R_ML_PROJECT02_Titanic

00 라이브러리

```
library(Amelia)
```

```
## Loading required package: Rcpp
```

```
## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.7.5, built: 2018-05-07)
## ## Copyright (C) 2005-2019 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##
```

```
library(ggplot2)
```

01 데이터 불러오기

```
train <-  read.csv("./R_Data/titanic_train.csv", stringsAsFactors=F, na.strings = c(
"", "NA"))
test <- read.csv("./R_Data/titanic_test.csv", stringsAsFactors=F, na.strings = c("",
"NA"))
# train <-  read.csv("./R_Data/titanic_train.csv", stringsAsFactors=F)
# test <- read.csv("./R_Data/titanic_test.csv", stringsAsFactors=F)
sub <- read.csv("./R_Data/sample_submission.csv", stringsAsFactors=F)
dim(train); dim(test); dim(sub)
```

```
## [1] 891  12
```

```
## [1] 418  11
```

```
## [1] 418   2
```

02 데이터 탐색

- 학습 데이터에 Survived 있음.
- 테스트 데이터에 Survived가 없음.

```
names(train)
```

```
##  [1] "PassengerId" "Survived"    "Pclass"      "Name"        "Sex"
##  [6] "Age"         "SibSp"       "Parch"       "Ticket"      "Fare"
## [11] "Cabin"       "Embarked"
```

```
cat("₩n")
```

```
names(test)
```

```
##  [1] "PassengerId" "Pclass"      "Name"        "Sex"         "Age"
##  [6] "SibSp"       "Parch"       "Ticket"      "Fare"        "Cabin"
## [11] "Embarked"
```

```
cat("₩n")
```

```
names(sub)
```

```
## [1] "PassengerId" "Survived"
```

```
cat("₩n")
```

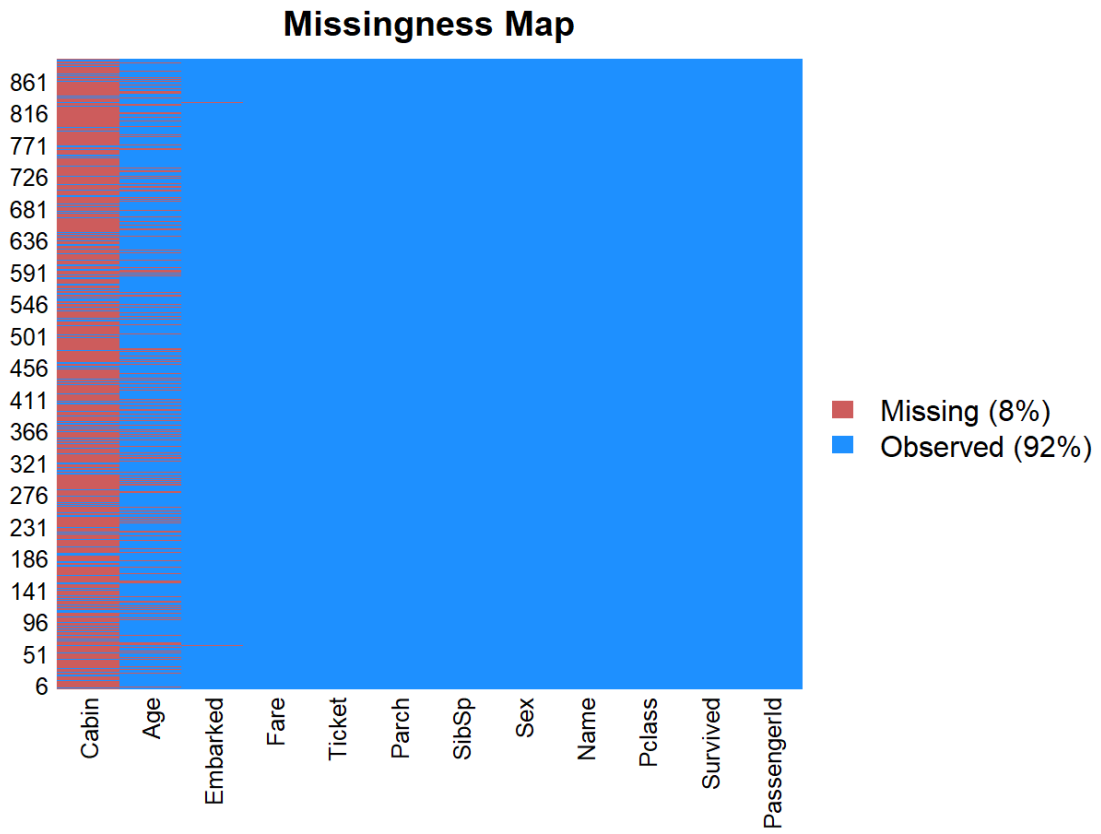
```
str(train)
```

```
## 'data.frame':    891 obs. of  12 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
##  $ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ Name       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)" "Heikkinen, Miss. Laina" "Futrelle, Mrs. Jacques Heath (Lily May Peel)" ...
##  $ Sex        : chr  "male" "female" "female" "female" ...
##  $ Age        : num  22 38 26 35 35 NA 54 2 27 14 ...
##  $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Ticket     : chr  "A/5 21171" "PC 17599" "STON/02. 3101282" "113803" ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : chr  NA "C85" NA "C123" ...
##  $ Embarked   : chr  "S" "C" "S" "S" ...
```

