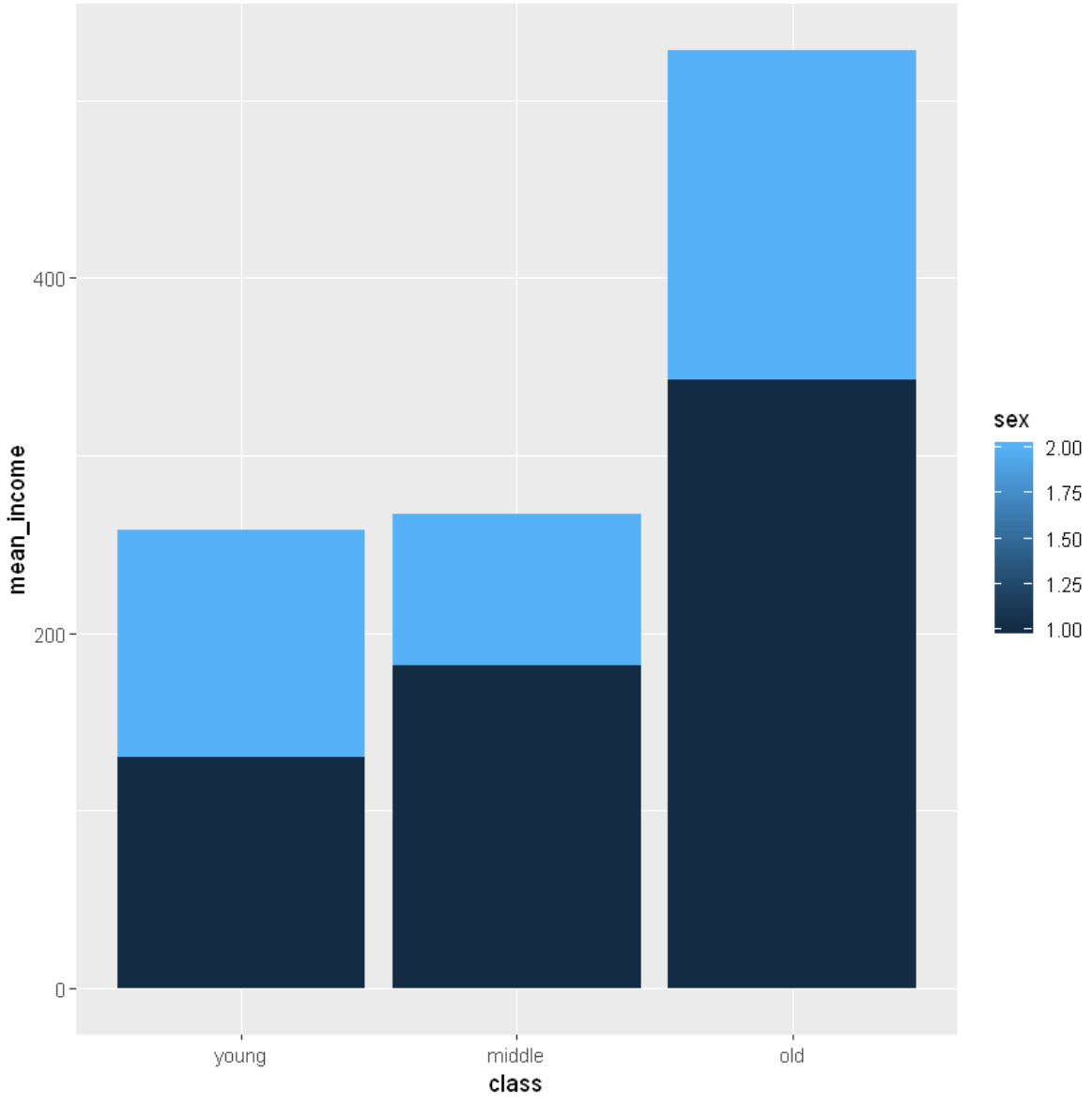
In [65]:

```
sex_income <- welfare %>% filter(!is.na(income)) %>%
                group_by(class, sex) %>%
                summarise(mean_income = mean(income))
sex_income
```

class	sex	mean_income
middle	1	181.82479
middle	2	85.24303
old	1	342.73009
old	2	185.72525
young	1	130.24825
young	2	127.85505

