Final

2023-07-02

Required Packages

```r
library(lattice)
library(caret)
```

```
## Loading required package: ggplot2
```

```r
library(pscl)
```

```
## Classes and Methods for R developed in the
## Political Science Computational Laboratory
## Department of Political Science
## Stanford University
## Simon Jackman
## hurdle and zeroinfl functions by Achim Zeileis
```

```r
library(zoo)
```

```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
library(lmtest)
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(readr)
```

Import both test and train data

```r
train <- read_csv("Downloads/titanic/train.csv")
```

```
## Rows: 891 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (5): Name, Sex, Ticket, Cabin, Embarked
## dbl (7): PassengerId, Survived, Pclass, Age, SibSp, Parch, Fare
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
test <- read_csv("Downloads/titanic/test.csv")
```

```
## Rows: 418 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (5): Name, Sex, Ticket, Cabin, Embarked
## dbl (6): PassengerId, Pclass, Age, SibSp, Parch, Fare
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
str(test)
```

```
## spc_tbl_ [418 x 11] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ PassengerId: num [1:418] 892 893 894 895 896 897 898 899 900 901 ...
##  $ Pclass     : num [1:418] 3 3 2 3 3 3 3 2 3 3 ...
##  $ Name       : chr [1:418] "Kelly, Mr. James" "Wilkes, Mrs. James (Ellen Needs)" "Myles, Mr. Thomas
##  $ Sex        : chr [1:418] "male" "female" "male" "male" ...
##  $ Age        : num [1:418] 34.5 47 62 27 22 14 30 26 18 21 ...
##  $ SibSp      : num [1:418] 0 1 0 0 1 0 0 1 0 2 ...
##  $ Parch      : num [1:418] 0 0 0 0 1 0 0 1 0 0 ...
##  $ Ticket     : chr [1:418] "330911" "363272" "240276" "315154" ...
##  $ Fare       : num [1:418] 7.83 7 9.69 8.66 12.29 ...
##  $ Cabin      : chr [1:418] NA NA NA NA ...
##  $ Embarked   : chr [1:418] "Q" "S" "Q" "S" ...
##  - attr(*, "spec")=
##   .. 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()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
str(train)
```

```
## spc_tbl_ [891 x 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ PassengerId: num [1:891] 1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : num [1:891] 0 1 1 1 0 0 0 0 1 1 ...
##  $ Pclass     : num [1:891] 3 1 3 1 3 3 1 3 3 2 ...
##  $ Name       : chr [1:891] "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Th
##  $ Sex        : chr [1:891] "male" "female" "female" "female" ...
##  $ Age        : num [1:891] 22 38 26 35 35 NA 54 2 27 14 ...
##  $ SibSp      : num [1:891] 1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch      : num [1:891] 0 0 0 0 0 0 0 1 2 0 ...
##  $ Ticket     : chr [1:891] "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
##  $ Fare       : num [1:891] 7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : chr [1:891] NA "C85" NA "C123" ...
##  $ Embarked   : chr [1:891] "S" "C" "S" "S" ...
##  - 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()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

Data Cleaning

Find and replace missing data

```
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
```

```r
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
```

```r
train$Age[is.na(train$Age)] <- median(train$Age,na.rm = T)
test$Age[is.na(test$Age)] <- median(test$Age,na.rm = T)
test$Fare[is.na(test$Fare)] <- mean(test$Fare,na.rm = T)
table(train$Embarked)
```

```
##
##   C   Q   S
## 168  77 644
```

```r
train$Embarked[is.na(train$Embarked)] <- "S"
```

Drop column that has great number of missing values and does not provide important information to the problem

```r
new_train <- select(train, -c(PassengerId, Name, Cabin, Ticket))
new_test <- select(test, -c(PassengerId, Name, Cabin, Ticket))
```

Create new variables for family size

```r
new_train$family_size <- train$SibSp + train$Parch
new_test$family_size <- test$SibSp + test$Parch
```

Change variables that are categorical to factor

```r
factor_cols <- c("Pclass", "Embarked", "Sex" )
new_train[,factor_cols] <- lapply(new_train[,factor_cols],factor)
new_test[,factor_cols] <- lapply(new_test[,factor_cols], factor)
new_train$Survived <- as.factor(new_train$Survived)
str(new_train)
```

```
## tibble [891 x 9] (S3: tbl_df/tbl/data.frame)
##  $ 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 ...
##  $ Sex        : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
##  $ Age        : num [1:891] 22 38 26 35 35 28 54 2 27 14 ...
##  $ SibSp      : num [1:891] 1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch      : num [1:891] 0 0 0 0 0 0 0 1 2 0 ...
##  $ Fare       : num [1:891] 7.25 71.28 7.92 53.1 8.05 ...
##  $ Embarked   : Factor w/ 3 levels "C","Q","S": 3 1 3 3 3 2 3 3 3 1 ...
##  $ family_size: num [1:891] 1 1 0 1 0 0 0 4 2 1 ...
```

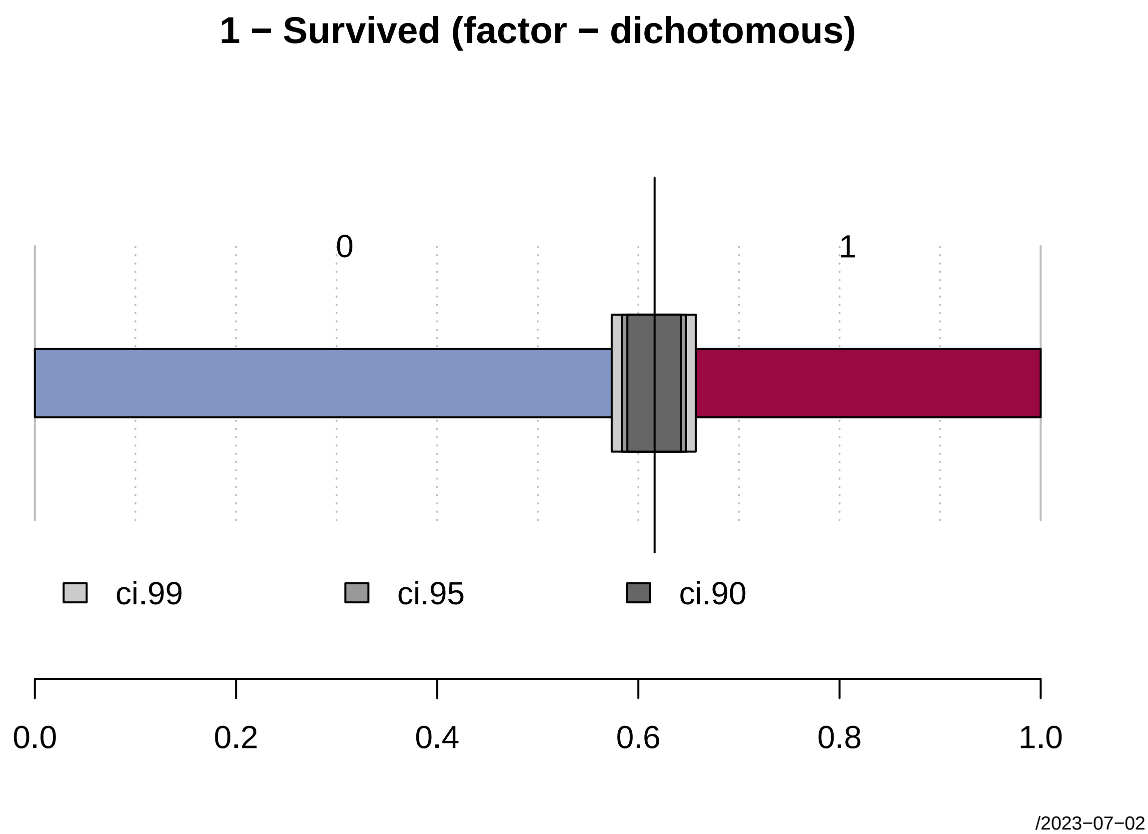
```r
str(new_test)
```