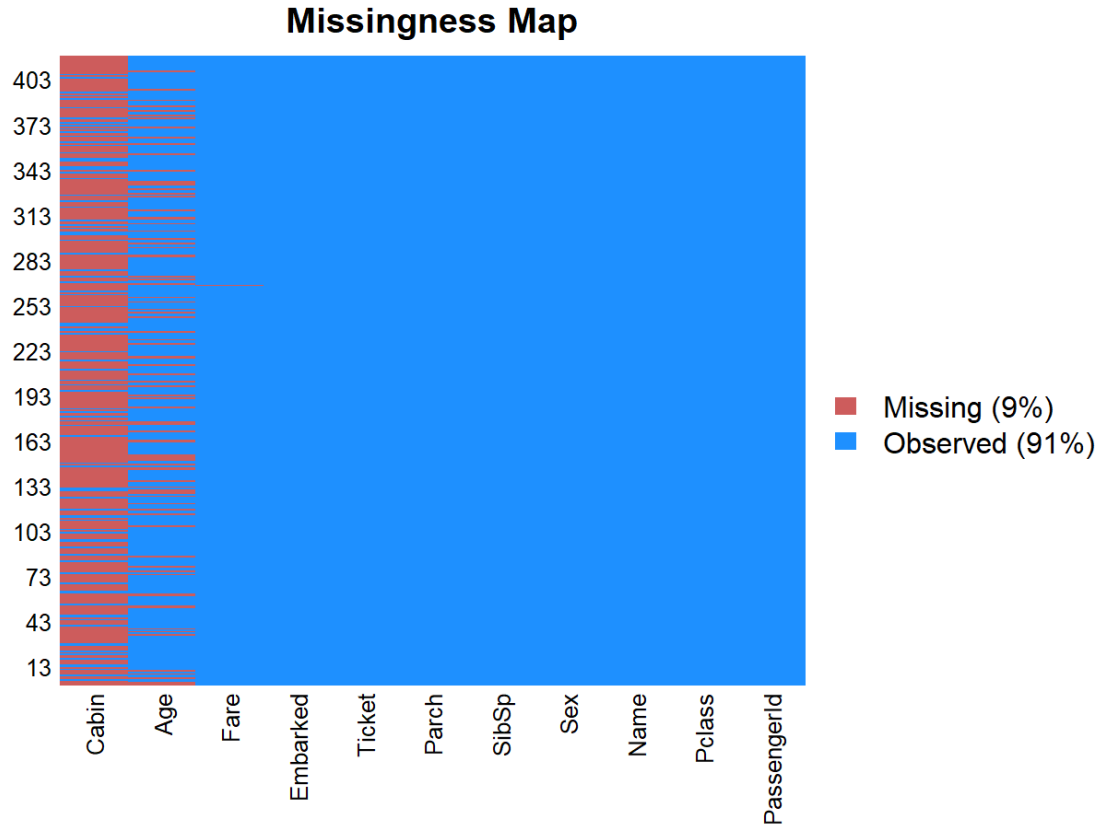
03 데이터 결측치 확인 및 처리

- train에 Age
- test에 Age와 Fare

```
# library(Amelia)
missmap(train)
```




```
missmap(test)
```