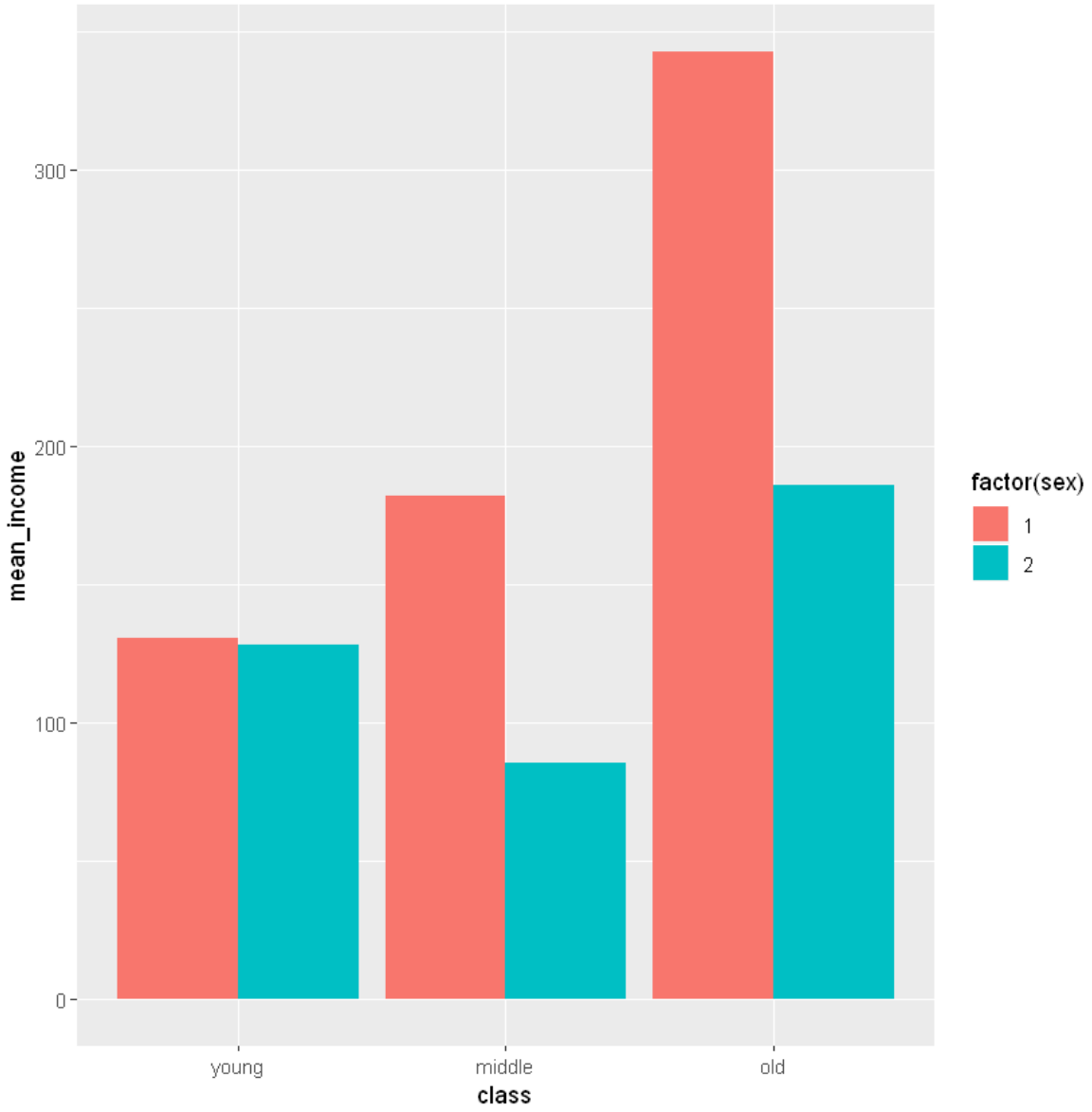
In [66]:

```
ggplot(data=sex_income, aes(x=class, y=mean_income, fill=sex)) +   # fill=성별에 따른 색깔 표시
             geom_col() +                                          # 그래프 종류
             scale_x_discrete(limits=c("young", "middle", "old"))  # 축 순서 설정
```



In [73]:

```
ggplot(data=sex_income, aes(x=class, y=mean_income, fill=factor(sex))) +   # fill=성별에 따른 색깔
            geom_col(position="dodge") +                                   # 그래프 종류
            scale_x_discrete(limits=c("young", "middle", "old"))   # 축 순서 설정
```



