Titanic Machine Learning from Disaster

Seunghwan Lee

2020 09 28

```
#setwd("C:/kaggle/Titanic Machine Learning from Disaster")
#getwd()
#list.files()

# Multiple plot function
#
# ggplot objects can be passed in ..., or to plotlist (as a list of ggplot objects)
# - cols:   Number of columns in layout
# - layout: A matrix specifying the layout. If present, 'cols' is ignored.
#
# If the layout is something like matrix(c(1,2,3,3), nrow=2, byrow=TRUE),
# then plot 1 will go in the upper left, 2 will go in the upper right, and
# 3 will go all the way across the bottom.
#
multiplot <- function(..., plotlist=NULL, file, cols=1, layout=NULL) {
  library(grid)

  # Make a list from the ... arguments and plotlist
  plots <- c(list(...), plotlist)

  numPlots = length(plots)

  # If layout is NULL, then use 'cols' to determine layout
  if (is.null(layout)) {
    # Make the panel
    # ncol: Number of columns of plots
    # nrow: Number of rows needed, calculated from # of cols
    layout <- matrix(seq(1, cols * ceiling(numPlots/cols)),
                     ncol = cols, nrow = ceiling(numPlots/cols))
  }

  if (numPlots==1) {
    print(plots[[1]])

  } else {
    # Set up the page
    grid.newpage()
    pushViewport(viewport(layout = grid.layout(nrow(layout), ncol(layout))))

    # Make each plot, in the correct location
    for (i in 1:numPlots) {
      # Get the i,j matrix positions of the regions that contain this subplot
```

```
      matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE))

      print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row,
                                      layout.pos.col = matchidx$col))
    }
  }
}
# http://www.cookbook-r.com/Graphs/Multiple_graphs_on_one_page_(ggplot2)/

# Data input, assesment : 데이터 불러들이기, 확인하는 과정
library(readr) # Data input with readr::read_csv()
library(descr) # descr::CrossTable() - 범주별 빈도수, 비율 수치로 확인

# Visualization
library(VIM)             # Missing values assesment used by VIM::aggr()
```

```
## Loading required package: colorspace
```

```
## Loading required package: grid
```

```
## VIM is ready to use.
```

```
## Suggestions and bug-reports can be submitted at: https://github.com/statistikat/VIM/issues
```

```
##
## Attaching package: 'VIM'
```

```
## The following object is masked from 'package:datasets':
##
##     sleep
```

```
library(ggplot2)         # Used in almost visualization
library(RColorBrewer)    # plot의 color 설정
library(scales)          # plot setting - x, y 축 설정
```

```
##
## Attaching package: 'scales'
```

```
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
# Feature engineering, Data Pre-processing
library(tidyverse)     # dplyr, ggplot2, purrr, etc...
```

```
## -- Attaching packages ---------------------------------------------------------------------------
```

```
## v tibble  3.0.3     v dplyr   1.0.1
## v tidyr   1.1.1     v stringr 1.4.0
## v purrr   0.3.4     v forcats 0.5.0
```

```
## -- Conflicts ------------------------------------------------------------------------------------
## x scales::col_factor() masks readr::col_factor()
## x purrr::discard()     masks scales::discard()
## x dplyr::filter()      masks stats::filter()
## x dplyr::lag()         masks stats::lag()
```

```
library(dplyr)           # Feature Engineering & Data Pre-processing
library(purrr)           # Check missing values
library(tidyr)           # tidyr::gather()
```

```r
library(rpart)           # prediction(tree)

train <- readr::read_csv('train.csv')
```

```
## Parsed with column specification:
## cols(
##   PassengerId = col_double(),
##   Survived = col_double(),
##   Pclass = col_double(),
##   Name = col_character(),
##   Sex = col_character(),
##   Age = col_double(),
##   SibSp = col_double(),
##   Parch = col_double(),
##   Ticket = col_character(),
##   Fare = col_double(),
##   Cabin = col_character(),
##   Embarked = col_character()
## )
```

```r
test  <- readr::read_csv('test.csv')
```

```
## Parsed with column specification:
## cols(
##   PassengerId = col_double(),
##   Pclass = col_double(),
##   Name = col_character(),
##   Sex = col_character(),
##   Age = col_double(),
##   SibSp = col_double(),
##   Parch = col_double(),
##   Ticket = col_character(),
##   Fare = col_double(),
##   Cabin = col_character(),
##   Embarked = col_character()
## )
```

```r
# rbind(train,test) # There is no Survived variable in test set
full <- dplyr::bind_rows(train, test)

full <- full %>%
  dplyr::mutate(Survived = factor(Survived),
                Pclass   = factor(Pclass, ordered=F),
                Name     = factor(Name),
                Sex      = factor(Sex),
                Ticket   = factor(Ticket),
                Cabin    = factor(Cabin),
                Embarked = factor(Embarked))
str(full)
```

```
## tibble [1,309 x 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ PassengerId: num [1:1309] 1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 1 2 2 ...
##  $ Pclass     : Factor w/ 3 levels "1","2","3": 3 1 3 1 3 3 1 3 3 2 ...
##  $ Name       : Factor w/ 1307 levels "Abbing, Mr. Anthony",..: 156 287 531 430 23 826 775 922 613 85
```

```
##  $ Sex         : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
##  $ Age         : num [1:1309] 22 38 26 35 35 NA 54 2 27 14 ...
##  $ SibSp       : num [1:1309] 1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch       : num [1:1309] 0 0 0 0 0 0 0 1 2 0 ...
##  $ Ticket      : Factor w/ 929 levels "110152","110413",..: 721 817 915 66 650 374 110 542 478 175 ..
##  $ Fare        : num [1:1309] 7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin       : Factor w/ 186 levels "A10","A11","A14",..: NA 107 NA 71 NA NA 164 NA NA NA ...
##  $ Embarked    : Factor w/ 3 levels "C","Q","S": 3 1 3 3 3 2 3 3 3 1 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   PassengerId = col_double(),
##   ..   Survived = col_double(),
##   ..   Pclass = col_double(),
##   ..   Name = col_character(),
##   ..   Sex = col_character(),
##   ..   Age = col_double(),
##   ..   SibSp = col_double(),
##   ..   Parch = col_double(),
##   ..   Ticket = col_character(),
##   ..   Fare = col_double(),
##   ..   Cabin = col_character(),
##   ..   Embarked = col_character()
##   .. )
```

```
summary(full)
```

```
##   PassengerId   Survived   Pclass                                      Name
##  Min.   :   1   0   :549   1:323   Connolly, Miss. Kate                  :   2
##  1st Qu.: 328   1   :342   2:277   Kelly, Mr. James                      :   2
##  Median : 655   NA's:418   3:709   Abbing, Mr. Anthony                   :   1
##  Mean   : 655                      Abbott, Master. Eugene Joseph         :   1
##  3rd Qu.: 982                      Abbott, Mr. Rossmore Edward           :   1
##  Max.   :1309                      Abbott, Mrs. Stanton (Rosa Hunt):   1
##                                    (Other)                               :1301
##      Sex           Age            SibSp           Parch           Ticket
##  female:466   Min.   : 0.17   Min.   :0.0000   Min.   :0.000   CA. 2343:  11
##  male  :843   1st Qu.:21.00   1st Qu.:0.0000   1st Qu.:0.000   1601    :   8
##               Median :28.00   Median :0.0000   Median :0.000   CA 2144 :   8
##               Mean   :29.88   Mean   :0.4989   Mean   :0.385   3101295 :   7
##               3rd Qu.:39.00   3rd Qu.:1.0000   3rd Qu.:0.000   347077  :   7
##               Max.   :80.00   Max.   :8.0000   Max.   :9.000   347082  :   7
##               NA's   :263                                      (Other) :1261
##       Fare                     Cabin      Embarked
##  Min.   :  0.000   C23 C25 C27    :   6   C   :270
##  1st Qu.:  7.896   B57 B59 B63 B66:   5   Q   :123
##  Median : 14.454   G6             :   5   S   :914
##  Mean   : 33.295   B96 B98        :   4   NA's:  2
##  3rd Qu.: 31.275   C22 C26        :   4
##  Max.   :512.329   (Other)        : 271
##  NA's   :1         NA's           :1014
```

```
# Unique value of variables
lapply(full, function(x) length(unique(x)))
```

```
## $PassengerId
```

```
## [1] 1309
##
## $Survived
## [1] 3
##
## $Pclass
## [1] 3
##
## $Name
## [1] 1307
##
## $Sex
## [1] 2
##
## $Age
## [1] 99
##
## $SibSp
## [1] 7
##
## $Parch
## [1] 8
##
## $Ticket
## [1] 929
##
## $Fare
## [1] 282
##
## $Cabin
## [1] 187
##
## $Embarked
## [1] 4
```

```
# Missing values
require(moonBook)
```

```
## Loading required package: moonBook
```