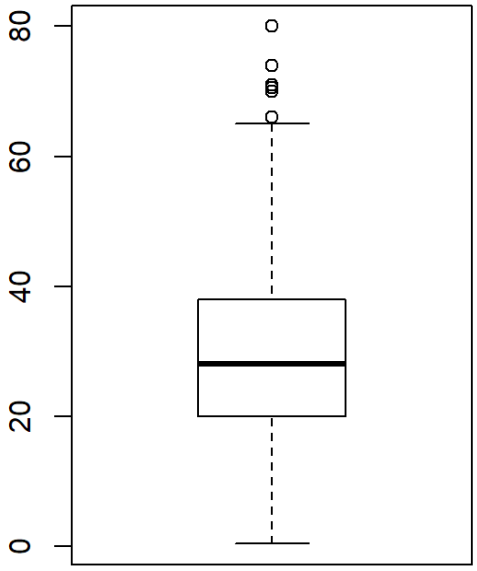

```
colSums(is.na(train))
```

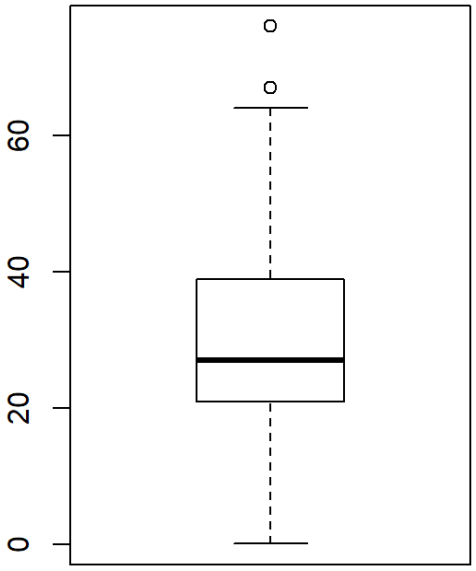
```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0           0           0           0           0         177
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0           0         687           2
```

```
colSums(is.na(test))
```

```
## PassengerId      Pclass        Name         Sex         Age       SibSp
##           0           0           0           0          86           0
##       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           1         327           0
```

```
par(mfrow=c(1,2))
boxplot(train$Age)
boxplot(test$Age)
```





결측치 처리

- 나이(Age)는 중앙값으로 대체
- 정박항(Embarked)는 많이 나온 값으로 대체

```
quantile(train$Age, na.r=T); quantile(test$Age, na.r=T)
```

```
##     0%    25%    50%    75%   100%
##  0.420 20.125 28.000 38.000 80.000
```

```
##    0%   25%   50%   75%  100%
##  0.17 21.00 27.00 39.00 76.00
```

나이(Age) 결측치 처리

- train(학습 데이터)는 177개 처리

```
## 학습용 데이터 처리
nrow( train[ is.na(train$Age), ] )
```

```
## [1] 177
```

```
train[ is.na(train$Age), 'Age'] = median(train$Age, na.rm=T)

## 테스트용 데이터 처리
nrow( test[ is.na(test$Age), ] )
```

```
## [1] 86
```

```
test[ is.na(test$Age), 'Age'] = median(test$Age, na.rm=T)

## 확인
nrow( train[ is.na(train$Age), ] ); nrow( test[ is.na(test$Age), ] )
```

```
## [1] 0
```

```
## [1] 0
```

