Titanic - Machine Learning from Disaster

Sangkon Han(sangkon@pusan.ac.kr)

2023-07-06

Contents

Kaggle의 대표적인 Competition 중 하나인 Titanic 생존자 예측에 관한 내용을 다루고 있습니다. 처음 Kaggle에 도전하시는 분들이 참고하실만한 자료가 되었으면 합니다.

데이터 불러오기

titanic competition에서는 Model을 생성하는데 사용하는 train data와 실제 예측(추정)에 사용하는 test data가 분리되어 있습니다. 여기서는 저 2개 data들을 불러와서 하나로 묶을 것 입니다. 따로 분리되어 있는 데이터들을 하나로 묶는 이유는 모델링에 사용되는 입력변수들을 Feature engineering, Pre-processing 할 때 동일하게 작업하기 위해서 입니다.

```
df_titanic <-
  read_csv("data/titanic_train.csv") %>%
  rename_all(tolower)
df_titanic
```

```
## # A tibble: 891 x 12
##    passengerid survived pclass name   sex     age sibsp parch ticket  fare cabin
##          <dbl>    <dbl>  <dbl> <chr>  <chr> <dbl> <dbl> <dbl> <chr>  <dbl> <chr>
##  1           1        0      3 Braun~ male     22     1     0 A/5 2~  7.25 <NA>
##  2           2        1      1 Cumin~ fema~    38     1     0 PC 17~ 71.3  C85
##  3           3        1      3 Heikk~ fema~    26     0     0 STON/~  7.92 <NA>
##  4           4        1      1 Futre~ fema~    35     1     0 113803 53.1  C123
##  5           5        0      3 Allen~ male     35     0     0 373450  8.05 <NA>
##  6           6        0      3 Moran~ male     NA     0     0 330877  8.46 <NA>
##  7           7        0      1 McCar~ male     54     0     0 17463  51.9  E46
```

```
##  8           8        0      3 Palss~ male      2     3     1 349909 21.1  <NA>
##  9           9        1      3 Johns~ fema~    27     0     2 347742 11.1  <NA>
## 10          10        1      2 Nasse~ fema~    14     1     0 237736 30.1  <NA>
## # i 881 more rows
## # i 1 more variable: embarked <chr>
```

```
df_titanic_competition <-
  read_csv("data/titanic_test.csv") %>%
  rename_all(tolower)
df_titanic_competition
```

```
## # A tibble: 418 x 11
##    passengerid pclass name   sex     age sibsp parch ticket  fare cabin embarked
##          <dbl>  <dbl> <chr>  <chr> <dbl> <dbl> <dbl> <chr>  <dbl> <chr> <chr>
##  1         892      3 Kelly~ male   34.5     0     0 330911  7.83 <NA>  Q
##  2         893      3 Wilke~ fema~  47       1     0 363272  7    <NA>  S
##  3         894      2 Myles~ male   62       0     0 240276  9.69 <NA>  Q
##  4         895      3 Wirz,~ male   27       0     0 315154  8.66 <NA>  S
##  5         896      3 Hirvo~ fema~  22       1     1 31012~ 12.3  <NA>  S
##  6         897      3 Svens~ male   14       0     0 7538    9.22 <NA>  S
##  7         898      3 Conno~ fema~  30       0     0 330972  7.63 <NA>  Q
##  8         899      2 Caldw~ male   26       1     1 248738 29    <NA>  S
##  9         900      3 Abrah~ fema~  18       0     0 2657    7.23 <NA>  C
## 10         901      3 Davie~ male   21       2     0 A/4 4~ 24.2  <NA>  S
## # i 408 more rows
```

데이터 확인 및 변환

```
head(df_titanic)
```

```
## # A tibble: 6 x 12
##   passengerid survived pclass name    sex     age sibsp parch ticket  fare cabin
##         <dbl>    <dbl>  <dbl> <chr>   <chr> <dbl> <dbl> <dbl> <chr>  <dbl> <chr>
## 1           1        0      3 Braund~ male     22     1     0 A/5 2~  7.25 <NA>
## 2           2        1      1 Cuming~ fema~    38     1     0 PC 17~ 71.3  C85
## 3           3        1      3 Heikki~ fema~    26     0     0 STON/~  7.92 <NA>
## 4           4        1      1 Futrel~ fema~    35     1     0 113803 53.1  C123
## 5           5        0      3 Allen,~ male     35     0     0 373450  8.05 <NA>
## 6           6        0      3 Moran,~ male     NA     0     0 330877  8.46 <NA>
## # i 1 more variable: embarked <chr>
```

변수 의미 설명

```
summary(df_titanic)
```

```
##   passengerid       survived          pclass          name
##  Min.   :  1.0   Min.   :0.0000   Min.   :1.000   Length:891
##  1st Qu.:223.5   1st Qu.:0.0000   1st Qu.:2.000   Class :character
##  Median :446.0   Median :0.0000   Median :3.000   Mode  :character
##  Mean   :446.0   Mean   :0.3838   Mean   :2.309
##  3rd Qu.:668.5   3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :891.0   Max.   :1.0000   Max.   :3.000
##
##      sex                 age            sibsp           parch
##  Length:891         Min.   : 0.42   Min.   :0.000   Min.   :0.0000
```

```
##  Class :character   1st Qu.:20.12   1st Qu.:0.000   1st Qu.:0.0000
##  Mode  :character   Median :28.00   Median :0.000   Median :0.0000
##                     Mean   :29.70   Mean   :0.523   Mean   :0.3816
##                     3rd Qu.:38.00   3rd Qu.:1.000   3rd Qu.:0.0000
##                     Max.   :80.00   Max.   :8.000   Max.   :6.0000
##                     NA's   :177
##     ticket               fare           cabin             embarked
##  Length:891         Min.   :  0.00   Length:891         Length:891
##  Class :character   1st Qu.:  7.91   Class :character   Class :character
##  Mode  :character   Median : 14.45   Mode  :character   Mode  :character
##                     Mean   : 32.20
##                     3rd Qu.: 31.00
##                     Max.   :512.33
##
```