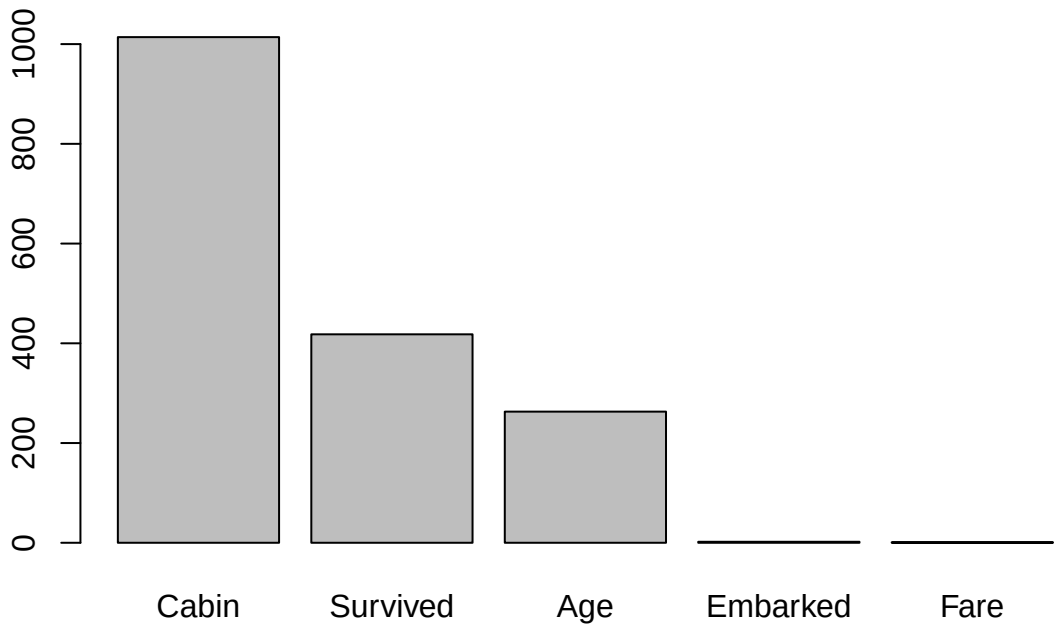
```
##
## Attaching package: 'moonBook'
```

```
## The following object is masked from 'package:scales':
##
##     comma
```

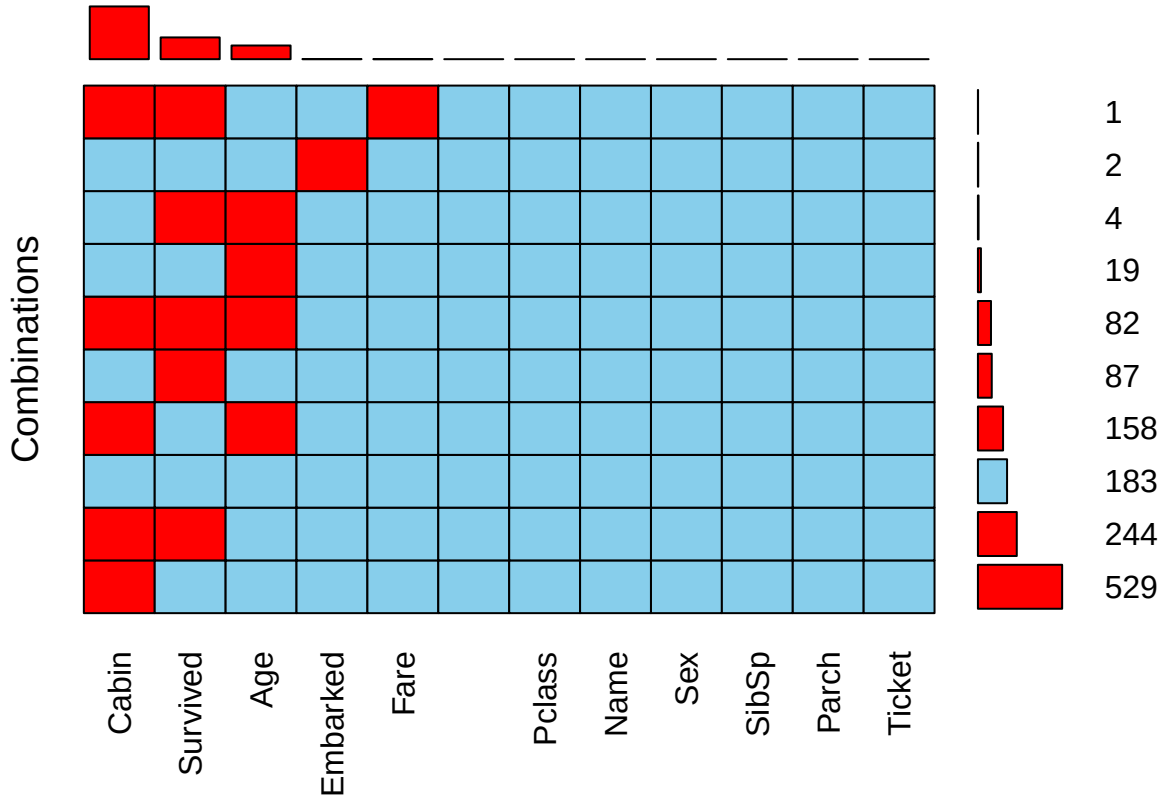
```
na.count=apply(full, 2, function(x) sum(is.na(x)))
na.count[na.count>0]
```

```
## Survived      Age     Fare    Cabin Embarked
##      418      263        1     1014        2
```

```r
sort(na.count[na.count>0], decreasing = T) %>% barplot
```