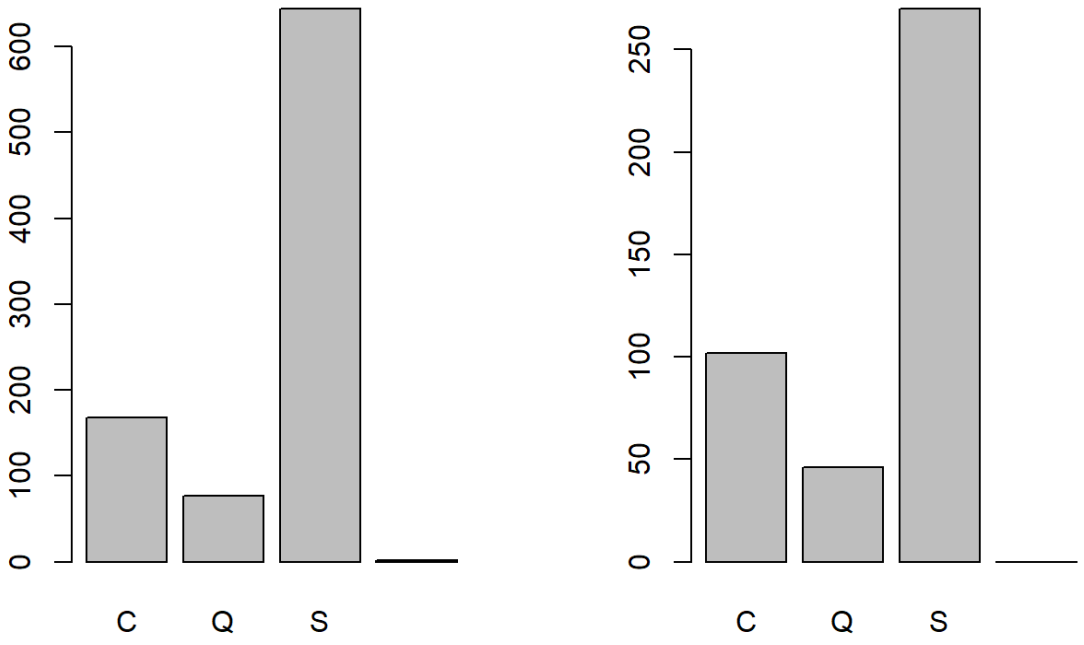
정박항(Embarked) 결측치 처리

```
cnt_tr <- table(train$Embarked, useNA='always')
cnt_test <- table(test$Embarked, useNA='always')
cnt_tr; cnt_test
```

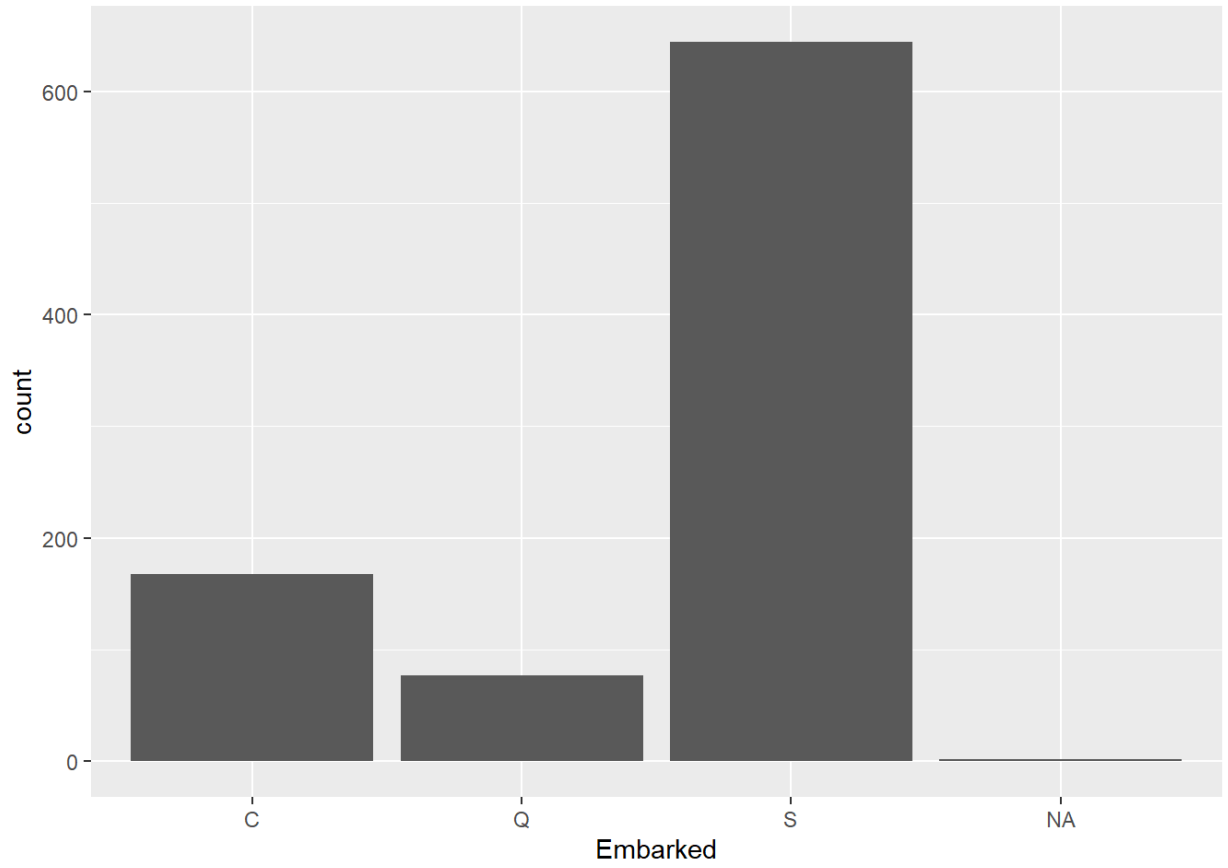
```
##
##    C    Q    S <NA>
##  168   77  644    2
```

```
##
##    C    Q    S <NA>
##  102   46  270    0
```