변수명	해석(의미)	Type
PassengerID	승객을 구별하는 고유 ID number	Int
Survived	승객의 생존 여부를 나타내며 생존은 1, 사망은 0 입니다.	Factor
Pclass	선실의 등급으로서 1등급(1)부터 3등급(3)까지 3개 범주입니다.	Ord.Factor
Name	승객의 이름	Factor
Sex	승객의 성별	Factor
Age	승객의 나이	Numeric
SibSp	각 승객과 동반하는 형제 또는 배우자의 수를 설명하는 변수이며 0부터 8까지 존재합니다.	Integer
Parch	각 승객과 동반하는 부모님 또는 자녀의 수를 설명하는 변수이며 0부터 9까지 존재합니다.	Integer
Ticket	승객이 탑승한 티켓에 대한 문자열 변수	Factor
Fare	승객이 지금까지 여행하면서 지불한 금액에 대한 변수	Numeric
Cabin	각 승객의 선실을 구분하는 변수이며 범주와 결측치가 너무 많습니다.	Factor
Embarked	승선항, 출항지를 나타내며 C, Q, S 3개 범주이다.	Factor

EDA

Age

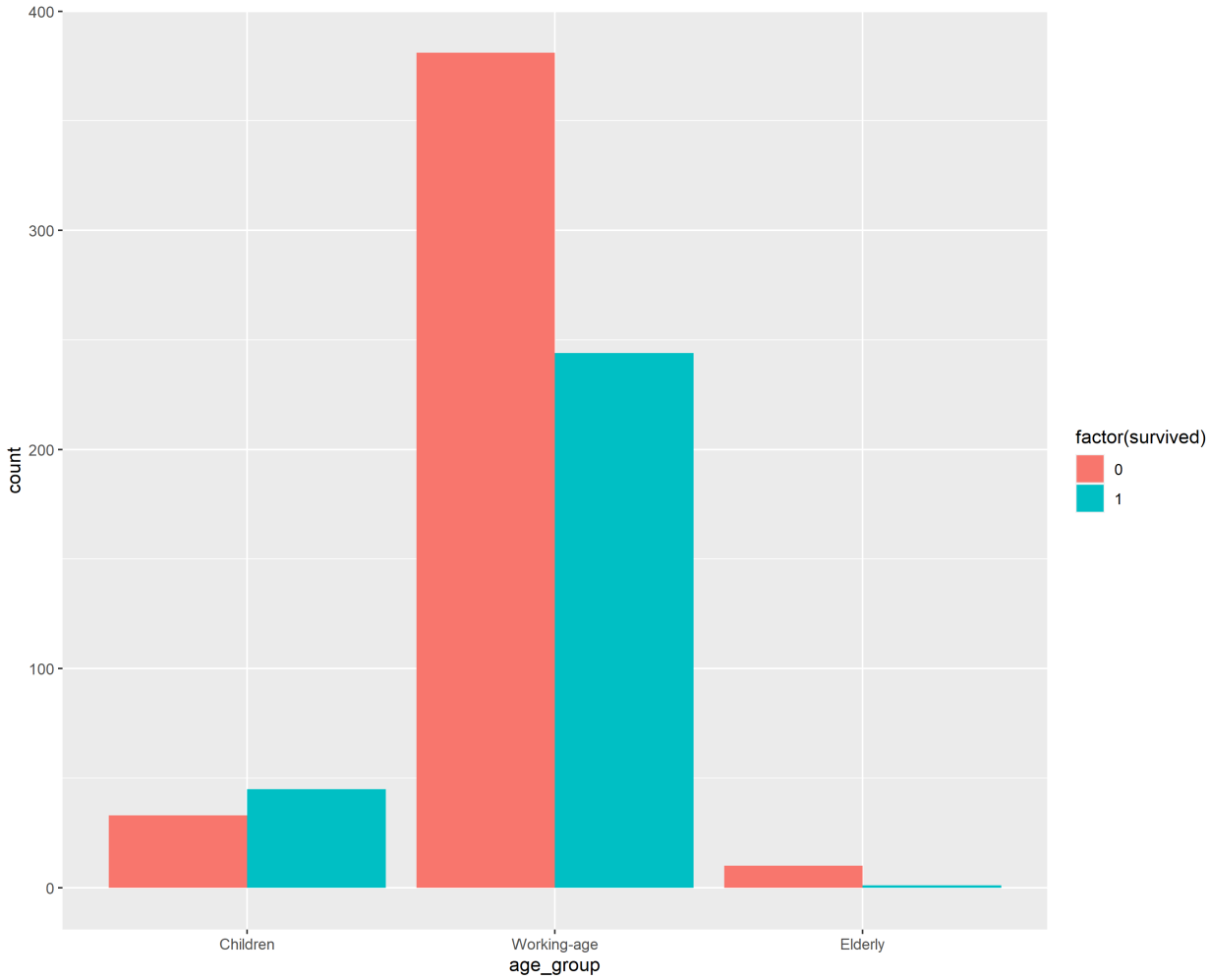
아이들이 다른 연령대에 비해 생존율이 높은 것을 알 수 있습니다.

```
df_titanic %>%
  group_by(survived) %>%
  summarise(mean_age = mean(age, na.rm = TRUE), min_age = min(age, na.rm = TRUE), max_age = max(age, na
```

```
## # A tibble: 2 x 4
##   survived mean_age min_age max_age
##      <dbl>    <dbl>   <dbl>   <dbl>
## 1        0     30.6    1         74
## 2        1     28.3    0.42      80
```

```
df_titanic %>%
  mutate(age_group =  ifelse(age<15, "Children", ifelse(age>=15 & age <=64, "Working-age", "Elderly")))
  filter(!is.na(age_group)) %>%
  ggplot()+
```

```
  geom_bar(mapping = aes(x = factor(age_group, level = c("Children", "Working-age", "Elderly")), fill =
  labs(x = "age_group")
```