```
## tibble [418 x 8] (S3: tbl_df/tbl/data.frame)
##  $ Pclass     : Factor w/ 3 levels "1","2","3": 3 3 2 3 3 3 3 2 3 3 ...
##  $ Sex        : Factor w/ 2 levels "female","male": 2 1 2 2 1 2 1 2 1 2 ...
##  $ Age        : num [1:418] 34.5 47 62 27 22 14 30 26 18 21 ...
##  $ SibSp      : num [1:418] 0 1 0 0 1 0 0 1 0 2 ...
##  $ Parch      : num [1:418] 0 0 0 0 1 0 0 1 0 0 ...
##  $ Fare       : num [1:418] 7.83 7 9.69 8.66 12.29 ...
##  $ Embarked   : Factor w/ 3 levels "C","Q","S": 2 3 2 3 3 3 2 3 1 3 ...
##  $ family_size: num [1:418] 0 1 0 0 2 0 0 2 0 2 ...
```

Descriptive Statistic

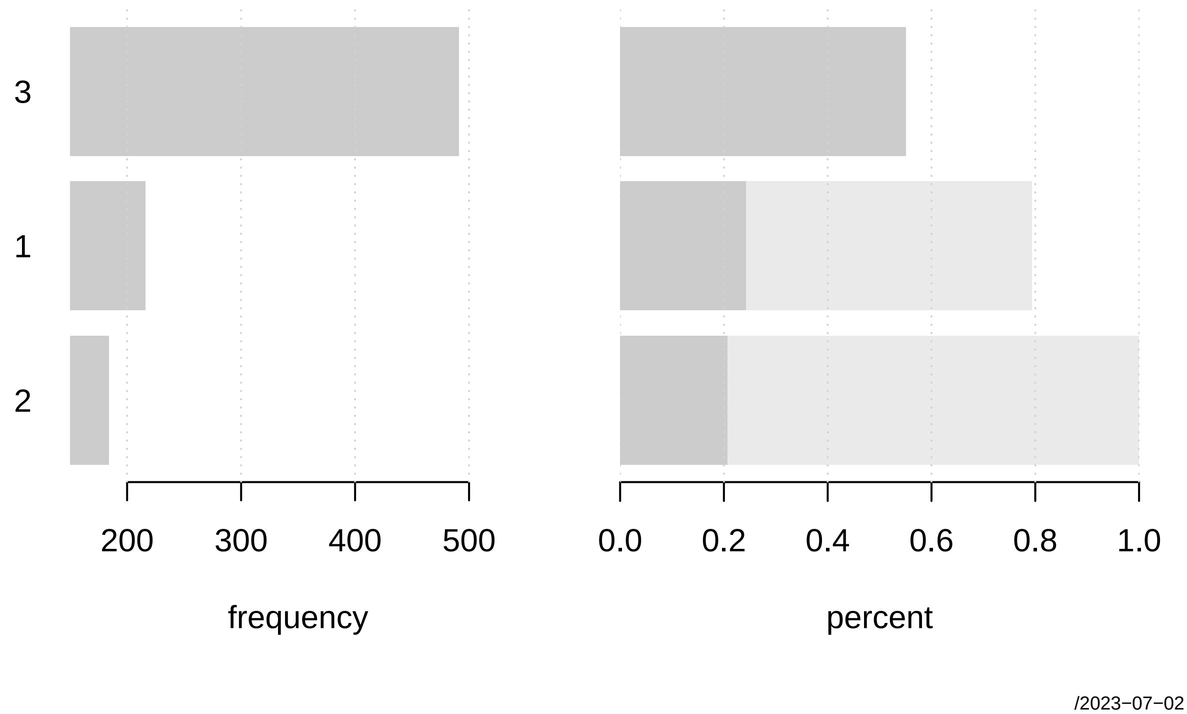
```r
DescTools::Desc(new_train)
```

```
## ------------------------------------------------------------------------------
## Describe new_train (tbl_df, tbl, data.frame):
##
## data frame:  891 obs. of  9 variables
##      891 complete cases (100.0%)
##
##   Nr  ColName      Class    NAs  Levels
##   1   Survived     factor   .    (2): 1-0, 2-1
##   2   Pclass       factor   .    (3): 1-1, 2-2, 3-3
##   3   Sex          factor   .    (2): 1-female, 2-male
##   4   Age          numeric  .
##   5   SibSp        numeric  .
##   6   Parch        numeric  .
##   7   Fare         numeric  .
##   8   Embarked     factor   .    (3): 1-C, 2-Q, 3-S
##   9   family_size  numeric  .
##
##
## ------------------------------------------------------------------------------
## 1 - Survived (factor - dichotomous)
##
##   length      n    NAs unique
##      891    891      0      2
##          100.0%   0.0%
##
##    freq   perc  lci.95  uci.95'
## 0   549  61.6%   58.4%   64.8%
## 1   342  38.4%   35.2%   41.6%
##
## ' 95%-CI (Wilson)
```

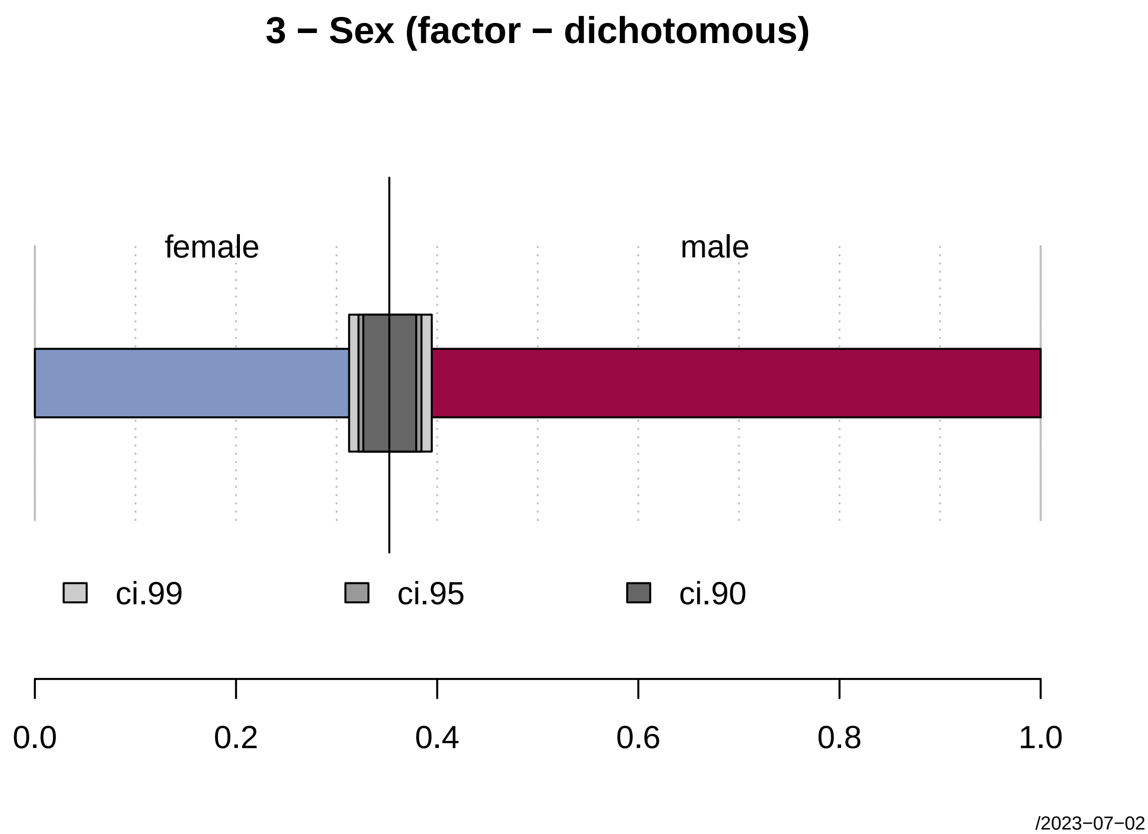