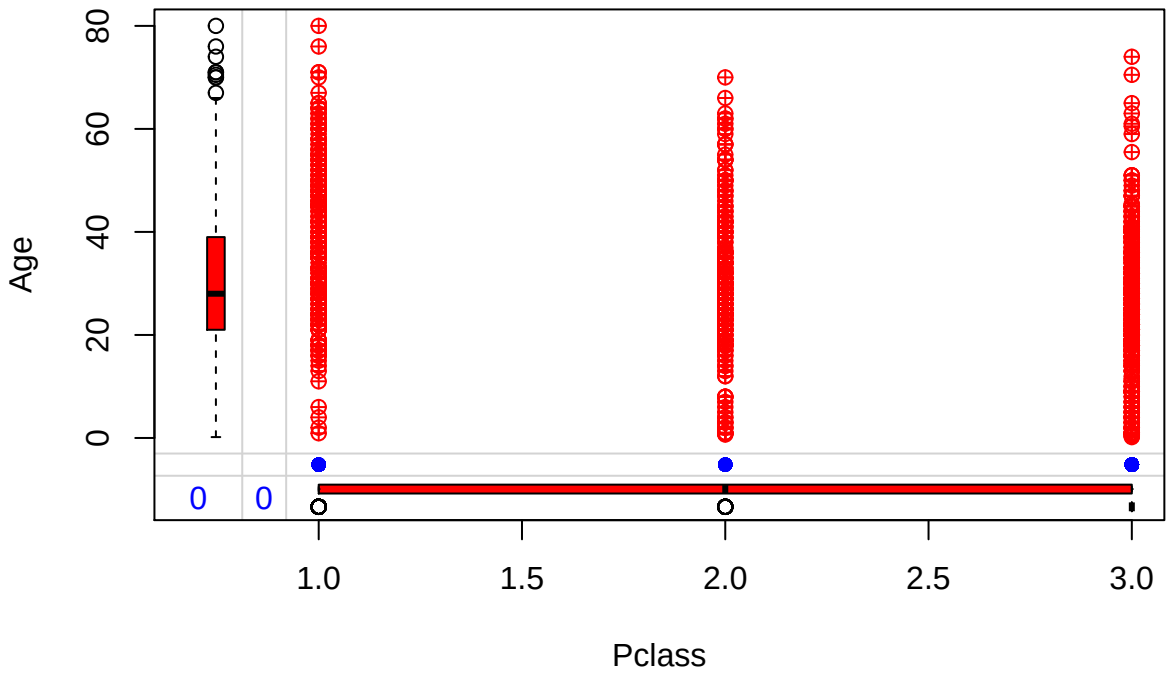



```r
require(VIM)
# prop: 결측치를 비율로 표시
# combined: 그래프를 합쳐서 하나로 표시
# numbers: 결합 누적 개수를 표시
# sortVars: 변수들을 sort
# sortCombs: 결합 변수들 sort
aggr(full, prop = FALSE, combined = TRUE, numbers = TRUE,
          sortVars = TRUE, sortCombs = TRUE)
```




```
##
##  Variables sorted by number of missings:
##      Variable Count
##         Cabin  1014
##      Survived   418
##           Age   263
##      Embarked     2
##          Fare     1
##   PassengerId     0
##        Pclass     0
##          Name     0
##           Sex     0
##         SibSp     0
##         Parch     0
##        Ticket     0
```

```
# cabin 결측치 529 -> cabin & Survived 결측치 244
marginplot(full[c("Pclass","Age")], pch=10, col=c("red", "blue"))
```

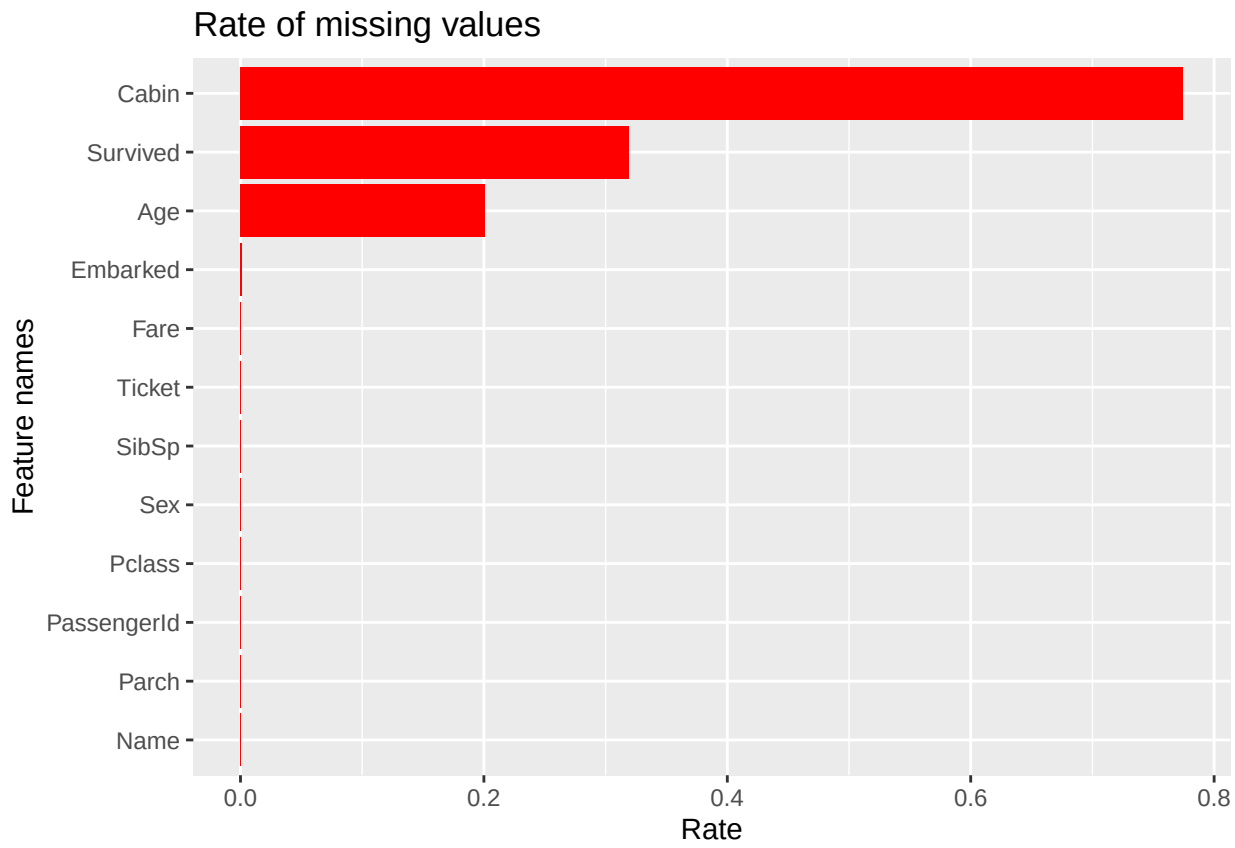



```
# ex) gather function
#iris.df = as.data.frame(iris)
#iris.df$row <- 1:nrow(iris.df)
#IRIS <- arrange(sample_n(iris.df[, -c(3:4)], 10), Species)
#IRIS
#iris_gather1 <- gather(IRIS, type, value, 1:2)
#iris_gather1
#iris_gather2 <- gather(IRIS, type, value, -Species, -row)
#iris_gather2
#gather(IRIS, key="Species", value="row")

# Check for missing values
missing_values <- full %>% summarize_all(funs(sum(is.na(.))/n()))
```