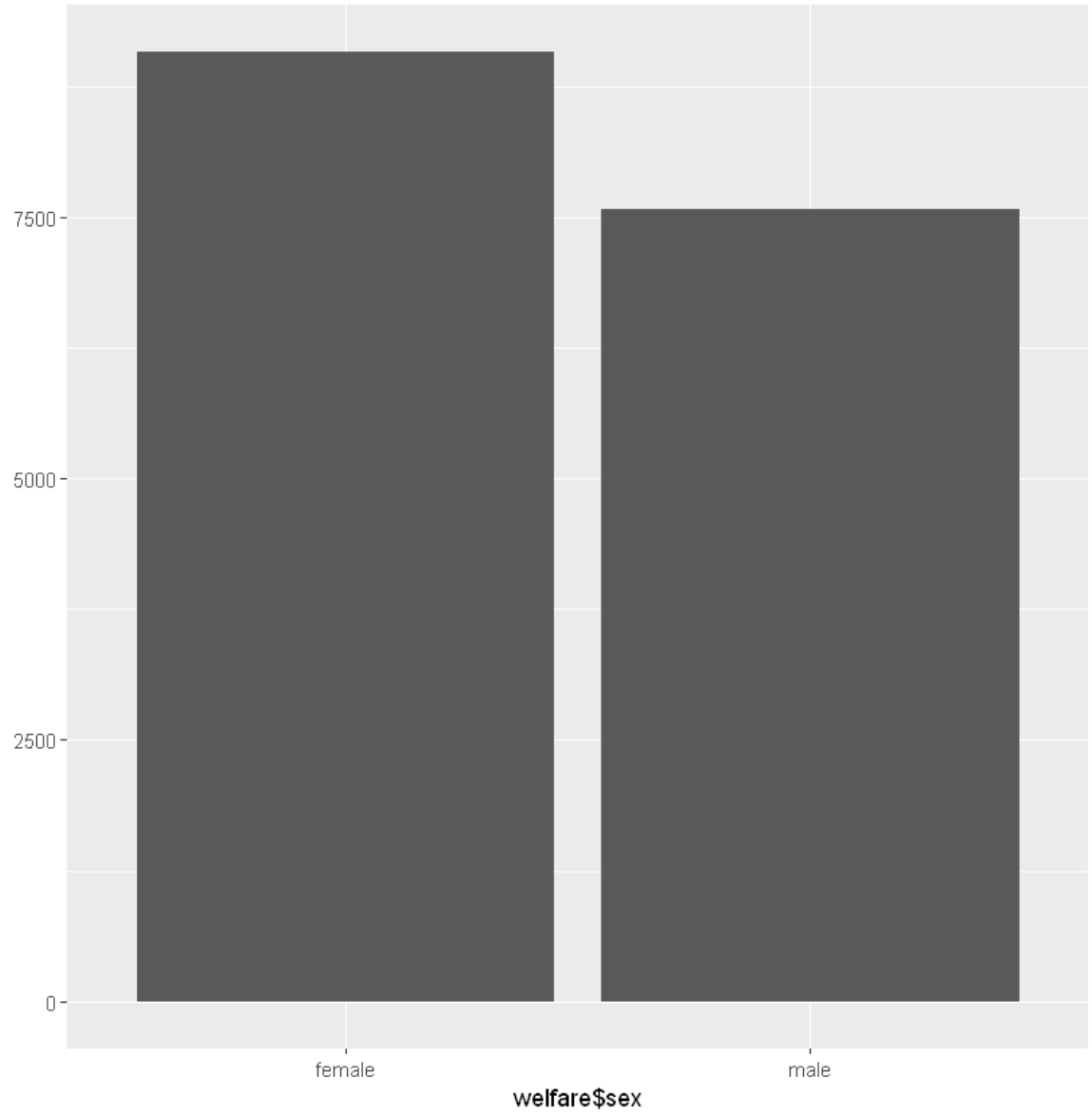
(실습과제) p2-2

- (가) 성별(1,2)를 'male', 'female'로 변경해 보자.
- (나) 연령대별 성별 그래프를 만들어보자
- (다) 성별과 어떤 연령대의 사람들이 많이 수입을 받는가? 남녀의 차이가 가장 적은 연령대는 어떤 연령대인가?