```
## ------------------------------------------------------------------------------
## 2 - Pclass (factor)
##
##   length      n    NAs unique levels  dupes
##      891    891      0      3      3      y
##          100.0%   0.0%
##
##    level  freq   perc  cumfreq  cumperc
## 1      3   491  55.1%      491    55.1%
## 2      1   216  24.2%      707    79.3%
## 3      2   184  20.7%      891   100.0%
```

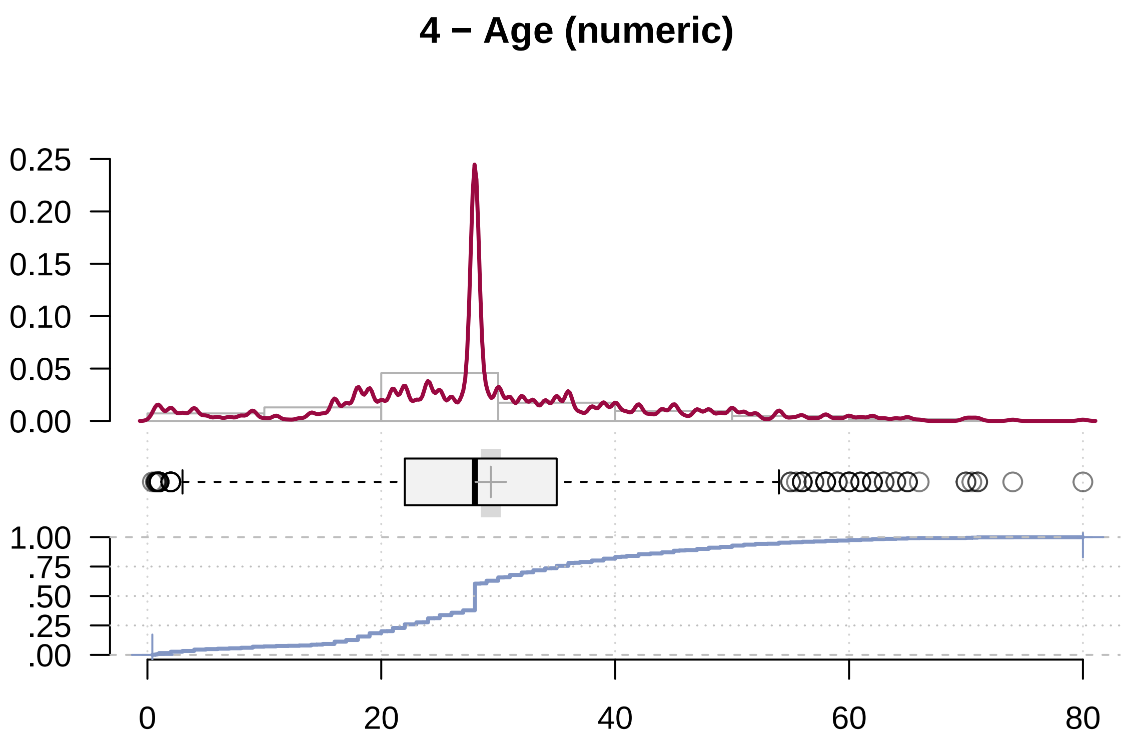
2 − Pclass (factor)



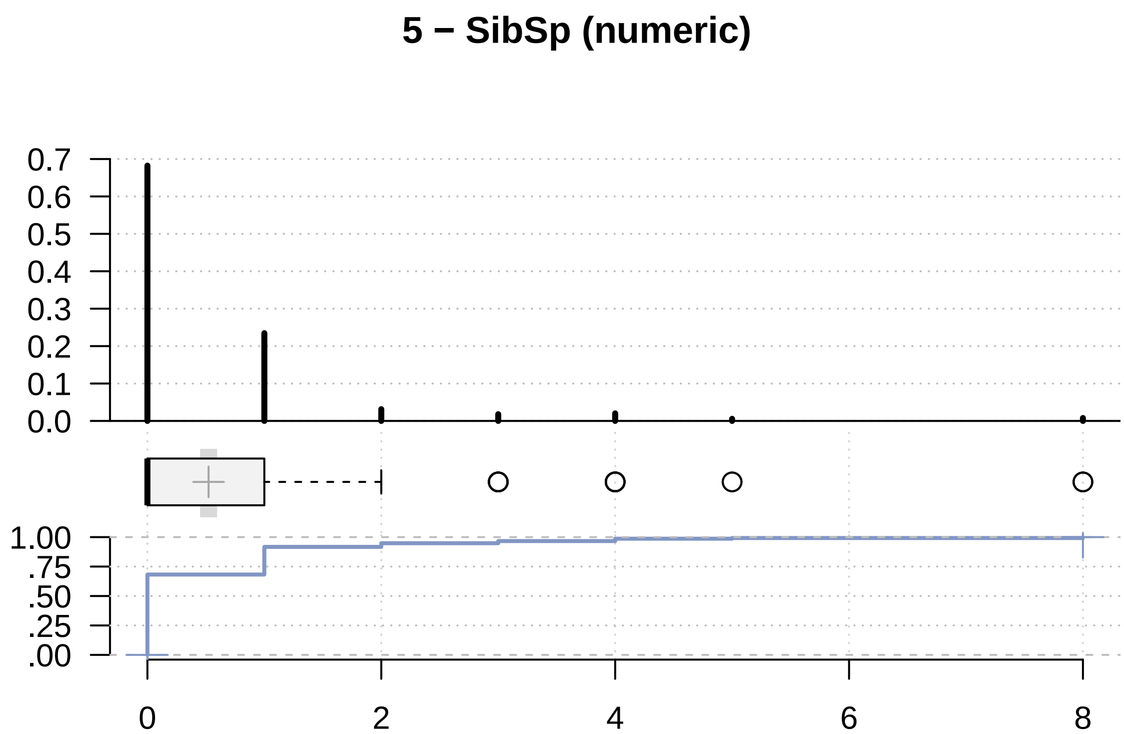

```
## ------------------------------------------------------------------------------
## 3 - Sex (factor - dichotomous)
##
##   length      n    NAs unique
##      891    891      0      2
##          100.0%   0.0%
##
##         freq   perc  lci.95  uci.95'
## female   314  35.2%   32.2%   38.4%
## male     577  64.8%   61.6%   67.8%
##
## ' 95%-CI (Wilson)
```




```
## ------------------------------------------------------------------------------
## 4 - Age (numeric)
##
##   length        n     NAs  unique      0s   mean  meanCI'
##      891      891       0      88       0  29.36    28.51
##            100.0%    0.0%           0.0%           30.22
##
##      .05      .10     .25  median     .75    .90      .95
##     6.00    16.00   22.00   28.00   35.00  47.00    54.00
##
##    range       sd   vcoef     mad     IQR   skew     kurt
##    79.58    13.02    0.44    8.90   13.00   0.51     0.97
##
## lowest : 0.42, 0.67, 0.75 (2), 0.83 (2), 0.92
## highest: 70.0 (2), 70.5, 71.0 (2), 74.0, 80.0
##
## heap(?): remarkable frequency (22.7%) for the mode(s) (= 28)
##
## ' 95%-CI (classic)
```

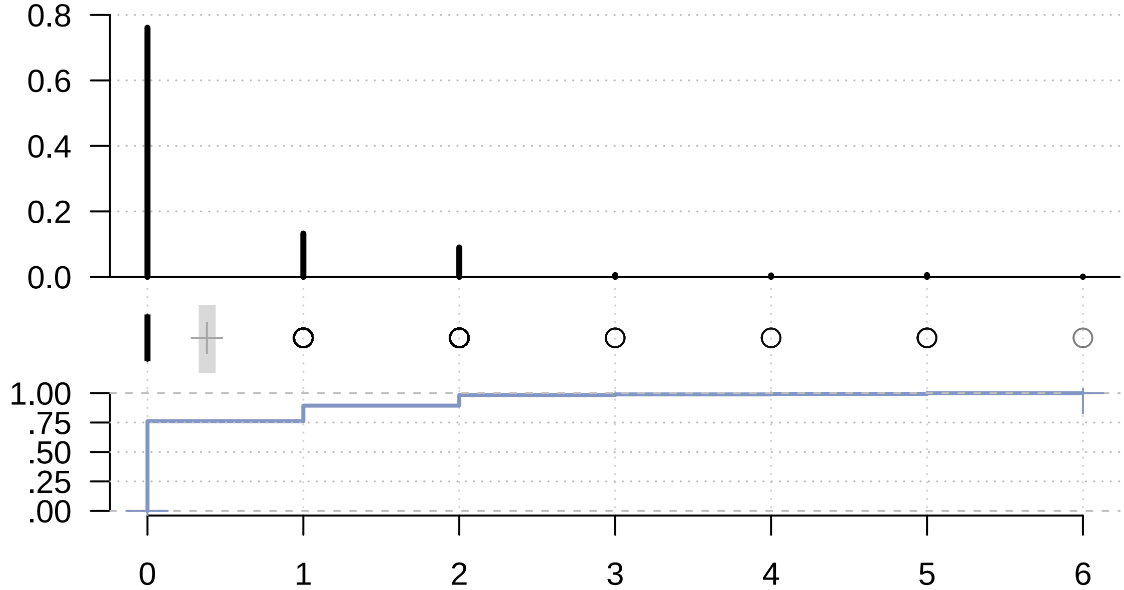