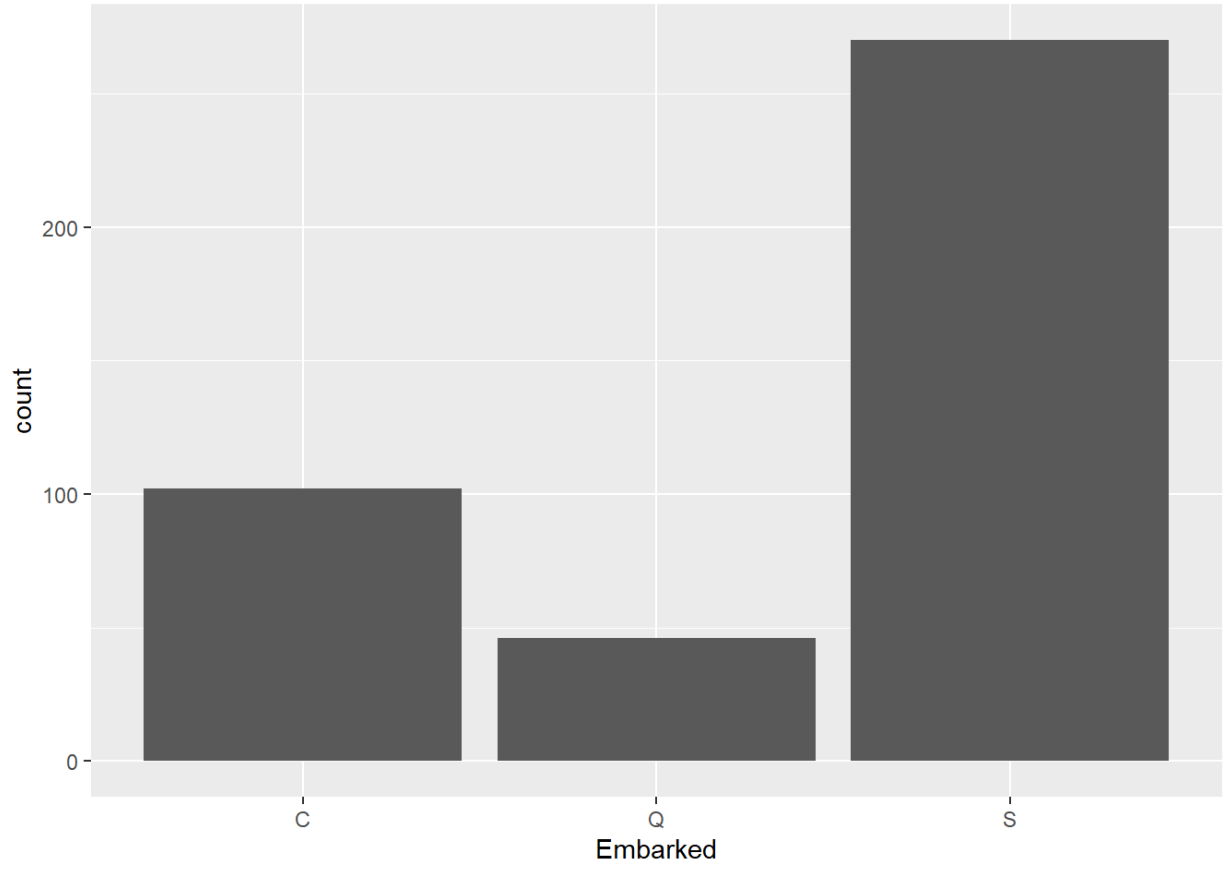
```
par(mfrow=c(1,2))
barplot(cnt_tr)
barplot(cnt_test)
```



```
ggplot(data=train, aes(x=Embarked)) + geom_bar()
```




```
ggplot(data=test, aes(x=Embarked)) + geom_bar()
```



```
## 학습용 데이터 처리
nrow( train[ is.na(train$Embarked), ] )
```

```
## [1] 2
```

```
train[ is.na(train$Age), 'Age'] = median(train$Age, na.rm=T)

## 테스트용 데이터 처리
nrow( test[ is.na(test$Embarked), ] )
```

```
## [1] 0
```

```
test[ is.na(test$Age), 'Age'] = median(test$Age, na.rm=T)

train[ is.na(train$Embarked), 'Embarked'] = 'S'
nrow( train[ is.na(train$Embarked), ] )
```

```
## [1] 0
```