연령대 구분없이 나이와 성별을 이용한 평균표

In [68]:

```
welfare$sex <- ifelse(welfare$sex==1, 'male', 'female')
qplot(welfare$sex)
```



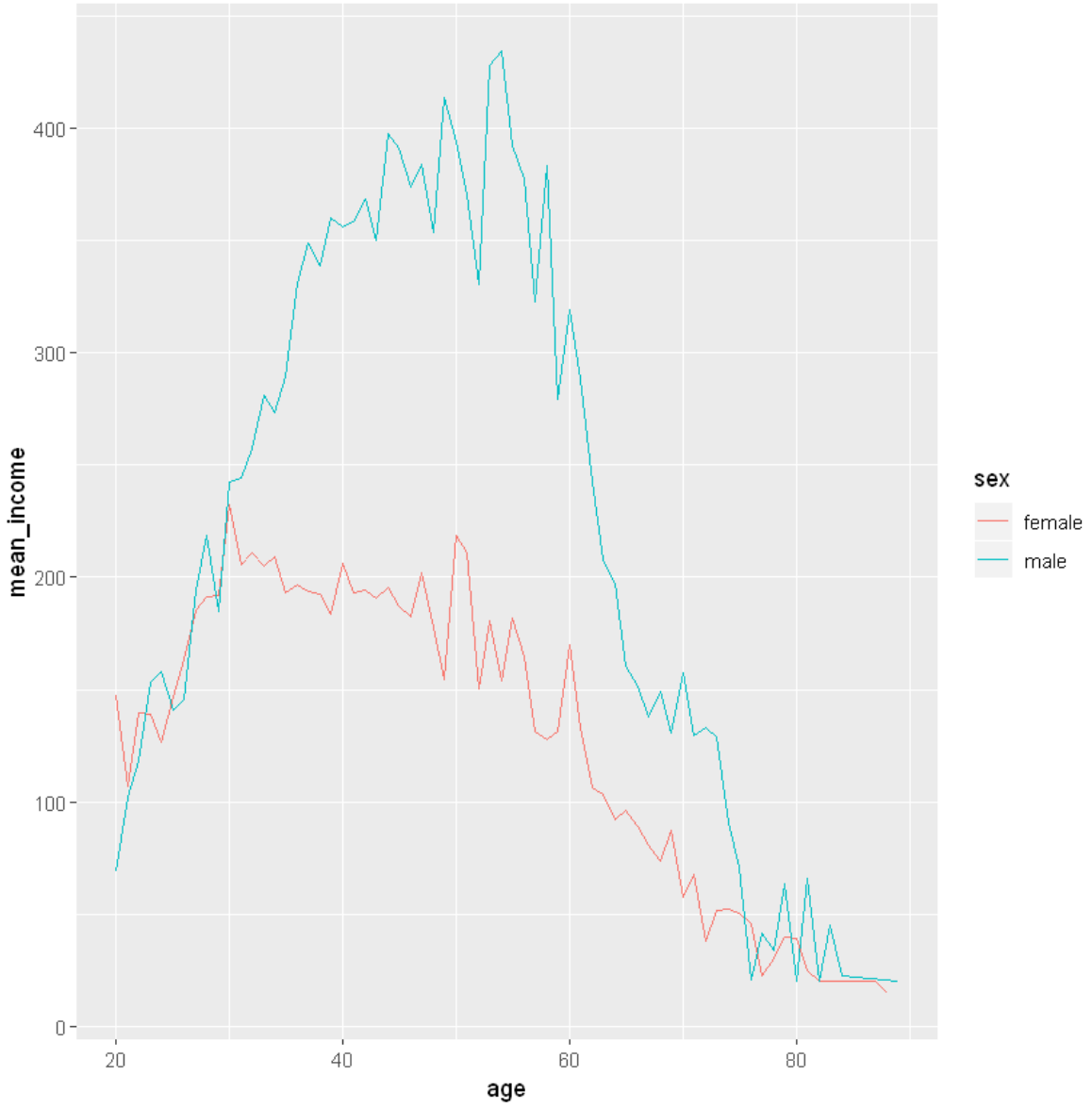
In [70]:

```
sex_age <- welfare %>%
          filter(!is.na(income)) %>%
          group_by(age, sex) %>%
          summarise(mean_income=mean(income))
head(sex_age)
```

age	sex	mean_income
20	female	147.4500
20	male	69.0000
21	female	106.9789
21	male	102.0500
22	female	139.8547
22	male	118.2379

In [71]:

```
ggplot(data=sex_age, aes(x=age, y=mean_income, col=sex)) + geom_line()
```



fill vs col : fill은 색 채우기, col : 테두리 색

In [ ]:

```

```