```
## ------------------------------------------------------------------------------
## 5 - SibSp (numeric)
##
##    length        n     NAs   unique      0s   mean   meanCI'
##       891      891       0        7     608   0.52     0.45
##            100.0%    0.0%           68.2%            0.60
##
##       .05      .10     .25   median     .75    .90      .95
##      0.00     0.00    0.00     0.00    1.00   1.00     3.00
##
##     range       sd   vcoef      mad     IQR   skew     kurt
##      8.00     1.10    2.11     0.00    1.00   3.68    17.73
##
##
##     value   freq    perc   cumfreq   cumperc
## 1       0    608   68.2%       608     68.2%
## 2       1    209   23.5%       817     91.7%
## 3       2     28    3.1%       845     94.8%
## 4       3     16    1.8%       861     96.6%
## 5       4     18    2.0%       879     98.7%
## 6       5      5    0.6%       884     99.2%
## 7       8      7    0.8%       891    100.0%
##
## ' 95%-CI (classic)
```

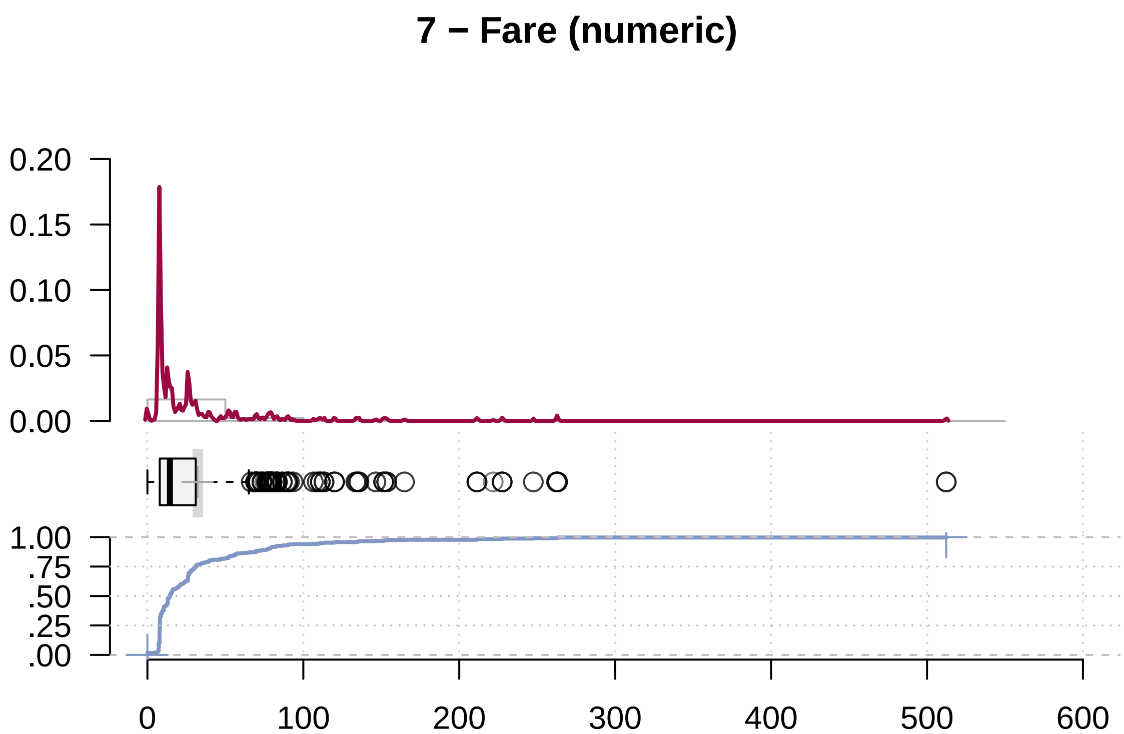



```
## ------------------------------------------------------------------------------
## 6 - Parch (numeric)
##
##    length        n     NAs  unique      0s  mean  meanCI'
##       891      891       0       7     678  0.38     0.33
##            100.0%    0.0%            76.1%           0.43
##
##       .05      .10     .25  median     .75   .90      .95
##      0.00     0.00    0.00    0.00    0.00  2.00     2.00
##
##     range       sd   vcoef     mad     IQR  skew     kurt
##      6.00     0.81    2.11    0.00    0.00  2.74     9.69
##
##
##     value  freq   perc  cumfreq  cumperc
## 1       0   678  76.1%      678    76.1%
## 2       1   118  13.2%      796    89.3%
## 3       2    80   9.0%      876    98.3%
## 4       3     5   0.6%      881    98.9%
## 5       4     4   0.4%      885    99.3%
## 6       5     5   0.6%      890    99.9%
## 7       6     1   0.1%      891   100.0%
##
## ' 95%-CI (classic)
```

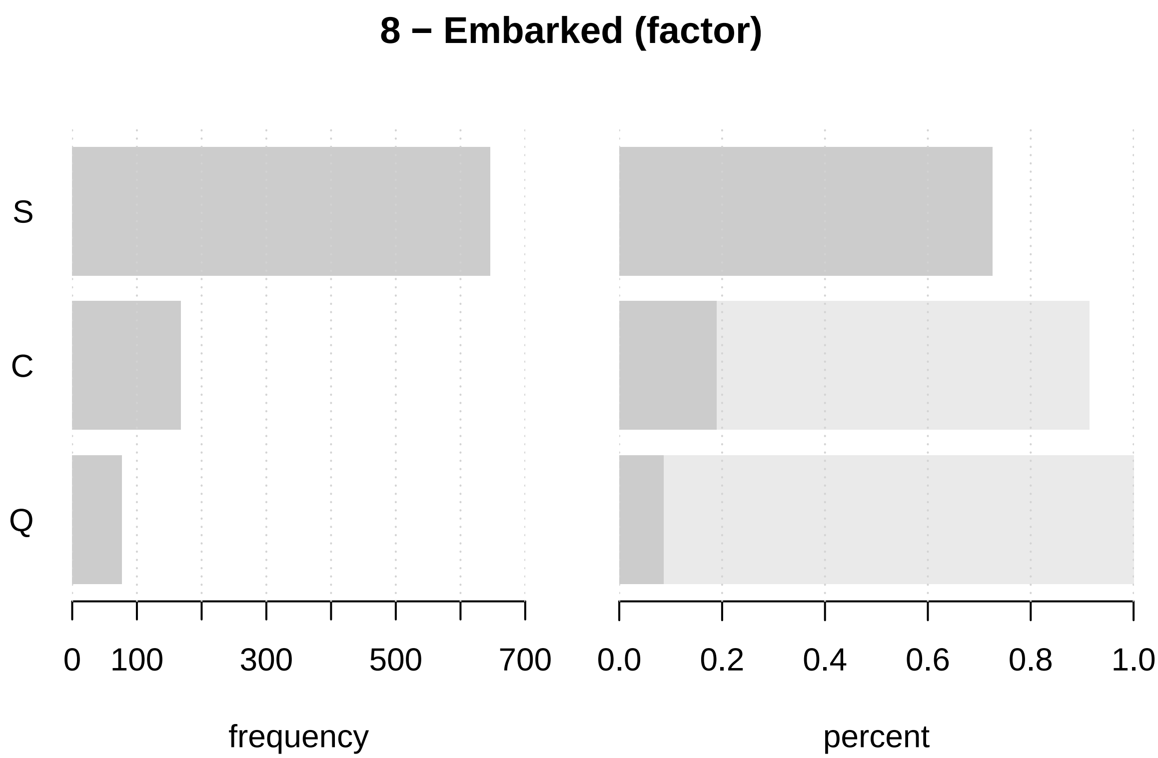
6 – Parch (numeric)



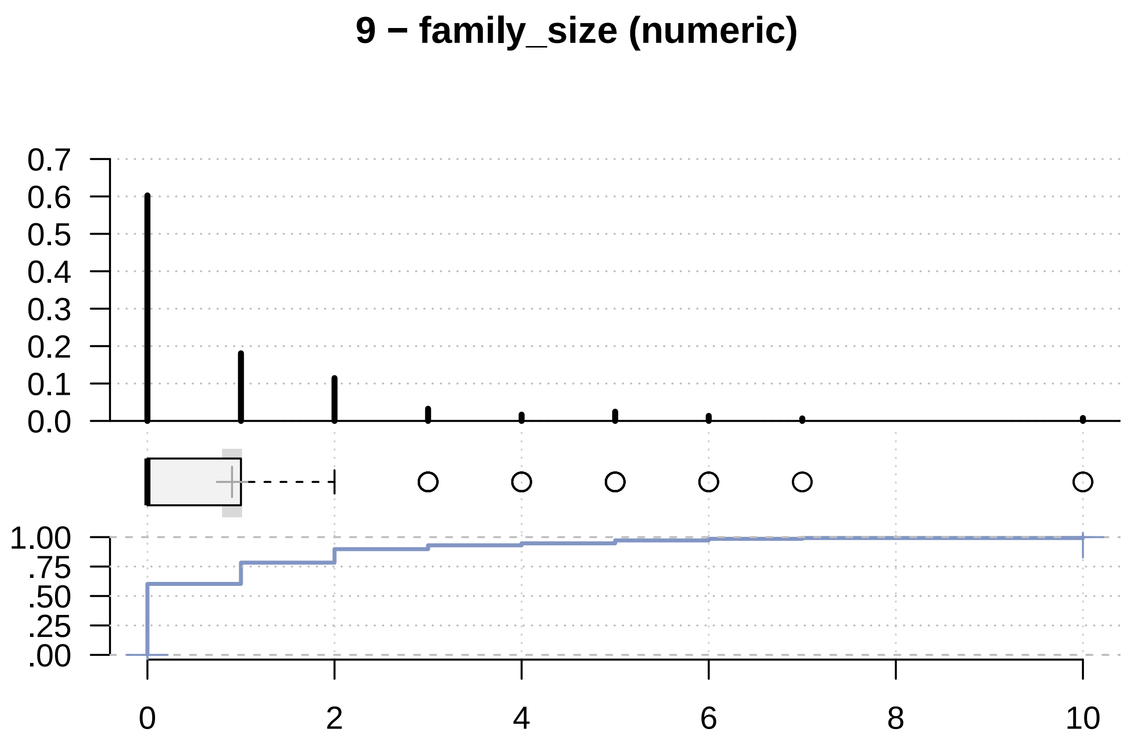