데이터 확인

```
colSums(is.na(train))
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0           0           0           0           0           0
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0           0         687           0
```

```
colSums(is.na(test))
```

```
## PassengerId      Pclass        Name         Sex         Age       SibSp
##           0           0           0           0           0           0
##       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           1         327           0
```

04 데이터 모델 만들기

```
m <- glm(Survived ~ Pclass + Age + SibSp, family=binomial, data=train)
summary(m)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Age + SibSp, family = binomial,
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0261  -0.8445  -0.6891   1.0182   2.2946
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.206375   0.369876   8.669  < 2e-16 ***
## Pclass      -1.086108   0.100157 -10.844  < 2e-16 ***
## Age         -0.039779   0.006596  -6.031 1.63e-09 ***
## SibSp       -0.125663   0.078222  -1.606    0.108
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1186.7  on 890  degrees of freedom
## Residual deviance: 1044.8  on 887  degrees of freedom
## AIC: 1052.8
##
## Number of Fisher Scoring iterations: 4
```

예측

```
pred <- predict(m, newdata=test, type = "response")
pred[0:15]
```

```
##         1         2         3         4         5         6         7
## 0.1939829 0.1143317 0.1927629 0.2449018 0.2586960 0.3523193 0.2235099
##         8         9        10        11        12        13        14
## 0.4685999 0.3169169 0.2425700 0.2449018 0.5720990 0.7464335 0.1683175
##        15
## 0.5312039
```

```
pred <- as.integer(pred > 0.5)
pred[0:15]
```

```
##  [1] 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1
```

제출

```
sub[ ,'Survived'] = pred
sub[0:15,]
```

```
##    PassengerId Survived
## 1          892        0
## 2          893        0
## 3          894        0
## 4          895        0
## 5          896        0
## 6          897        0
## 7          898        0
## 8          899        0
## 9          900        0
## 10         901        0
## 11         902        0
## 12         903        1
## 13         904        1
## 14         905        0
## 15         906        1
```

```
getwd()
```

```
## [1] "C:/Users/WITHJS/Documents/GitHub/RBasic"
```

```
write.csv(sub, file="firstSub.csv", row.names = F)
list.files(path=".", pattern=NULL)
```

```
##  [1] "df_score.csv"
##  [2] "df_score.rda"
##  [3] "firstSub.csv"
##  [4] "img"
##  [5] "pdf"
##  [6] "R_Data"
##  [7] "R_STAT_ANALYSIS"
##  [8] "RBasic_Source"
##  [9] "README.md"
## [10] "RLevelUp_Source"
## [11] "RProject_practice_withdoit.ipynb"
## [12] "RProject01A_dplyr_withdoit_v11.ipynb"
## [13] "RProject01B_dplyr_ggplot_withdoit.ipynb"
## [14] "RProject01C_dplyr_ggplot_withdoit.ipynb"
## [15] "RProject02A_Titanic.html"
## [16] "RProject02A_Titanic.rmd"
## [17] "RProject02A_Titanic_files"
```