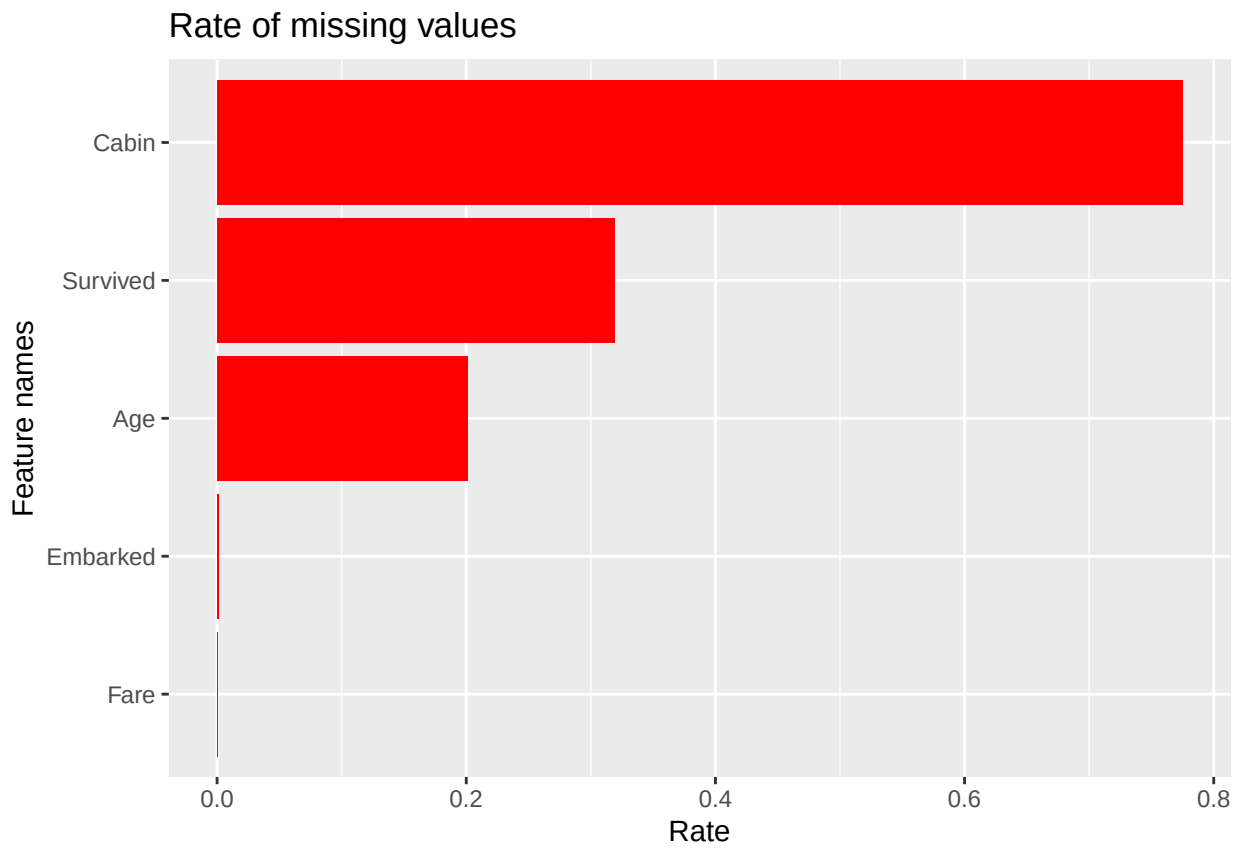
```
## Warning: `funs()` is deprecated as of dplyr 0.8.0.
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once every 8 hours.
```

```
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
# wide to long
missing_values <- gather(missing_values, key="feature", value="missing_pct")
missing_values %>%
  # Aesthetic setting : reorder(-missing_pct) : 내림차순으로 정렬
  # reorder(정렬하고 싶은 변수, 연속형 데이터, 함수)
  ggplot(aes(x=reorder(feature,missing_pct),y=missing_pct)) +
  geom_bar(stat="identity",fill="red")+ # y축의 높이를 데이터의 값으로
  #theme_bw() +
  coord_flip() + # 축 변환
  labs(x = "Feature names", y = "Rate") +
  ggtitle("Rate of missing values")
```



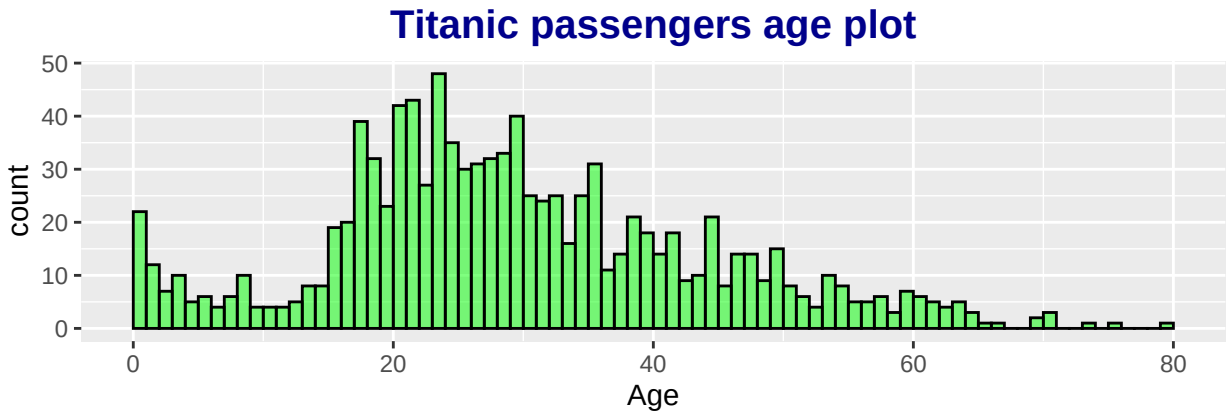

```
# https://rpubs.com/paul_0907/438825
# https://m.blog.naver.com/PostView.nhn?blogId=hwan0447&logNo=221325812408&proxyReferer=https:%2F%2Fwww

missing_values2 <- missing_values %>% filter(missing_pct>0)
missing_values2 %>%
  ggplot(aes(x=reorder(feature,missing_pct),y=missing_pct)) +
  geom_bar(stat="identity",fill="red")+ # y축의 높이를 데이터의 값으로
  #theme_bw() +
  coord_flip() + # 축 변환
  labs(x = "Feature names", y = "Rate") +
  ggtitle("Rate of missing values")
```

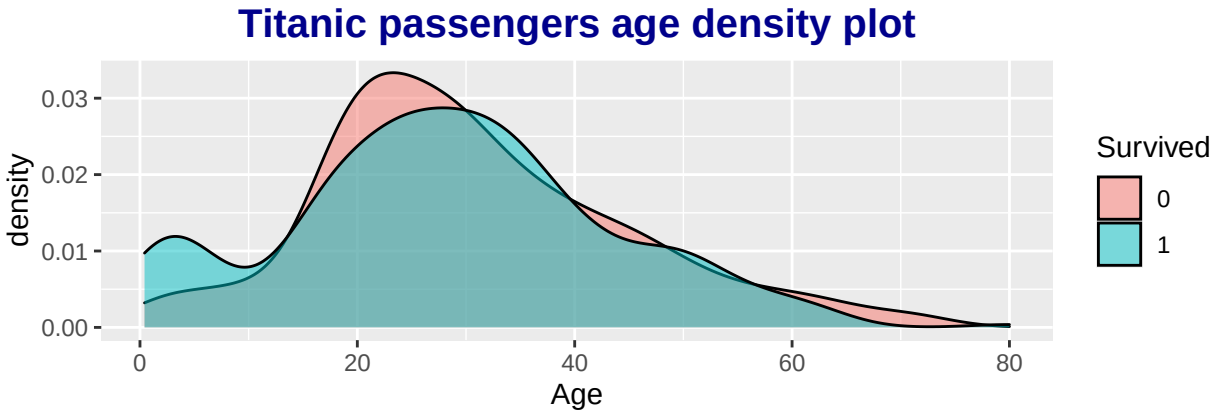



```
# Age
age.p1 <- full %>%
  ggplot(aes(Age)) +
  geom_histogram(breaks = seq(0, 80, by = 1), # 간격 설정
                 col    = "black",            # 막대 경계선 색깔
                 fill   = "green",            # 막대 내부 색깔
                 alpha  = .5) +               # 막대 투명도 = 50%
  ggtitle("Titanic passengers age plot") +
  theme(plot.title = element_text(face = "bold",     # 글씨체
                                  hjust = 0.5,       # Horizon(가로비율) = 0.5
                                  size = 15,
                                  color = "darkblue"))

age.p2 <- full %>%
  filter(!is.na(Survived)) %>%
  ggplot(aes(Age, fill = Survived)) +
  geom_density(alpha = .5) +
  ggtitle("Titanic passengers age density plot") +
  theme(plot.title = element_text(face = "bold",
                                  hjust = 0.5,
                                  size = 15,
                                  color = "darkblue"))
multi.layout = matrix(c(1,1,2,2), nrow=2, byrow=T) # 세로로 2개
multiplot(age.p1, age.p2, layout = multi.layout)
```

```
## Warning: Removed 263 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 177 rows containing non-finite values (stat_density).
```