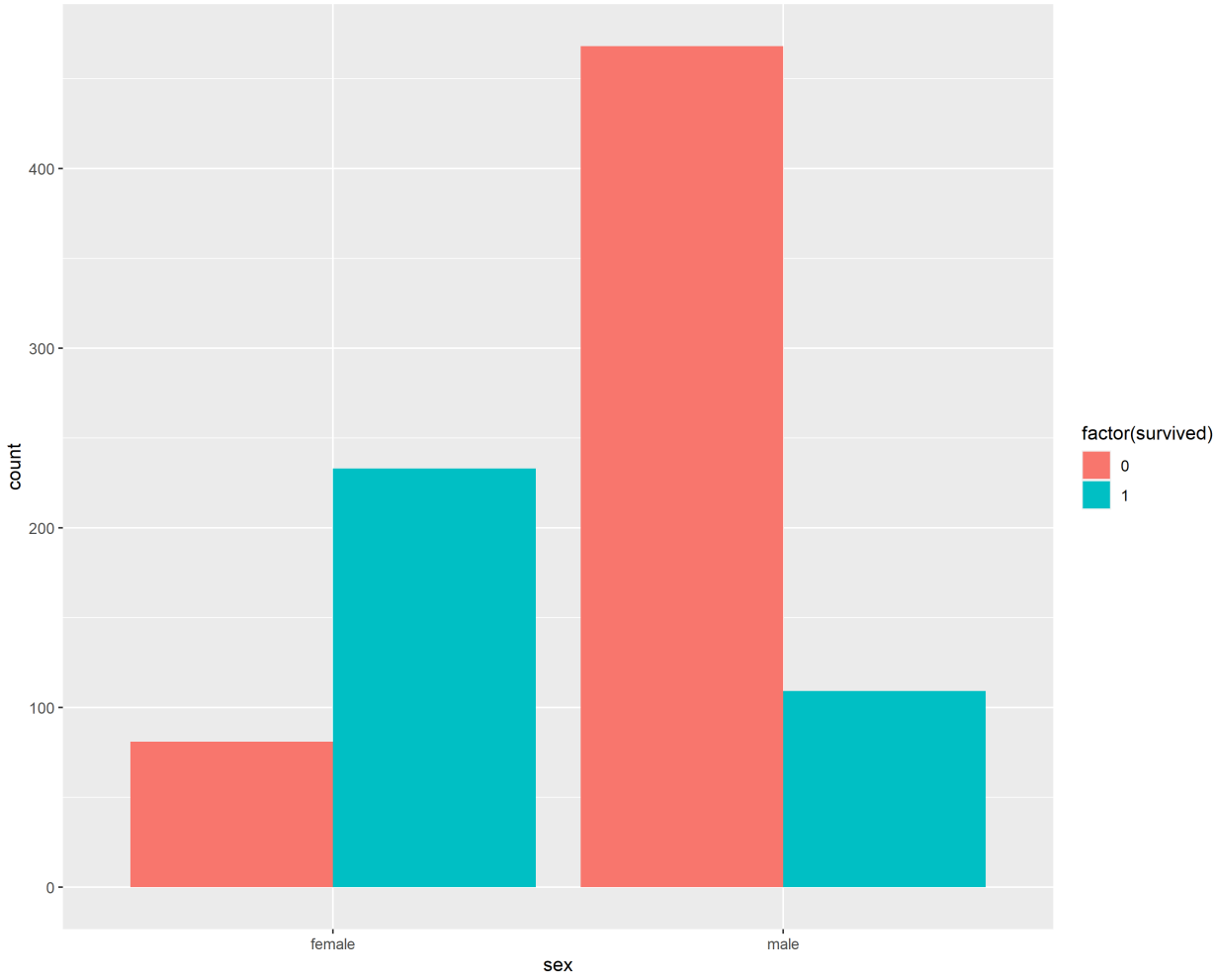


Sex

```
df_titanic %>%
  group_by(survived) %>%
  count(sex)
```

```
## # A tibble: 4 x 3
## # Groups:   survived [2]
##   survived sex        n
##      <dbl> <chr>  <int>
## 1        0 female    81
## 2        0 male     468
## 3        1 female   233
## 4        1 male     109
```

```
df_titanic %>%
  ggplot()+
  geom_bar(mapping = aes(x = sex, fill = factor(survived)), position = "dodge")
```