```
## ------------------------------------------------------------------------------
## 7 - Fare (numeric)
##
##      length         n       NAs    unique        0s      mean    meanCI'
##         891       891         0       248        15   32.2042    28.9368
##                 100.0%      0.0%                1.7%              35.4716
##
##         .05       .10       .25    median       .75       .90        .95
##      7.2250    7.5500    7.9104   14.4542   31.0000   77.9583   112.0791
##
##       range        sd     vcoef       mad       IQR      skew       kurt
##    512.3292   49.6934    1.5431   10.2362   23.0896    4.7712    33.1231
##
## lowest : 0.0 (15), 4.0125, 5.0, 6.2375, 6.4375
## highest: 227.525 (4), 247.5208 (2), 262.375 (2), 263.0 (4), 512.3292 (3)
##
## ' 95%-CI (classic)
```




```
## ------------------------------------------------------------------------------
## 8 - Embarked (factor)
##
##   length       n     NAs unique levels  dupes
##      891     891       0      3      3      y
##           100.0%    0.0%
##
##    level  freq   perc  cumfreq  cumperc
## 1      S   646  72.5%      646    72.5%
## 2      C   168  18.9%      814    91.4%
## 3      Q    77   8.6%      891   100.0%
```




```
## ------------------------------------------------------------------------------
## 9 - family_size (numeric)
##
##   length       n    NAs  unique     0s  mean  meanCI'
##      891     891      0       9    537  0.90     0.80
##           100.0%   0.0%          60.3%           1.01
##
##      .05     .10    .25  median    .75   .90      .95
##     0.00    0.00   0.00    0.00   1.00  3.00     5.00
##
##    range      sd  vcoef     mad    IQR  skew     kurt
##    10.00    1.61   1.78    0.00   1.00  2.72     9.07
##
##
##    value  freq   perc  cumfreq  cumperc
## 1      0   537  60.3%      537    60.3%
## 2      1   161  18.1%      698    78.3%
## 3      2   102  11.4%      800    89.8%
## 4      3    29   3.3%      829    93.0%
## 5      4    15   1.7%      844    94.7%
## 6      5    22   2.5%      866    97.2%
## 7      6    12   1.3%      878    98.5%
## 8      7     6   0.7%      884    99.2%
## 9     10     7   0.8%      891   100.0%
##
## ' 95%-CI (classic)
```

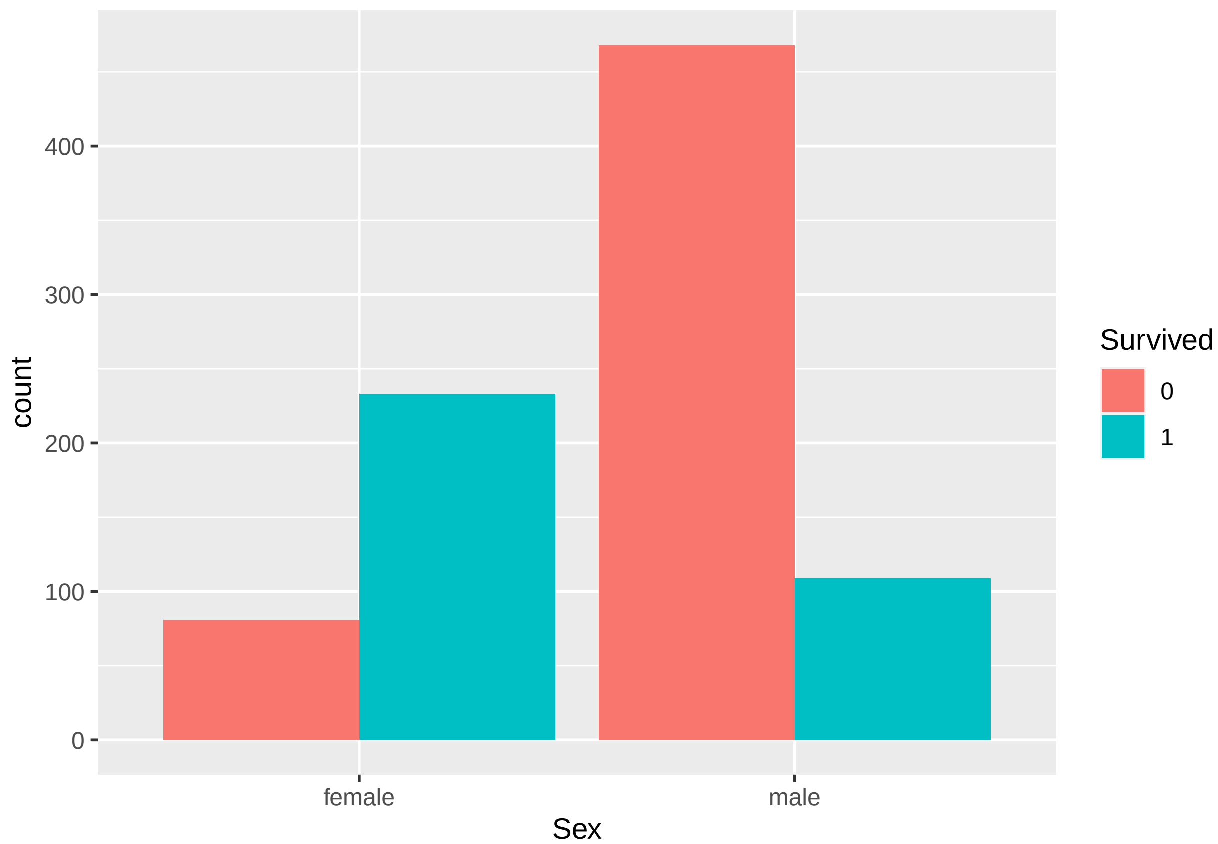