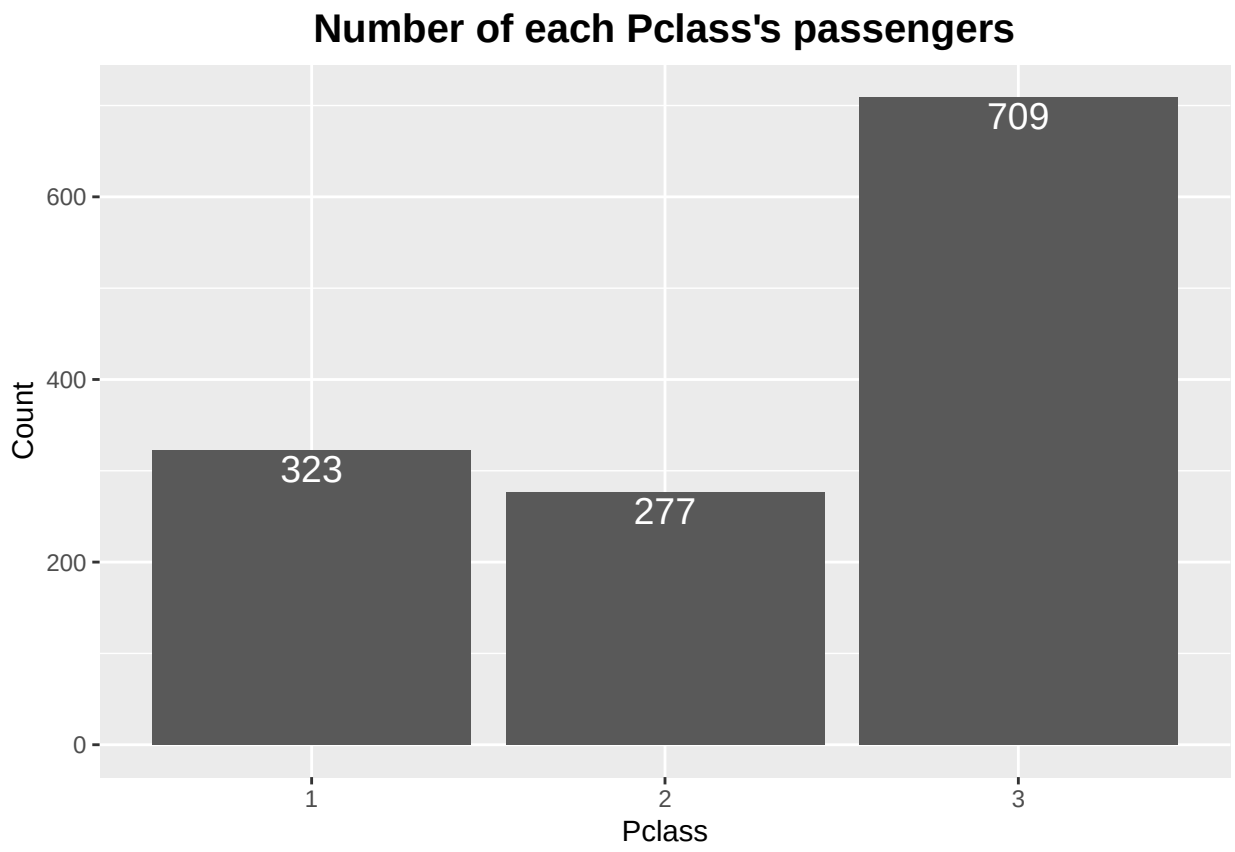





```r
# multi.layout = matrix(c(1,1,2,2), nrow=2, byrow=F) # 가로로 2개
# multiplot(age.p1, age.p2, layout = multi.layout)

# SibSp & Parch -> FamilySized
full <- full %>%
  # SibSp + Parch + 1(myself) => FamilySize
  mutate(FamilySize = .$SibSp + .$Parch + 1,
         FamilySized = case_when(FamilySize == 1 ~ "Single",
                                 FamilySize >= 2 & FamilySize < 5 ~ "Small",
                                 FamilySize >= 5 ~ "Big"),
         FamilySized = factor(FamilySized, levels = c("Single", "Small", "Big")))

# Pclass
full %>%
  group_by(Pclass) %>%
  summarize(N = n()) %>%
  ggplot(aes(Pclass, N)) +
  geom_col() +
  geom_text(aes(label = N),        # Plot의 y에 해당하는 N(빈도수)를 매핑
            size = 5,              # 글씨 크기
            vjust = 1.2,           # vertical(가로) 위치 설정
            colour = "white") +    # 글씨 색깔 : 흰색
          # color = "#FFFFFF")     # 글씨 색깔 : 흰색
  ggtitle("Number of each Pclass's passengers") +
```

```
  theme(plot.title = element_text(face = "bold",
                                  hjust = 0.5,
                                  size = 15)) +
  labs(x = "Pclass", y = "Count")
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

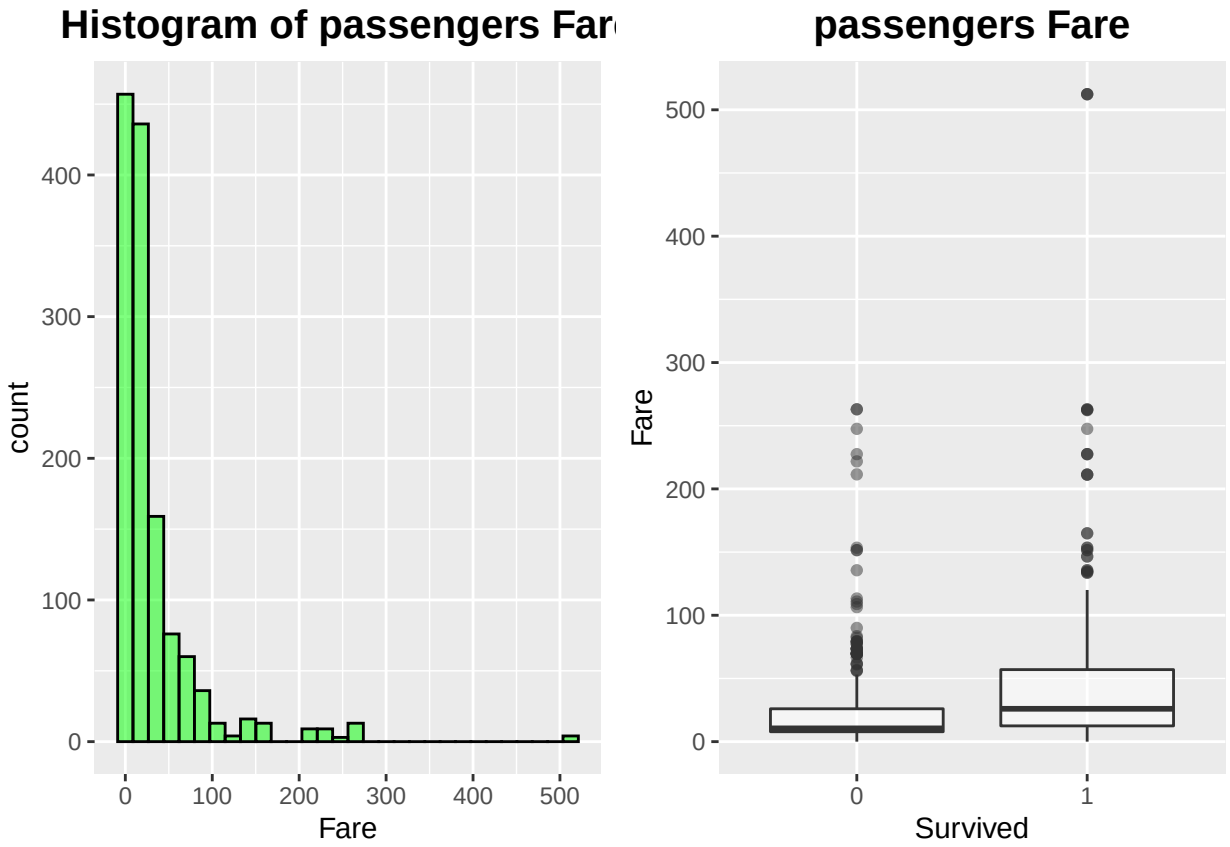