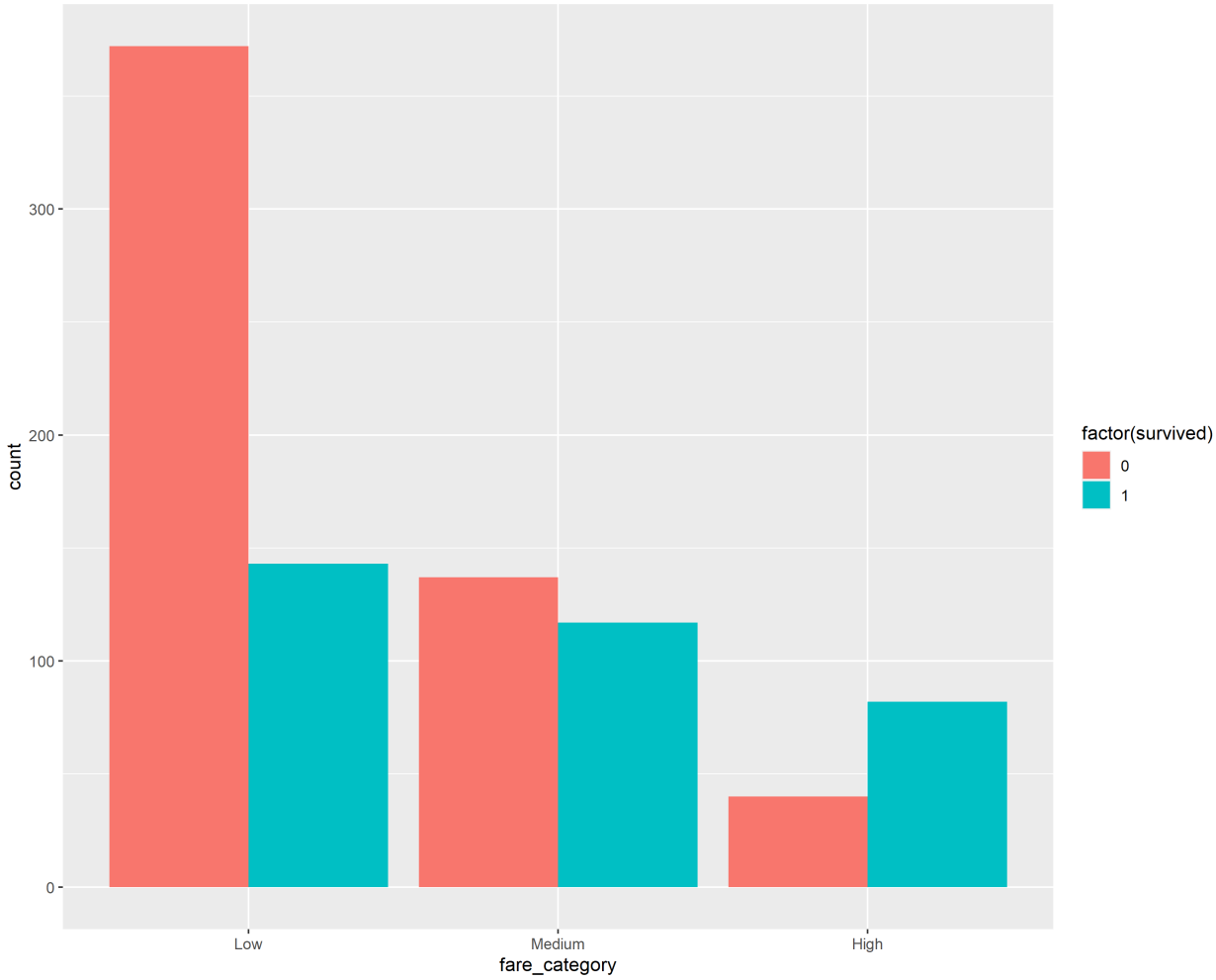
Fare

운임이 높을수록 생존율이 높다는 것을 알 수 있습니다.

```
df_titanic %>%
  group_by(survived) %>%
  summarise(mean_fare = mean(fare, na.rm = TRUE), min_fare = min(fare, na.rm = TRUE), max_fare = max(fa
```

```
## # A tibble: 2 x 4
##   survived mean_fare min_fare max_fare
##      <dbl>     <dbl>    <dbl>    <dbl>
## 1        0      22.1        0     263
## 2        1      48.4        0     512.
```

```
df_titanic %>%
  mutate(fare_category =  ifelse(fare<20, "Low", ifelse(fare>=20 & fare <=60, "Medium", "High"))) %>%
  ggplot()+
  geom_bar(mapping = aes(x = factor(fare_category, level = c("Low", "Medium", "High")), fill = factor(su
  labs(x = "fare_category")
```



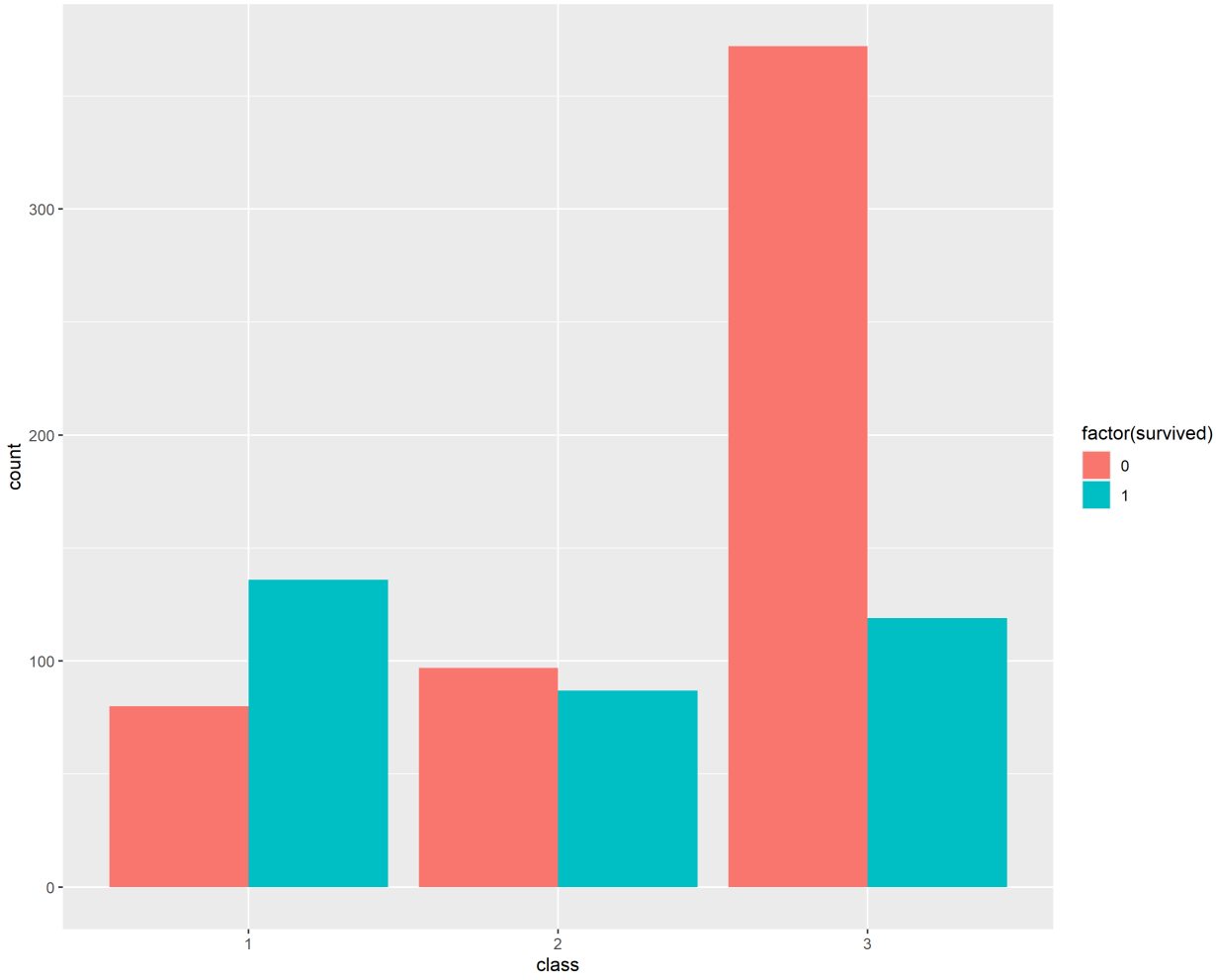
Class

상위 클래스는 다른 클래스에 비해 생존율이 높은 것을 알 수 있습니다.

```
df_titanic %>%
  group_by(survived) %>%
  count(pclass)
```

```
## # A tibble: 6 x 3
## # Groups:   survived [2]
##   survived pclass     n
##      <dbl>  <dbl> <int>
## 1        0      1    80
## 2        0      2    97
## 3        0      3   372
## 4        1      1   136
## 5        1      2    87
## 6        1      3   119
```

```
df_titanic %>%
  ggplot()+
  geom_bar(mapping = aes(x = factor(pclass), fill = factor(survived)), position = "dodge")+
  labs(x = "class")
```