```r
summary(new_train)
```

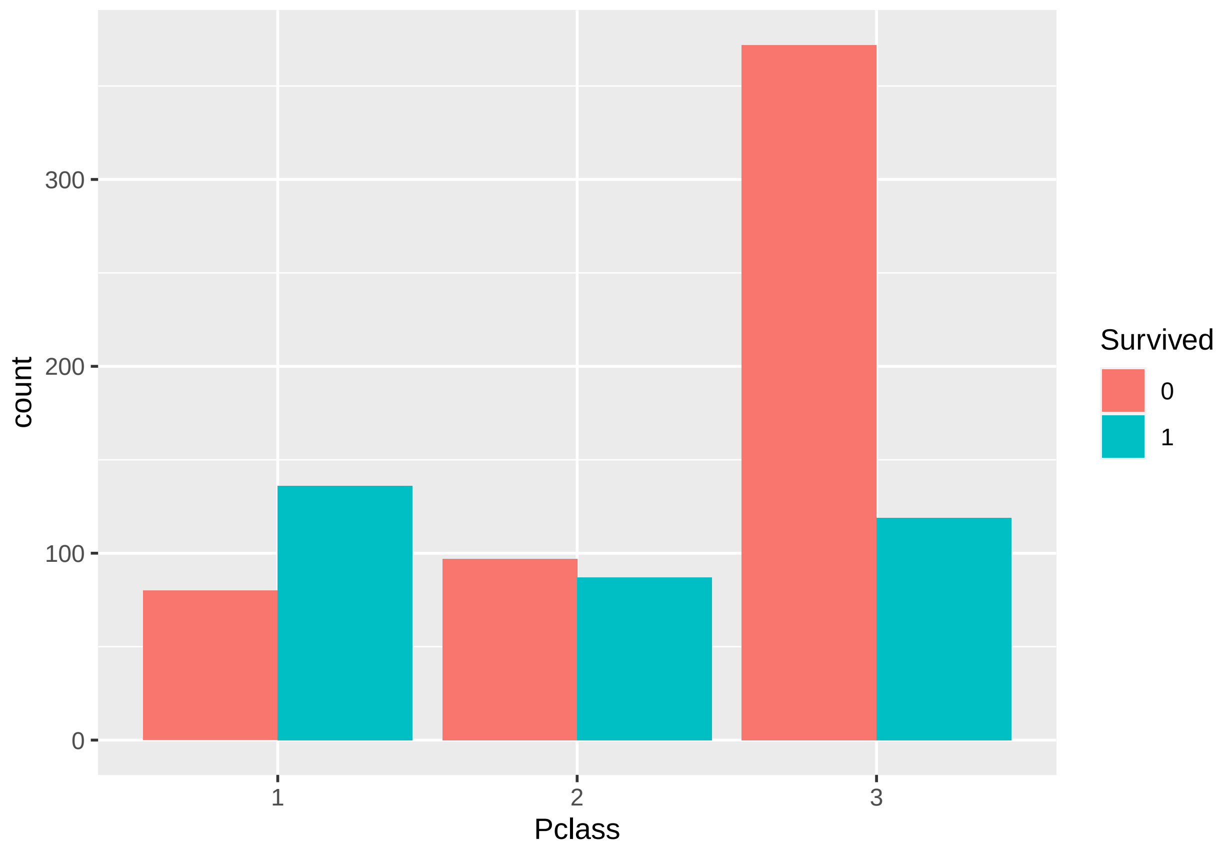
```
##  Survived Pclass      Sex           Age            SibSp           Parch
##  0:549    1:216   female:314   Min.   : 0.42   Min.   :0.000   Min.   :0.0000
##  1:342    2:184   male  :577   1st Qu.:22.00   1st Qu.:0.000   1st Qu.:0.0000
##           3:491                Median :28.00   Median :0.000   Median :0.0000
##                                Mean   :29.36   Mean   :0.523   Mean   :0.3816
##                                3rd Qu.:35.00   3rd Qu.:1.000   3rd Qu.:0.0000
##                                Max.   :80.00   Max.   :8.000   Max.   :6.0000
##       Fare        Embarked  family_size
##  Min.   :  0.00   C:168    Min.   : 0.0000
##  1st Qu.:  7.91   Q: 77    1st Qu.: 0.0000
##  Median : 14.45   S:646    Median : 0.0000
##  Mean   : 32.20            Mean   : 0.9046
##  3rd Qu.: 31.00            3rd Qu.: 1.0000
##  Max.   :512.33            Max.   :10.0000
```

Data Visualization

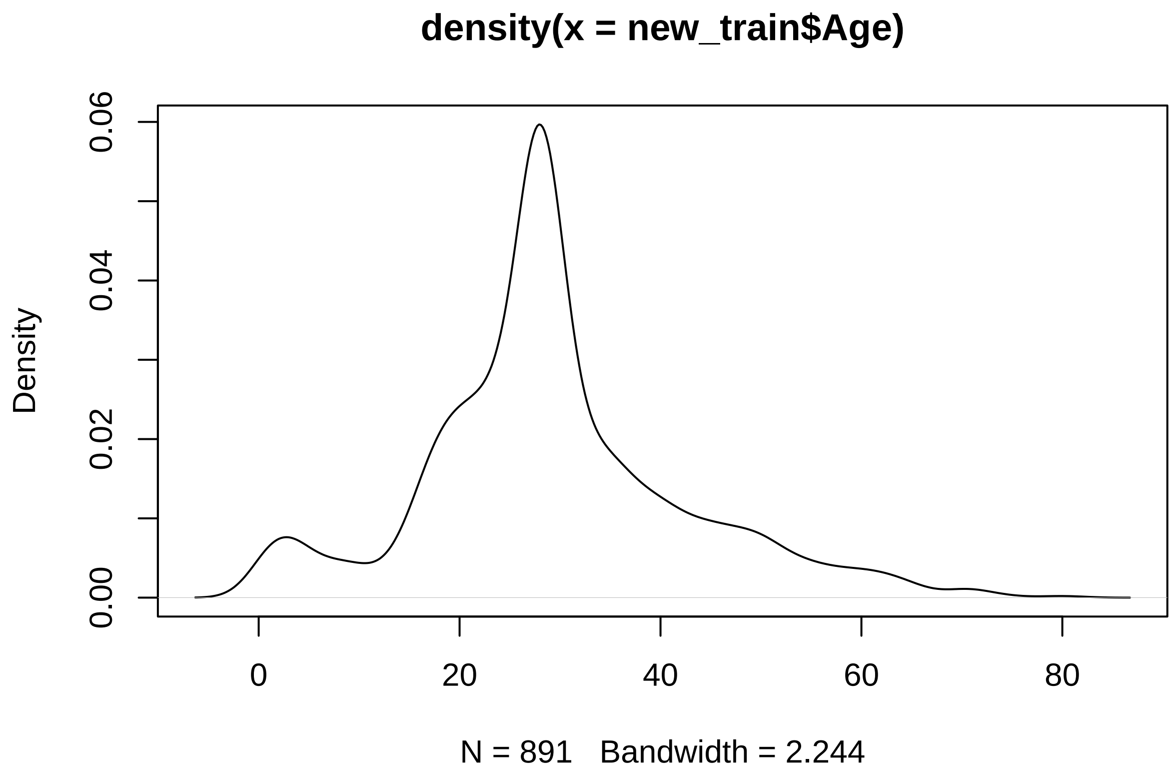
```r
sexplot<- ggplot(data=new_train, aes(x=Sex,fill=Survived)) + geom_bar(position = "dodge")
sexplot
```



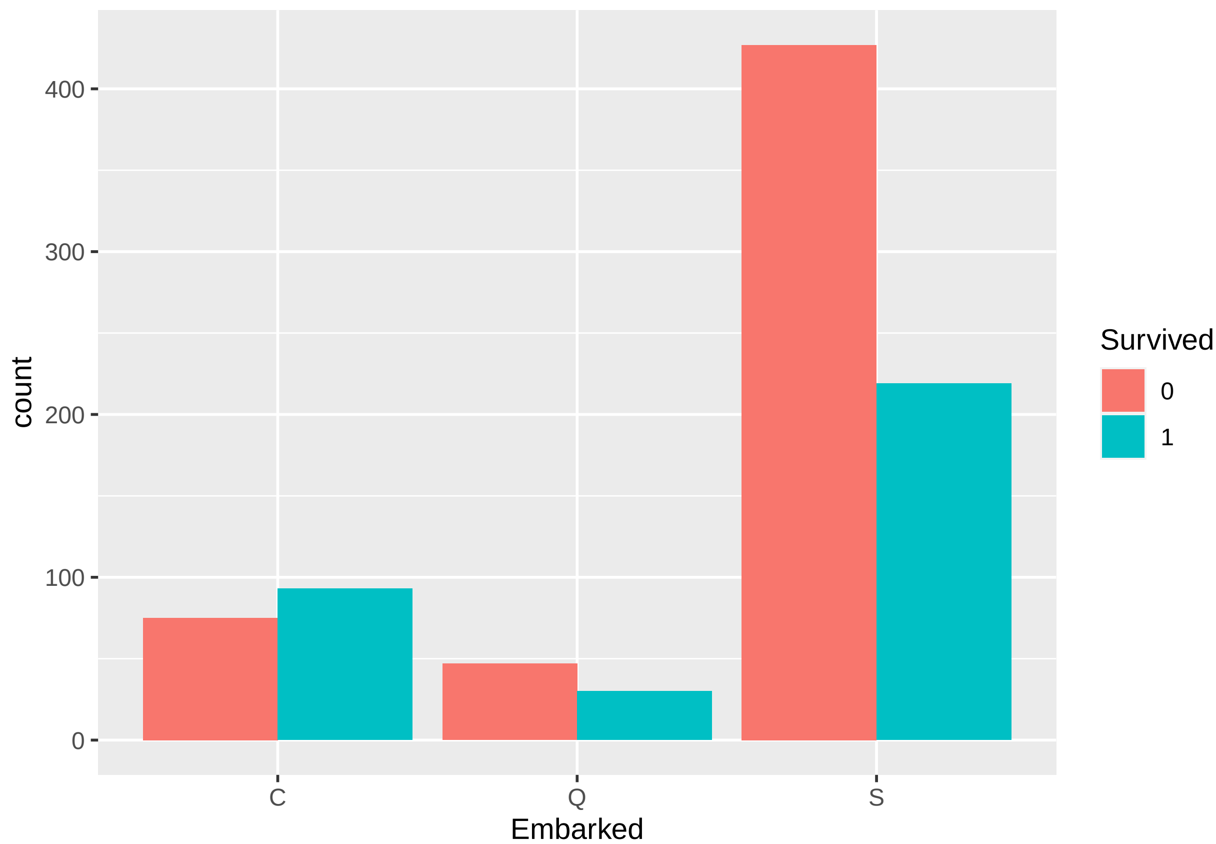

```
pclassplot <- ggplot(data=new_train, aes(x=Pclass, fill=Survived)) + geom_bar(position = "dodge")
pclassplot
```



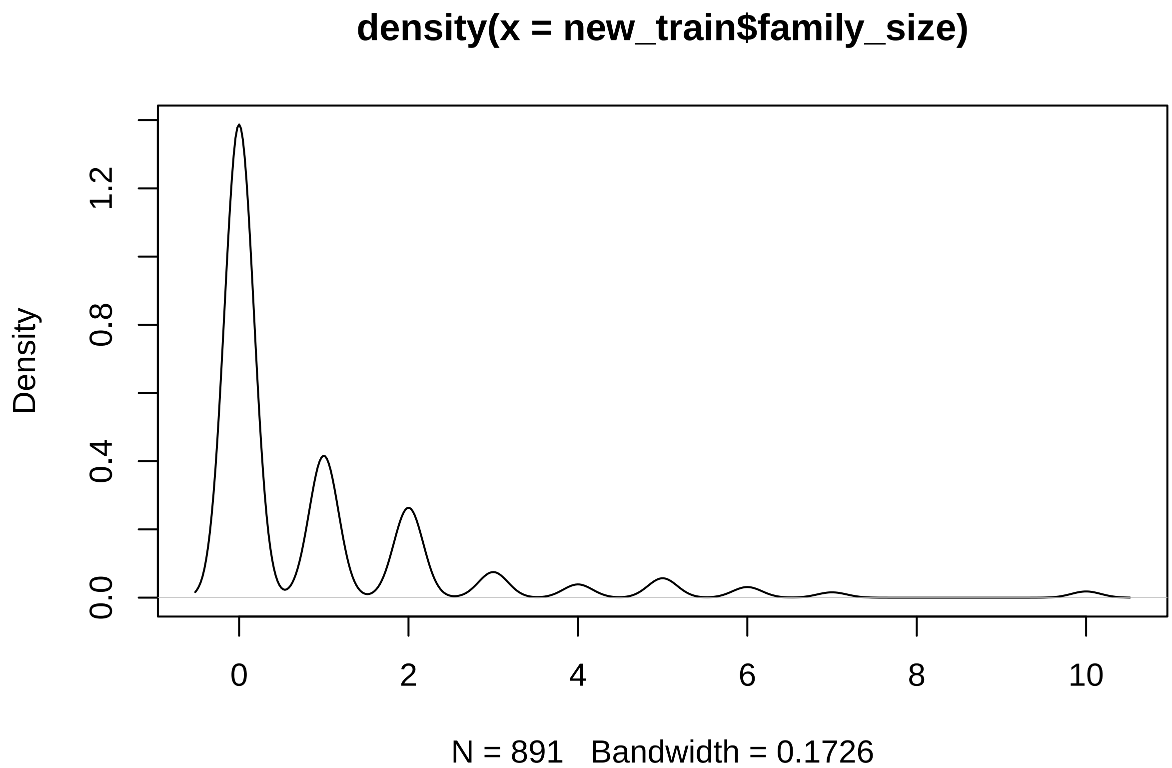