```
# Fare
Fare.p1 <- full %>%
  ggplot(aes(Fare)) +
  geom_histogram(col    = "black",
                 fill   = "green",
                 alpha  = .5) +
  ggtitle("Histogram of passengers Fare") +
  theme(plot.title = element_text(face = "bold",
                                  hjust = 0.5,
                                  size = 15))

Fare.p2 <- full %>%
  filter(!is.na(Survived)) %>%
  ggplot(aes(Survived, Fare)) +
  # 관측치를 회색점으로 찍되, 중복되는 부분은 퍼지게 그려줍니다.
  #geom_jitter(col = "gray") +
  geom_boxplot(alpha = .5) +
  ggtitle("passengers Fare") +
  theme(plot.title = element_text(face = "bold",
```

```
                                    hjust = 0.5,
                                    size = 15))
multi.layout = matrix(c(1,1,2,2), 2, 2, byrow=F) # 가로로 2개
multiplot(Fare.p1, Fare.p2, layout = multi.layout)
```

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

```
## Warning: Removed 1 rows containing non-finite values (stat_bin).
```

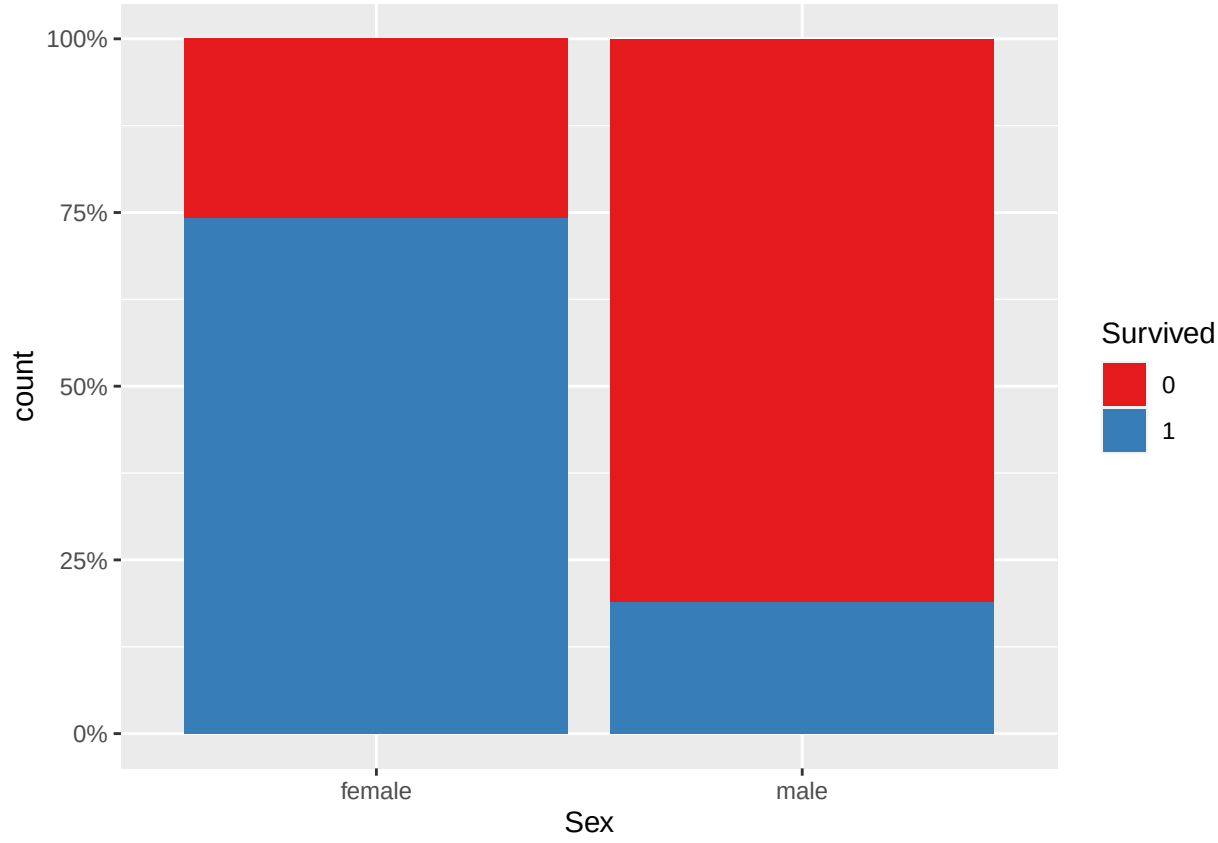