Family on board

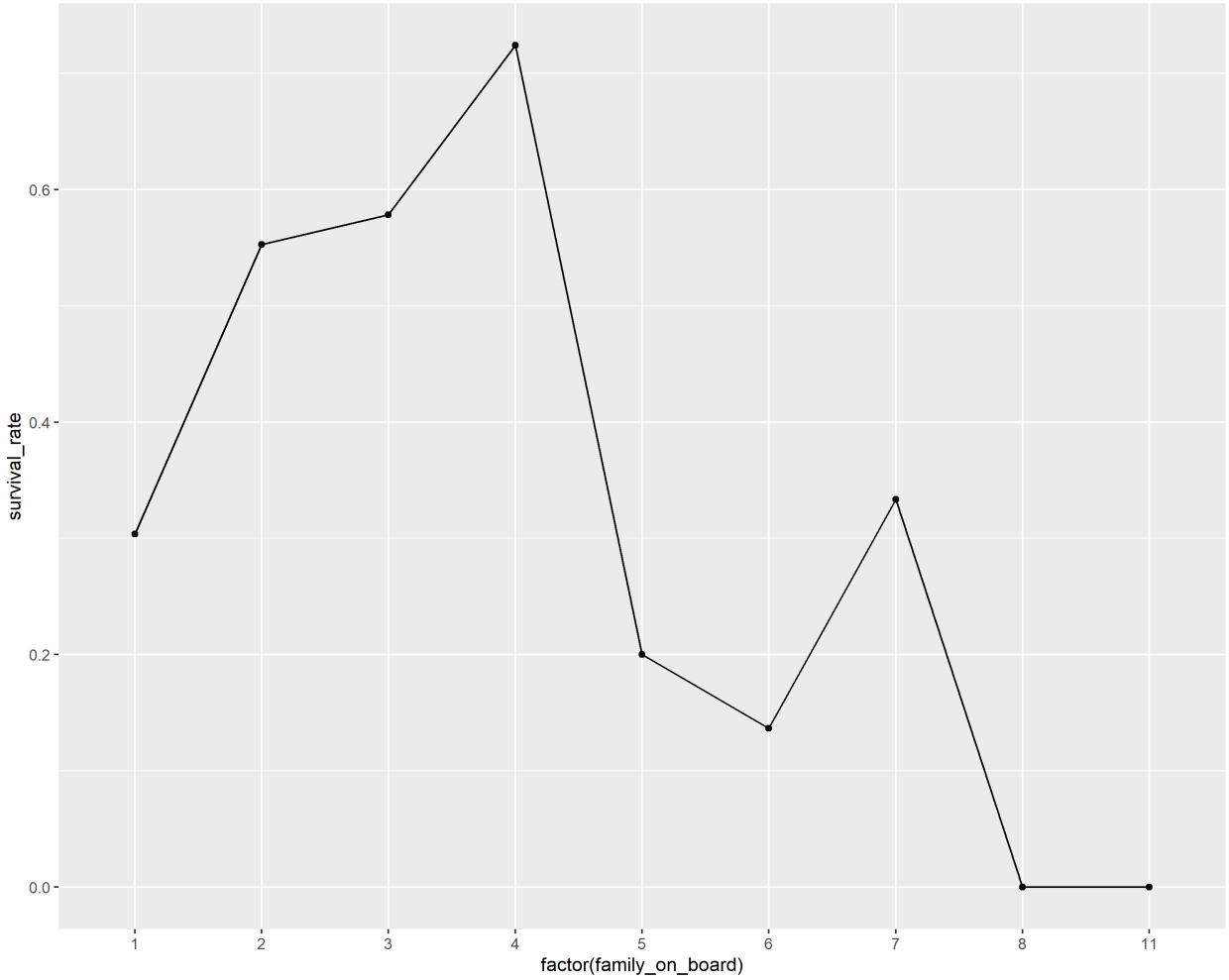
가족 규모가 3-4명에 이르는 경우 생존율이 가장 높은 것을 알 수 있습니다.

```
df_titanic %>%
  mutate(family_on_board = sibsp + parch + 1) %>%
  group_by(survived) %>%
  count(family_on_board)
```

```
## # A tibble: 16 x 3
## # Groups:   survived [2]
##    survived family_on_board     n
##       <dbl>           <dbl> <int>
##  1        0               1   374
##  2        0               2    72
##  3        0               3    43
##  4        0               4     8
##  5        0               5    12
##  6        0               6    19
##  7        0               7     8
##  8        0               8     6
##  9        0              11     7
```

```
## 10         1                1   163
## 11         1                2    89
## 12         1                3    59
## 13         1                4    21
## 14         1                5     3
## 15         1                6     3
## 16         1                7     4
```

```
df_titanic <- df_titanic %>%
  mutate(family_on_board = sibsp + parch + 1)

df_titanic %>%
  group_by(family_on_board) %>%
  mutate(survival_rate = sum(survived)/n()) %>%
  ggplot(mapping = aes(x = factor(family_on_board), y = survival_rate, group = 1 ))+
  geom_point()+
  geom_line()
```