```r
ageplot<-plot(density(new_train$Age))
```



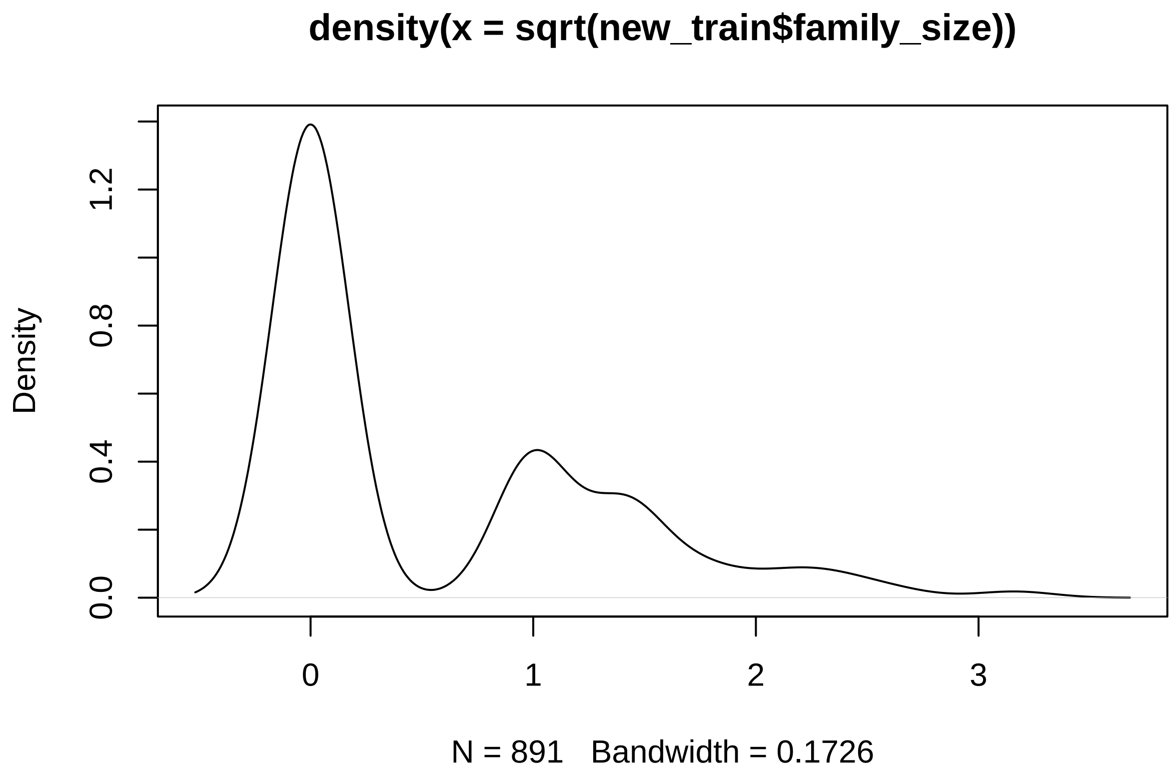

```
embarkedplot <- ggplot(data = new_train, aes(x = Embarked, fill=Survived)) + geom_bar(position = "dodge
embarkedplot
```




```
familyplot <- plot(density(new_train$family_size))
```




```r
upd_familyplot <- plot(density(sqrt(new_train$family_size)))
```



Building Model and use base accuracy to evaluate the accuracy of the model

I built a model just with Pclass because I am curious about the prediction using Pclass.

```
model1 <- glm(new_train$Survived~new_train$Pclass, family = "binomial")
summary(model1)
```

```
##
## Call:
## glm(formula = new_train$Survived ~ new_train$Pclass, family = "binomial")
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)         0.5306     0.1409   3.766 0.000166 ***
## new_train$Pclass2  -0.6394     0.2041  -3.133 0.001731 **
## new_train$Pclass3  -1.6704     0.1759  -9.496  < 2e-16 ***
## ---
## 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: 1083.1  on 888  degrees of freedom
## AIC: 1089.1
```

```
##
## Number of Fisher Scoring iterations: 4
```

```r
predict1 <- predict.glm(model1, new_train, type = "response")
predict1[predict1<.5]=0
predict1[predict1>=.5]=1
predict1 <- table(predict1, new_train$Survived)
predict1
```

```
##
## predict1   0   1
##        0 469 206
##        1  80 136
```

```r
base_accuracy <- 549/891
prediction1_accuracy <- (469+136)/891
base_accuracy
```

```
## [1] 0.6161616
```

```r
prediction1_accuracy
```

```
## [1] 0.6790123
```

I used model2 to find out which variables is significant to the model and prediction.

```r
model2 <- glm(new_train$Survived ~., data = new_train,family = "binomial")
summary(model2)
```

```
##
## Call:
## glm(formula = new_train$Survived ~ ., family = "binomial", data = new_train)
##
## Coefficients: (1 not defined because of singularities)
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.064159   0.472813   8.596  < 2e-16 ***
## Pclass2     -0.919468   0.297326  -3.092  0.00199 **
## Pclass3     -2.150048   0.297720  -7.222 5.13e-13 ***
## Sexmale     -2.719444   0.200977 -13.531  < 2e-16 ***
## Age         -0.038517   0.007855  -4.903 9.43e-07 ***
## SibSp       -0.321794   0.109193  -2.947  0.00321 **
## Parch       -0.093329   0.118856  -0.785  0.43232
## Fare         0.002339   0.002469   0.947  0.34346
## EmbarkedQ   -0.056267   0.381471  -0.148  0.88274
## EmbarkedS   -0.434226   0.239530  -1.813  0.06986 .
## family_size        NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 1186.66  on 890  degrees of freedom
## Residual deviance:  785.04  on 881  degrees of freedom
## AIC: 805.04
##
## Number of Fisher Scoring iterations: 5
```

```r
varImp(model2, scale=F)
```

```
##              Overall
## Pclass2    3.0924567
## Pclass3    7.2217091
## Sexmale   13.5310966
## Age        4.9032342
## SibSp      2.9470210
## Parch      0.7852282
## Fare       0.9473497
## EmbarkedQ  0.1475003
## EmbarkedS  1.8128230
```

```r
predict2 <- predict.glm(model2, new_train, type = "response")
predict2[predict2<.5]=0
predict2[predict2>=.5]=1
predict2 <- table(predict2, new_train$Survived)
predict2
```