```
# Sex
sex.p1 <- full %>%
  group_by(Sex) %>%
  summarize(N = n()) %>%
  ggplot(aes(Sex, N)) +
  geom_col() +
  geom_text(aes(label = N),
            size = 5,
            vjust = 1.2,
            color = "#FFFFFF") +
  ggtitle("Bar plot of Sex") +
  labs(x = "Sex", y = "Count")
```

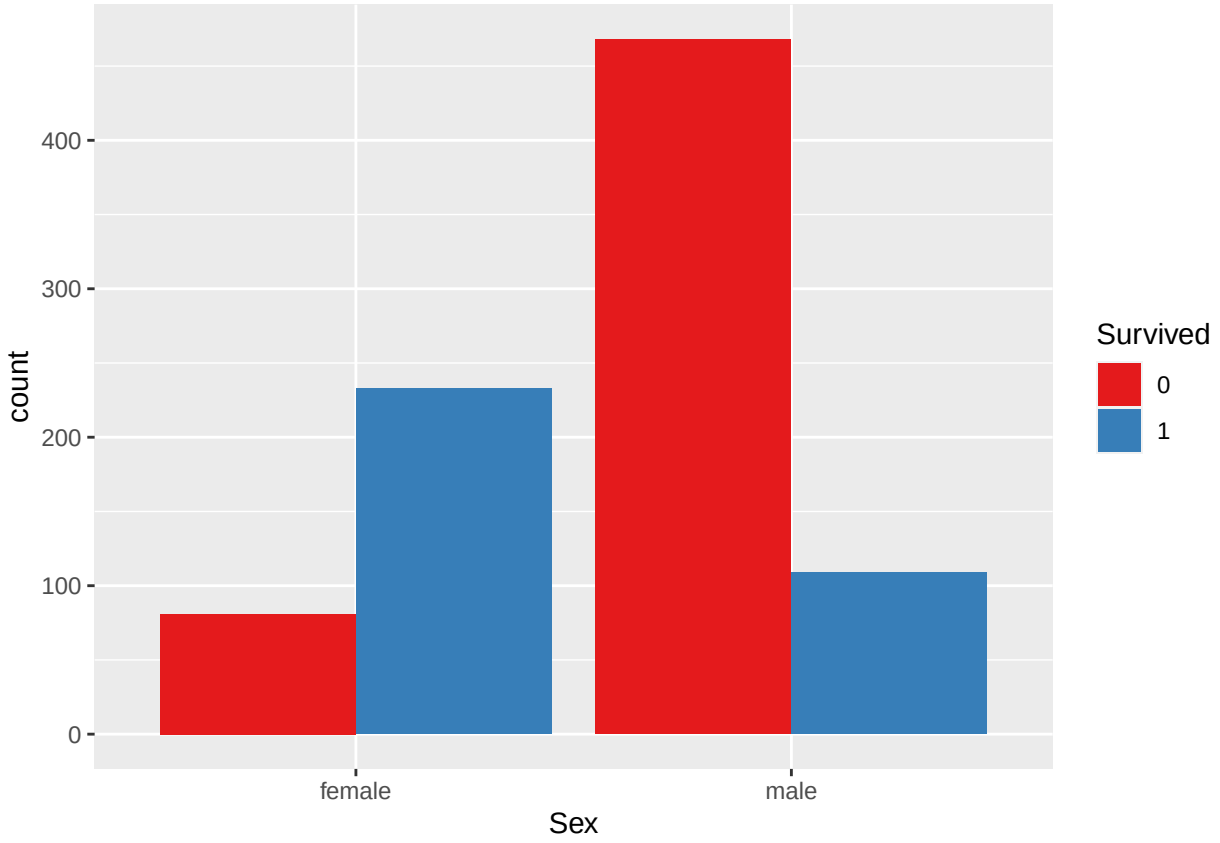
```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
sex.p2 <- full[1:891, ] %>%
  ggplot(aes(Sex, fill = Survived)) +
  geom_bar(position = "fill") +
```

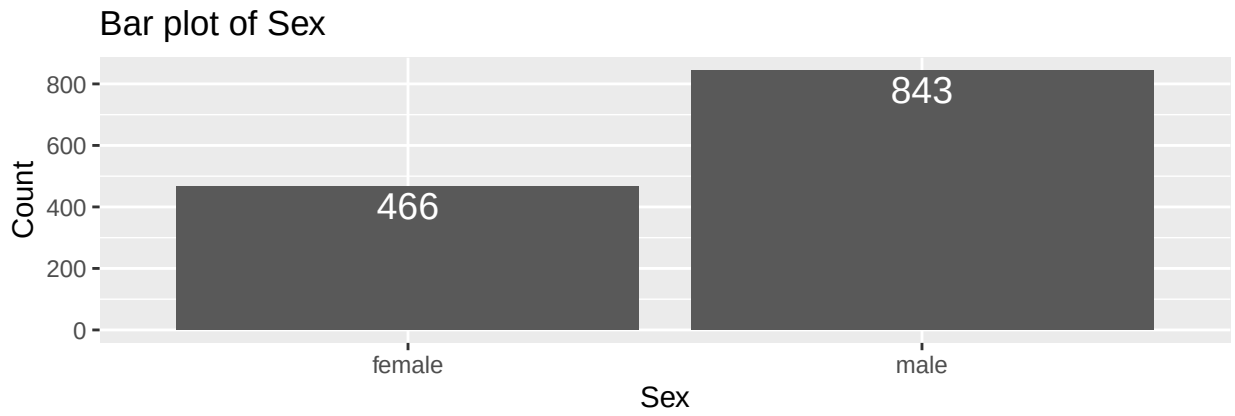
```
  scale_fill_brewer(palette = "Set1") +
  scale_y_continuous(labels = percent) +
  ggtitle("Survival Rate by Sex") +
  labs(x = "Sex", y = "Rate")

# position="fill" : 데이터의 종류를 비율로 표시 해주는 barplot
full[1:891,] %>%
  ggplot(aes(Sex, fill=Survived)) +
  geom_bar(position="fill") +
  scale_fill_brewer(palette = "Set1") +
  scale_y_continuous(labels = percent) # y축을 %로 나타냄
```

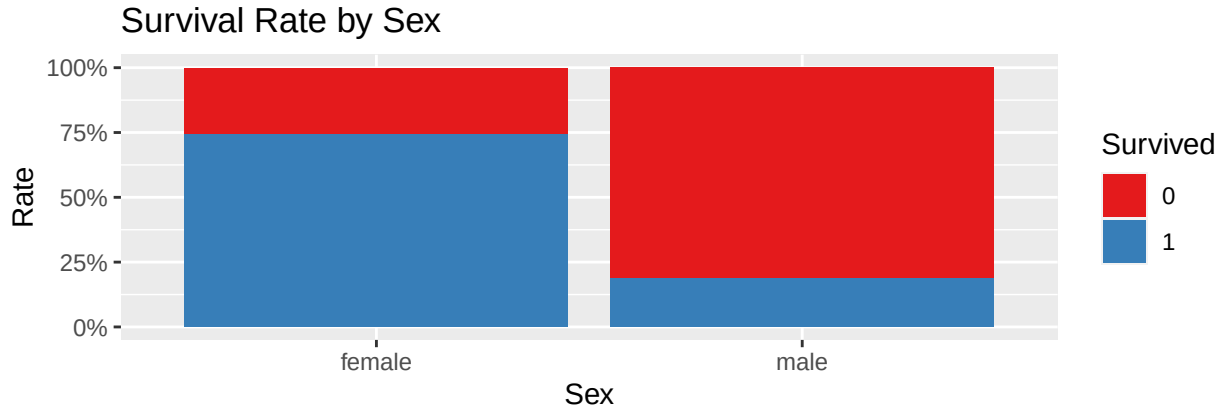



```
# position="dodge" : 데이터의 종류를 따로 표시 해주는 barplot
full[1:891,] %>%
  ggplot(aes(Sex, fill=Survived)) +
  geom_bar(position="dodge") +
  scale_fill_brewer(palette = "Set1") +
  scale_y_continuous() # y축을 %로 나타냄
```

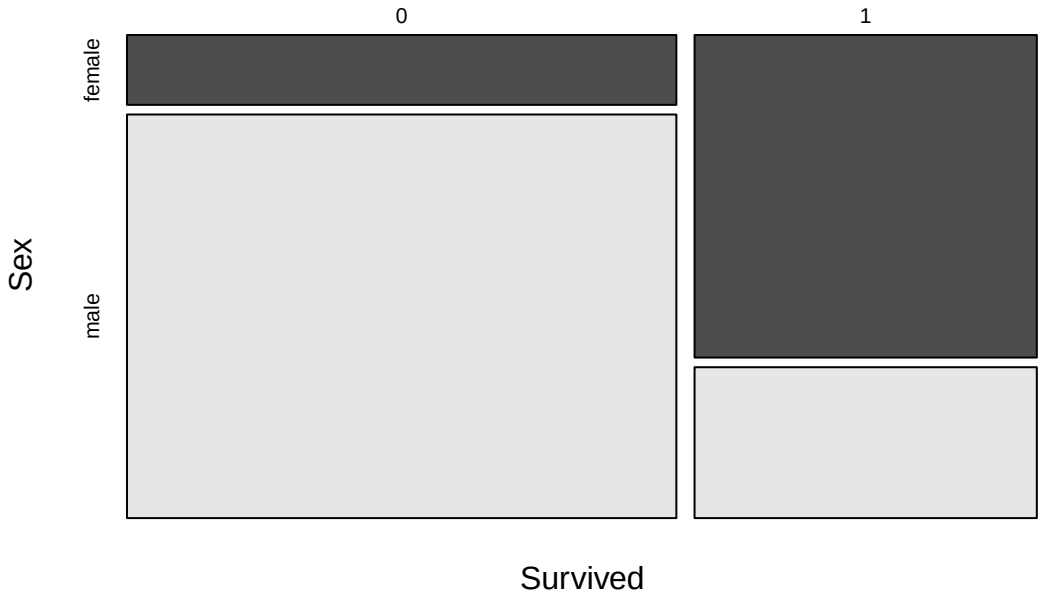



```
multi.layout = matrix(c(1,1,2,2), 2, 2, byrow=T)
multiplot(sex.p1, sex.p2, layout = multi.layout)
```