Embarkation

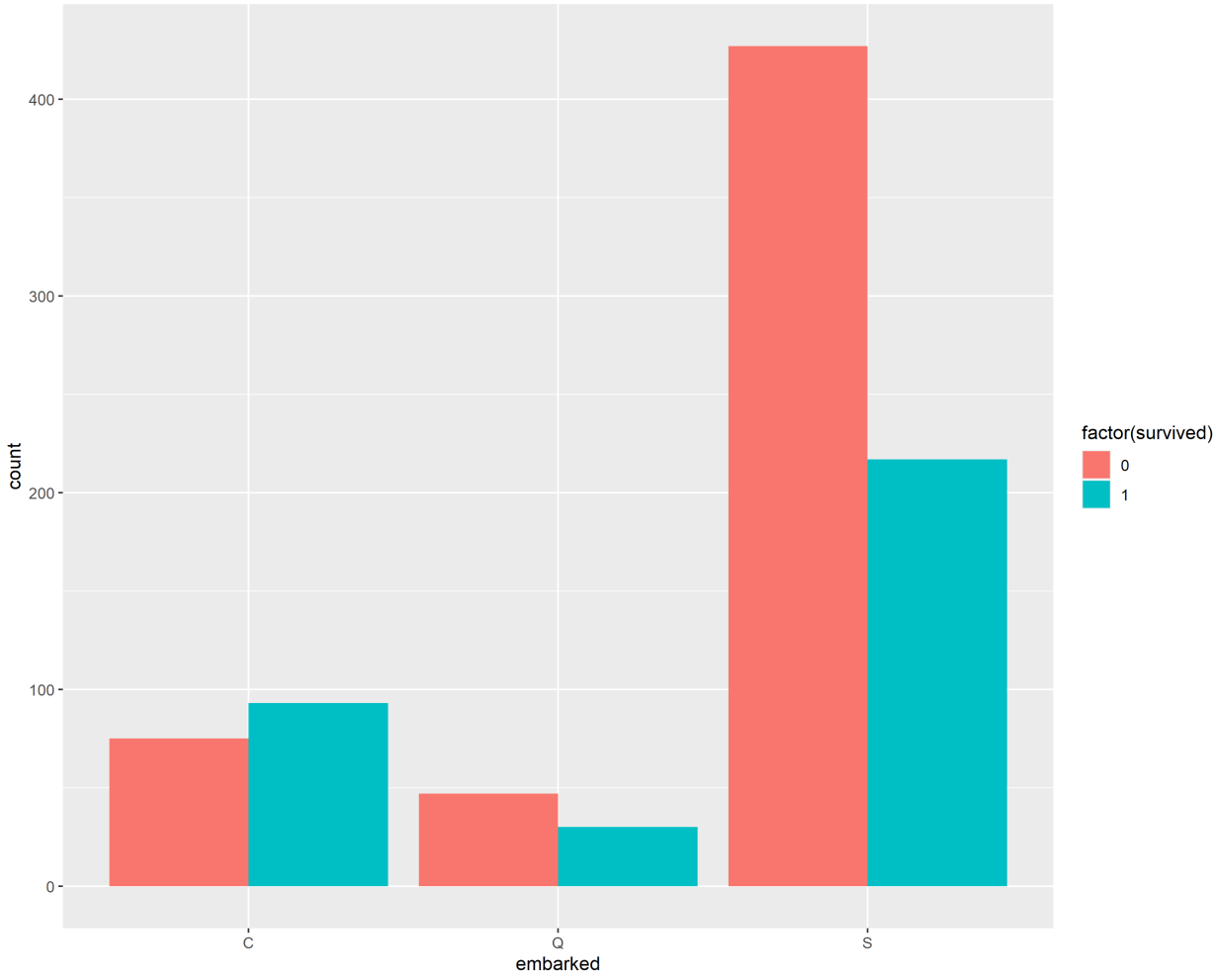
C 승선은 다른 승선에 비해 생존율이 높은 것 같습니다.

```
df_titanic %>%
  group_by(survived) %>%
```

```
  count(embarked)
```

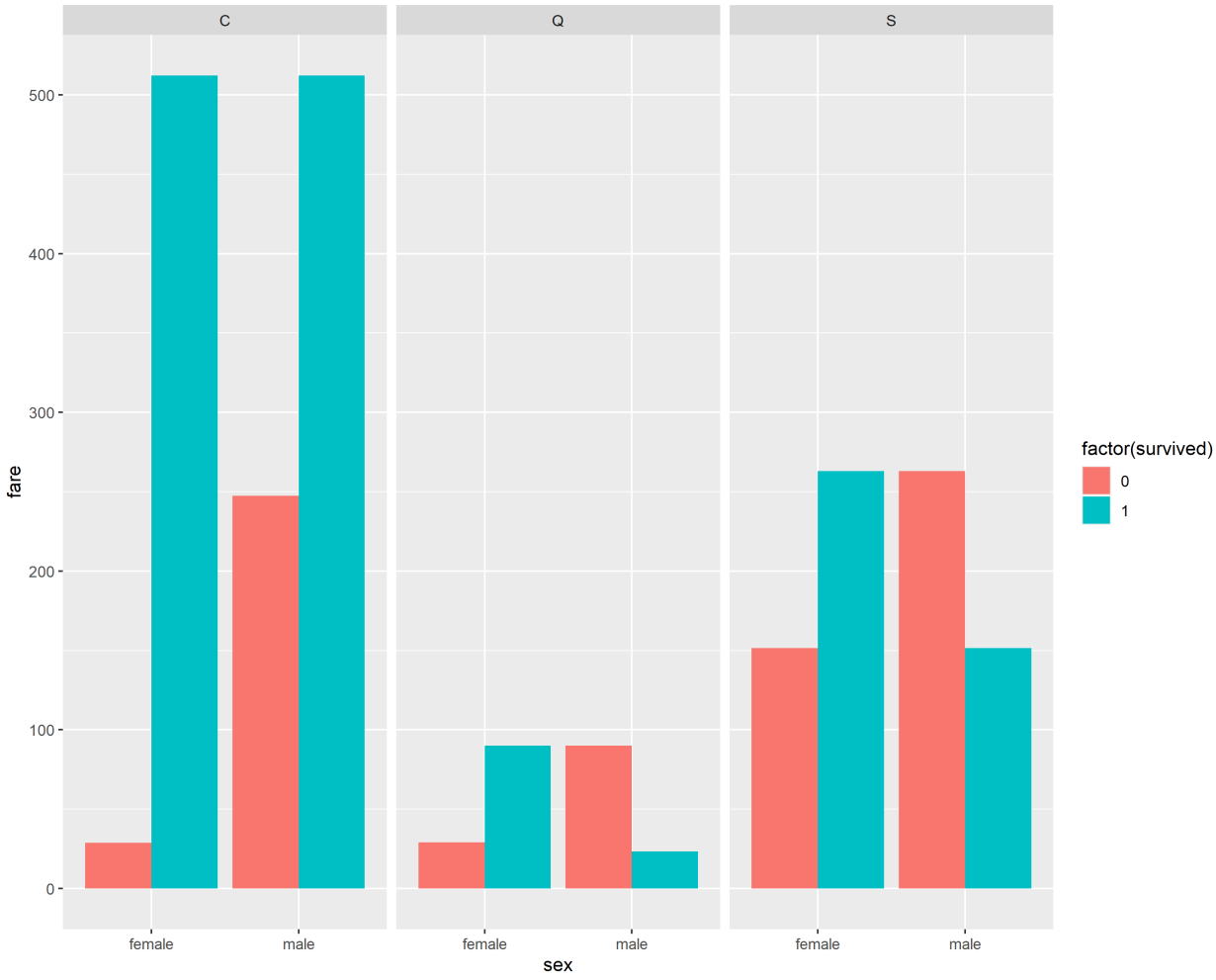
```
## # A tibble: 7 x 3
## # Groups:   survived [2]
##   survived embarked     n
##      <dbl> <chr>    <int>
## 1        0 C           75
## 2        0 Q           47
## 3        0 S          427
## 4        1 C           93
## 5        1 Q           30
## 6        1 S          217
## 7        1 <NA>         2
```

```
df_titanic %>%
  filter(!is.na(embarked)) %>%
  ggplot()+
  geom_bar(mapping = aes(x = embarked, fill = factor(survived)), position = "dodge")
```




```
df_titanic %>%
  filter(!is.na(embarked)) %>%
  ggplot()+
```

```
  geom_col(mapping = aes(x = sex, y = fare, fill = factor(survived)), position = "dodge")+
  facet_wrap(~embarked)
```