```r
# mosaicplot
mosaicplot(Survived ~ Sex, data=full[1:891,], col=T,
           main="Survival tate by passengers sex")
```




```
# Embarked
full$Embarked <- replace(full$Embarked, which(is.na(full$Embarked)), 'S')

# Title
full$Name %>% head
```

```
## [1] Braund, Mr. Owen Harris
## [2] Cumings, Mrs. John Bradley (Florence Briggs Thayer)
## [3] Heikkinen, Miss. Laina
## [4] Futrelle, Mrs. Jacques Heath (Lily May Peel)
## [5] Allen, Mr. William Henry
## [6] Moran, Mr. James
## 1307 Levels: Abbing, Mr. Anthony ... Zimmerman, Mr. Leo
```

```
Title <- gsub('(.*, )|(\\..*)', '', full$Name)
# 쉼표 전까지의 모든 문자,숫자,공백을 날리고 쉼표 후 한칸도 날린다
# 마침표를 찾아서(\\.) 그 뒤의 모든 문자,숫자,공백을 날린다.
# ^ : ~기호 뒤에 있는 글자로 시작하는 문장을 찾음
# . : 문자, 숫자, 공백을 가리지 않고 어떤 것이라도 매칭
# * : 무한번
# \\: 특수문자(^, $, ., ...)을 매칭
# $ : 문자열의 끝
# https://blog.naver.com/sw4r/221119461120
# https://statart.tistory.com/64
# https://statkclee.github.io/nlp2/regex-index.html
# Another way
Title <- gsub("^.*, (.*?)\\..*$", "\\1", full$Name)
```

```
# ( needs to be escaped
# \\(, . means everything,
# * means repeated 0 to n,
# ? means non greedy to remove not everything from the first to the last match.
full$Title <- Title
unique(Title)
```

```
##  [1] "Mr"           "Mrs"          "Miss"         "Master"       "Don"
##  [6] "Rev"          "Dr"           "Mme"          "Ms"           "Major"
## [11] "Lady"         "Sir"          "Mlle"         "Col"          "Capt"
## [16] "the Countess" "Jonkheer"     "Dona"
```

```
table(Title)
```

```
## Title
##         Capt          Col          Don         Dona           Dr     Jonkheer
##            1            4            1            1            8            1
##         Lady        Major       Master         Miss         Mlle          Mme
##            1            2           61          260            2            1
##           Mr          Mrs           Ms          Rev          Sir the Countess
##          757          197            2            8            1            1
```

```
# 18 -> 5 범주화
full <- full %>%
  mutate(Title = ifelse(Title %in% c("Mlle", "Ms", "Lady", "Dona"), "Miss", Title),
         Title = ifelse(Title == "Mme", "Mrs", Title),
         Title = ifelse(Title %in% c("Capt", "Col", "Major", "Dr", "Rev", "Don",
                                     "Sir", "the Countess", "Jonkheer"), "Officer", Title),
         Title = factor(Title))
table(full$Title)
```

```
##
##  Master    Miss      Mr     Mrs Officer
##      61     266     757     198      27
```

```
# Generate new variables: Age.Group
fit_Age <- rpart(Age ~ Title + Pclass + SibSp + Parch, data=full)
full$Age[is.na(full$Age)] <- predict(fit_Age, newdata=full[is.na(full$Age),])
fit_Fare <- rpart(Fare ~ Title + Pclass + Embarked + Sex + Age, data=full)
full$Fare[is.na(full$Fare)] <- predict(fit_Fare, newdata=full[is.na(full$Fare),])

full <- full %>%
  mutate(Age.Group = case_when(Age < 13 ~ "Age.0012",
                               Age >= 13 & Age < 18 ~ "Age.1317",
                               Age >= 18 & Age < 60 ~ "Age.1859",
                               Age >= 60 ~ "Age.60inf"),
         Age.Group = factor(Age.Group))

colnames(full)
```

```
##  [1] "PassengerId" "Survived"    "Pclass"      "Name"        "Sex"
##  [6] "Age"         "SibSp"       "Parch"       "Ticket"      "Fare"
## [11] "Cabin"       "Embarked"    "FamilySize"  "FamilySized" "Title"
## [16] "Age.Group"
```

```
train <- full[1:891,]
test <- full[892:1309,]
```

```r
train <- train %>%
  select("Pclass", "Sex", "Embarked", "FamilySized", "Fare",
         "Age.Group", "Title", "Survived")

Id <- test$PassengerId
test <- test %>%
  select("Pclass", "Sex", "Embarked", "FamilySized", "Fare",
         "Age.Group", "Title", "Survived")

set.seed(123)
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

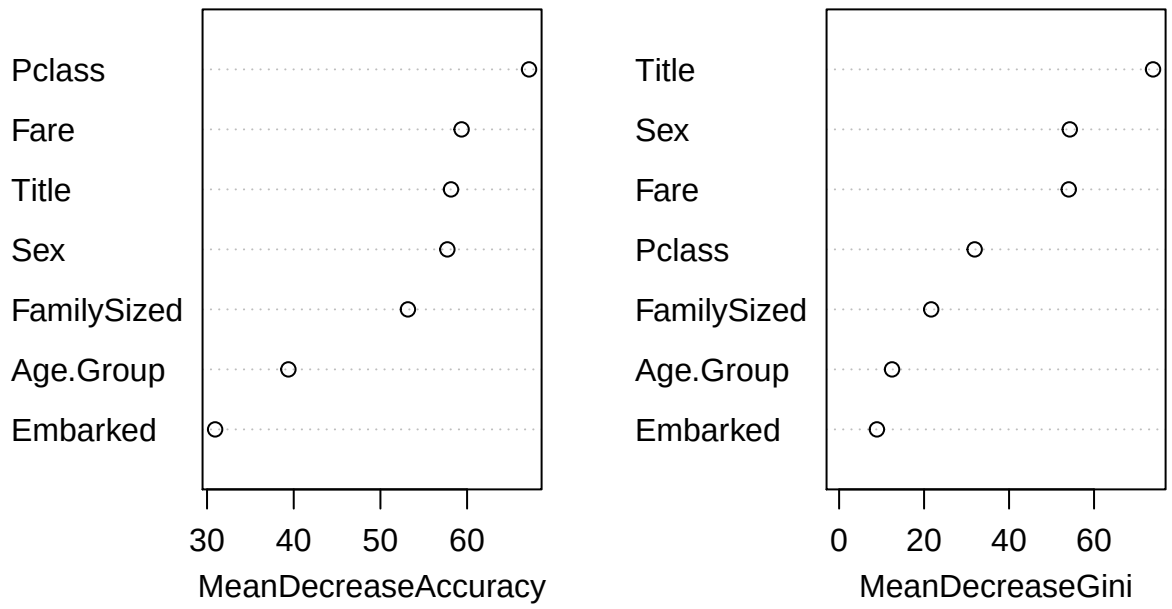
```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```r
titanic.rf <- randomForest(Survived ~ ., data = train, importance = T, ntree = 2000)
importance(titanic.rf)
```

```
##                     0        1 MeanDecreaseAccuracy MeanDecreaseGini
## Pclass      40.295219 55.30305             67.14490        31.904058
## Sex         53.194568 36.87923             57.72365        54.290373
## Embarked     9.214956 29.65093             30.93200         8.896919
## FamilySized 37.135771 32.72393             53.17215        21.672987
## Fare        34.995192 46.84666             59.36427        54.050141
## Age.Group   25.353259 36.03375             39.38689        12.456783
## Title       50.250622 42.95881             58.14402        73.887003
```

```r
varImpPlot(titanic.rf)
```




```r
pred.rf <- predict(object = titanic.rf, newdata = test, type = "class")

submit <- data.frame(PassengerId = Id, Survived = pred.rf)
write.csv(submit, file = './titanic_submit.csv', row.names = F)
```