전처리

```
mice_mod <- mice(df_titanic[, c("age","fare", "sex", "pclass", "embarked")], method='cart')
```

```
##
##  iter imp variable
##    1   1  age
##    1   2  age
##    1   3  age
##    1   4  age
##    1   5  age
##    2   1  age
##    2   2  age
##    2   3  age
##    2   4  age
##    2   5  age
##    3   1  age
```

```
##   3   2  age
##   3   3  age
##   3   4  age
##   3   5  age
##   4   1  age
##   4   2  age
##   4   3  age
##   4   4  age
##   4   5  age
##   5   1  age
##   5   2  age
##   5   3  age
##   5   4  age
##   5   5  age
```

```
mice_complete <- complete(mice_mod)
df_titanic$age <- mice_complete$age
df_titanic$age <- mice_complete$age
df_titanic$fare <- mice_complete$fare
df_titanic$sex <- mice_complete$sex
df_titanic$pclass <- mice_complete$pclass
df_titanic$embarked <- mice_complete$embarked
```

```
df_titanic <-
  df_titanic %>%
  mutate(age_group =  ifelse(age<15, "1", ifelse(age>=15 & age <=64, "2", "3"))) %>%
  mutate(fare_category =  ifelse(fare<20, "1", ifelse(fare>=20 & fare <=60, "2", "3")))
```

모델링

```
set.seed(42)
df_titanic_xg <- df_titanic %>%
              mutate(survived = as.factor(survived),
                     age_group = as.numeric(age_group),
                     fare_category = as.numeric(fare_category),
                     sex = as.numeric(ifelse(sex == "male", 1, 0)),
                     pclass = as.numeric(pclass),
                     family_on_board = as.numeric(family_on_board))
```

XGBoost

```
df_split_xg <- initial_split(df_titanic_xg)
df_train_xg <- training(df_split_xg)
df_test_xg <- testing(df_split_xg)

xgb_spec <- boost_tree(
  trees = 1000,
  tree_depth = tune(), min_n = tune(),
  loss_reduction = tune(),
  sample_size = tune(), mtry = tune(),
  learn_rate = tune()
) %>%
  set_engine("xgboost") %>%
```

```
  set_mode("classification")

xgb_grid <- grid_latin_hypercube(
  tree_depth(),
  min_n(),
  loss_reduction(),
  sample_size = sample_prop(),
  finalize(mtry(), df_train_xg),
  learn_rate(),
  size = 10
)

xgb_grid
```

```
## # A tibble: 10 x 6
##    tree_depth min_n loss_reduction sample_size  mtry learn_rate
##         <int> <int>          <dbl>       <dbl> <int>      <dbl>
##  1         11    33       1.72e+ 1       0.118     5   4.19e- 8
##  2          2    30       4.97e- 6       0.756    14   1.03e- 4
##  3         10    11       1.71e- 6       0.473    10   5.31e- 4
##  4          9    37       9.16e- 9       0.263     1   5.31e- 3
##  5         12    18       1.43e+ 0       0.398     6   1.35e- 6
##  6          8     5       9.12e- 3       0.636     4   5.85e-10
##  7          5     8       2.29e- 7       0.889     9   1.99e- 2
##  8          5    22       2.67e- 4       0.707    12   3.41e- 7
##  9          3    28       1.43e-10       0.935    12   4.28e- 6
## 10         14    17       1.58e- 1       0.340     7   4.28e- 9
```

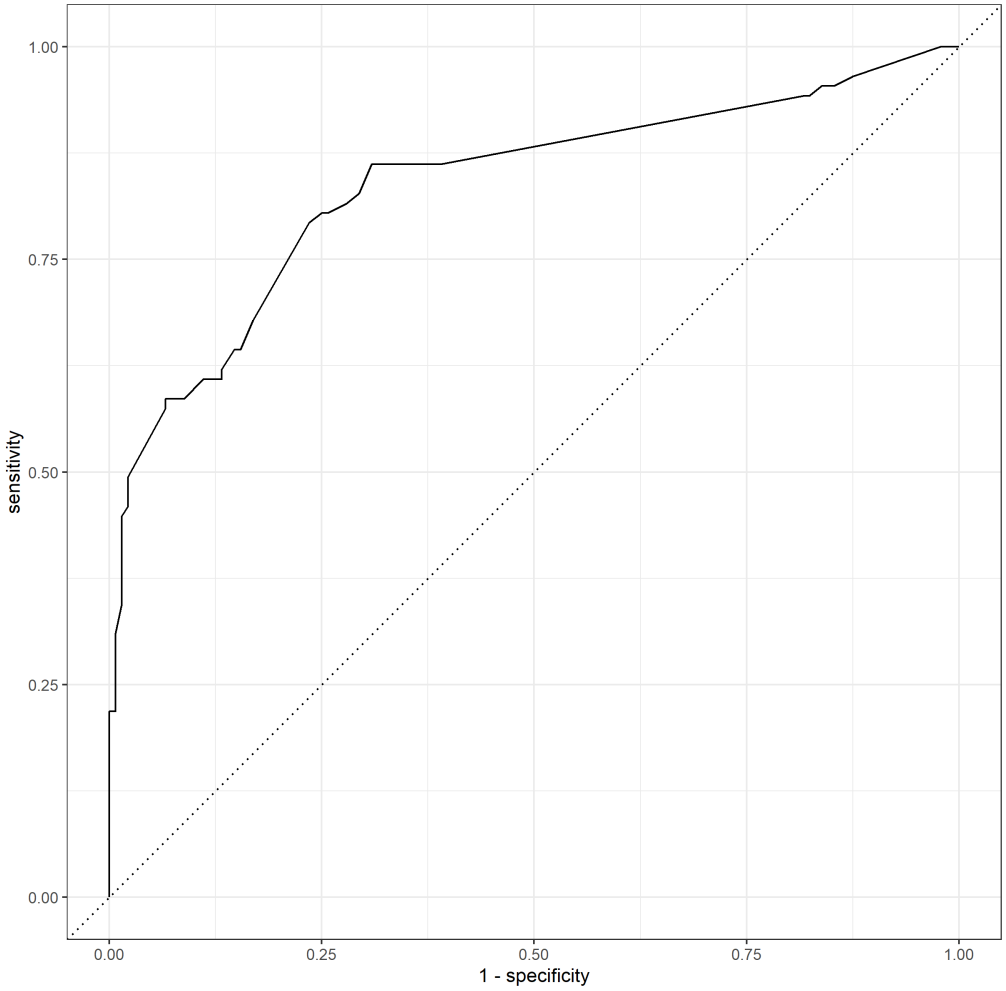
```
recipe_xg <-
  recipe(survived ~ age_group + fare_category + sex + pclass + family_on_board, data = df_train_xg)

xgb_wf <- workflow() %>%
  add_recipe(recipe_xg) %>%
  add_model(xgb_spec)

dfa_folds <- vfold_cv(df_train_xg)

doParallel::registerDoParallel()

xgb_res <- tune_grid(
  xgb_wf,
  resamples = dfa_folds,
  grid = xgb_grid,
  control = control_grid(save_pred = TRUE)
)

best_auc <- select_best(xgb_res, "roc_auc")
best_auc
```

```
## # A tibble: 1 x 7
##    mtry min_n tree_depth learn_rate loss_reduction sample_size .config
##   <int> <int>      <int>      <dbl>          <dbl>       <dbl> <chr>
## 1     9     8          5     0.0199    0.000000229       0.889 Preprocessor1_Mo~
```

```r
final_xgb <- finalize_workflow(
  xgb_wf,
  best_auc
)

fitxgb <-fit(final_xgb, data = df_train_xg)

results_xg <-
  predict(fitxgb, df_test_xg, type = 'prob') %>%
  pluck(2) %>%
  bind_cols(df_test_xg, Predicted_Probability = .) %>%
  mutate(predictedClass = as.factor(ifelse(Predicted_Probability > 0.5, 2, 1)))

roc_auc(results_xg, truth = survived, Predicted_Probability, event_level = 'second')
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.837
```

```r
roc_curve(results_xg, truth = survived,
          Predicted_Probability,
          event_level = 'second') %>%
  ggplot(aes(x = 1 - specificity,
             y = sensitivity)) +
  geom_path() +
  geom_abline(lty = 3) +
  coord_equal() +
  theme_bw()
```



예측과 정답지

```
mice_mod_competition <- mice(df_titanic_competition[, c("age","fare", "sex", "pclass", "embarked")], me
```

```
##
##  iter imp variable
##   1   1  age  fare
##   1   2  age  fare
##   1   3  age  fare
##   1   4  age  fare
##   1   5  age  fare
##   2   1  age  fare
##   2   2  age  fare
##   2   3  age  fare
##   2   4  age  fare
##   2   5  age  fare
##   3   1  age  fare
##   3   2  age  fare
##   3   3  age  fare
##   3   4  age  fare
##   3   5  age  fare
```

```
##   4   1  age  fare
##   4   2  age  fare
##   4   3  age  fare
##   4   4  age  fare
##   4   5  age  fare
##   5   1  age  fare
##   5   2  age  fare
##   5   3  age  fare
##   5   4  age  fare
##   5   5  age  fare
```

```
mice_complete_competition <- complete(mice_mod_competition)
df_titanic_competition$age <- mice_complete_competition$age
df_titanic_competition$fare <- mice_complete_competition$fare
df_titanic_competition$sex <- mice_complete_competition$sex
df_titanic_competition$pclass <- mice_complete_competition$pclass
df_titanic_competition$embarked <- mice_complete_competition$embarked
df_titanic_competition <- df_titanic_competition %>%
  mutate(family_on_board = sibsp + parch + 1) %>%
  mutate(age_group =  ifelse(age<15, "1", ifelse(age>=15 & age <=64, "2", "3"))) %>%
  mutate(fare_category =  ifelse(fare<20, "1", ifelse(fare>=20 & fare <=60, "2", "3")))
df_titanic_competition_xg <-
  df_titanic_competition %>%
  mutate(age_group = as.numeric(age_group),
                     fare_category = as.numeric(fare_category),
                     sex = as.numeric(ifelse(sex == "male", 1, 0)),
                     pclass = as.numeric(pclass),
                     family_on_board = as.numeric(family_on_board))
Prediction <-
  predict(fitxgb, df_titanic_competition_xg) %>%
  pluck(1) %>%
  bind_cols(df_titanic_competition_xg$passengerid, Predicted_Class = .)
Prediction_xg <-
  Prediction %>%
  mutate(Survived = Predicted_Class, PassengerId = ...1) %>%
  select(PassengerId, Survived)
write.csv(Prediction_xg, file = "Titanic_XGBoost.csv", row.names = FALSE)
```