```
##
## predict2   0   1
##        0 477 102
##        1  72 240
```

```r
(476+240)/891
```

```
## [1] 0.8035915
```

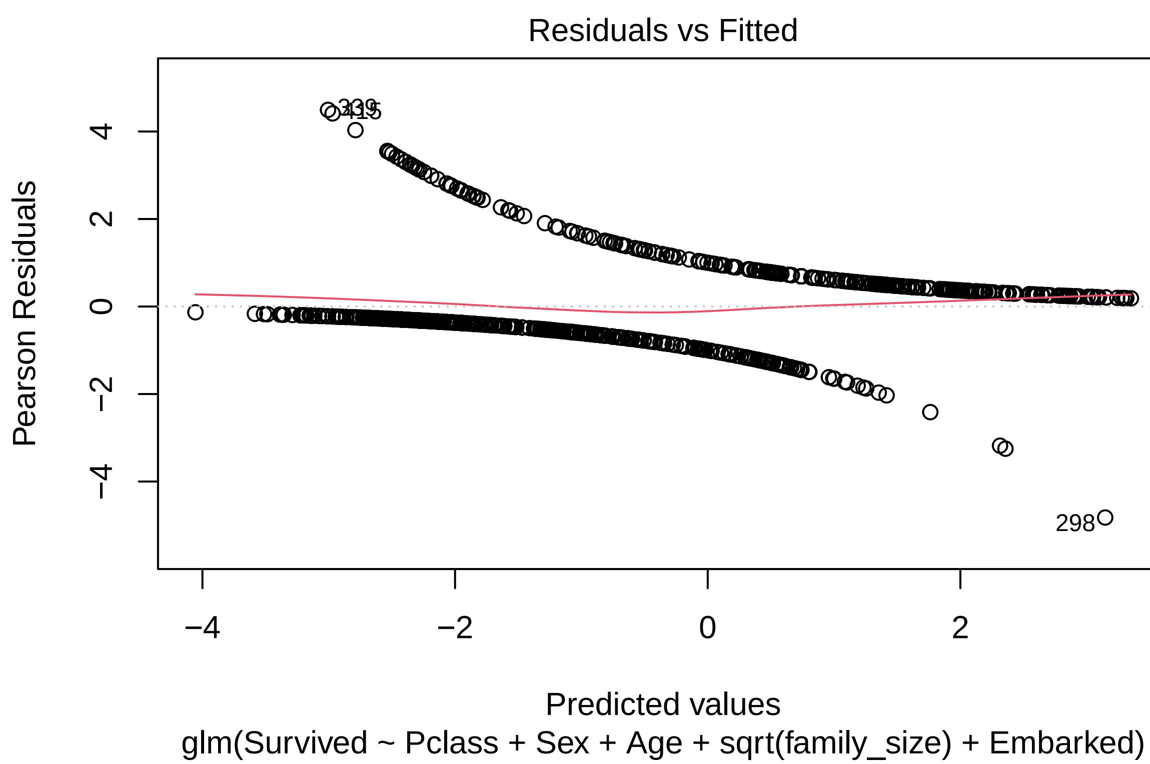
I found out that Parch and Fare variables does not play a huge role in the model according to the significant and the variables importance table. Other than that, I transform family_size variable with sqrt(x) to increase its normality.

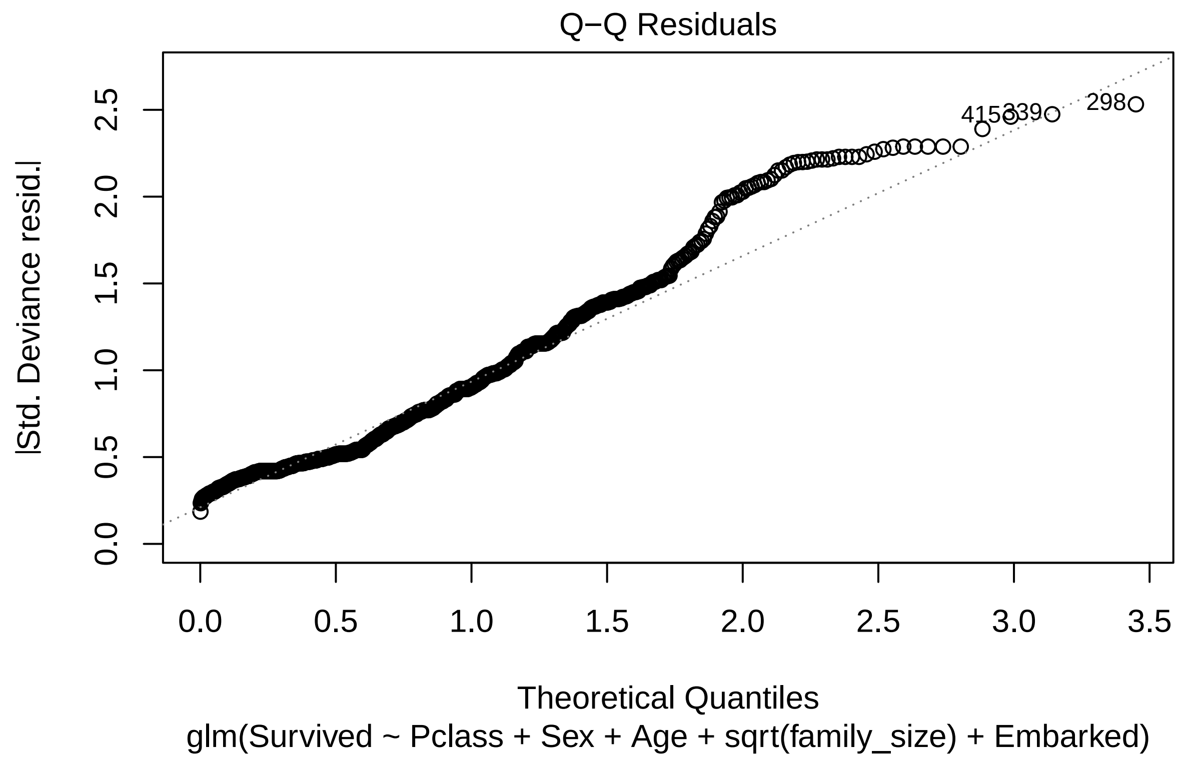
```r
model3 <- glm(Survived~ Pclass + Sex + Age  + sqrt(family_size) + Embarked, data = new_train,family = "
summary(model3)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Sex + Age + sqrt(family_size) +
##     Embarked, family = "binomial", data = new_train)
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)        4.166932   0.435505   9.568  < 2e-16 ***
## Pclass2           -0.999507   0.267118  -3.742 0.000183 ***
## Pclass3           -2.318650   0.253571  -9.144  < 2e-16 ***
```

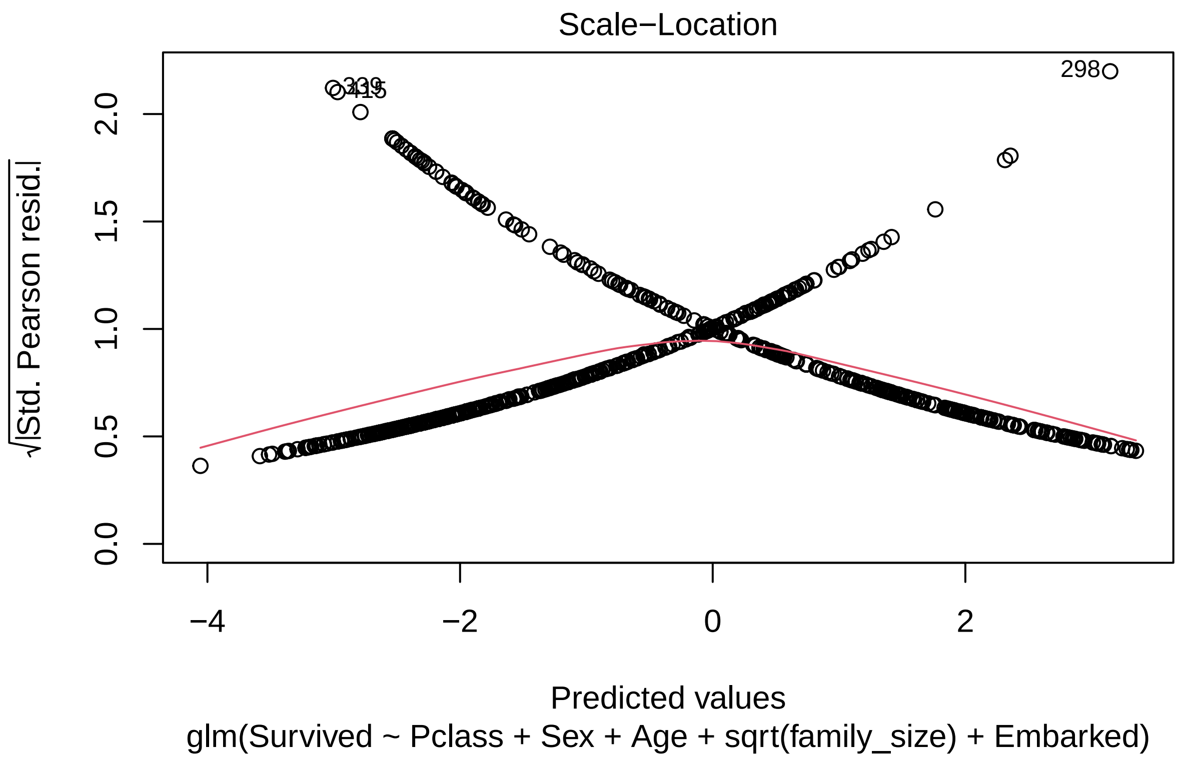
```
## Sexmale           -2.688409   0.199347 -13.486  < 2e-16 ***
## Age               -0.036196   0.007711  -4.694 2.68e-06 ***
## sqrt(family_size) -0.237471   0.125057  -1.899 0.057578 .
## EmbarkedQ         -0.093867   0.375889  -0.250 0.802804
## EmbarkedS         -0.536743   0.234174  -2.292 0.021902 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1186.66  on 890  degrees of freedom
## Residual deviance:  794.63  on 883  degrees of freedom
## AIC: 810.63
##
## Number of Fisher Scoring iterations: 5
```

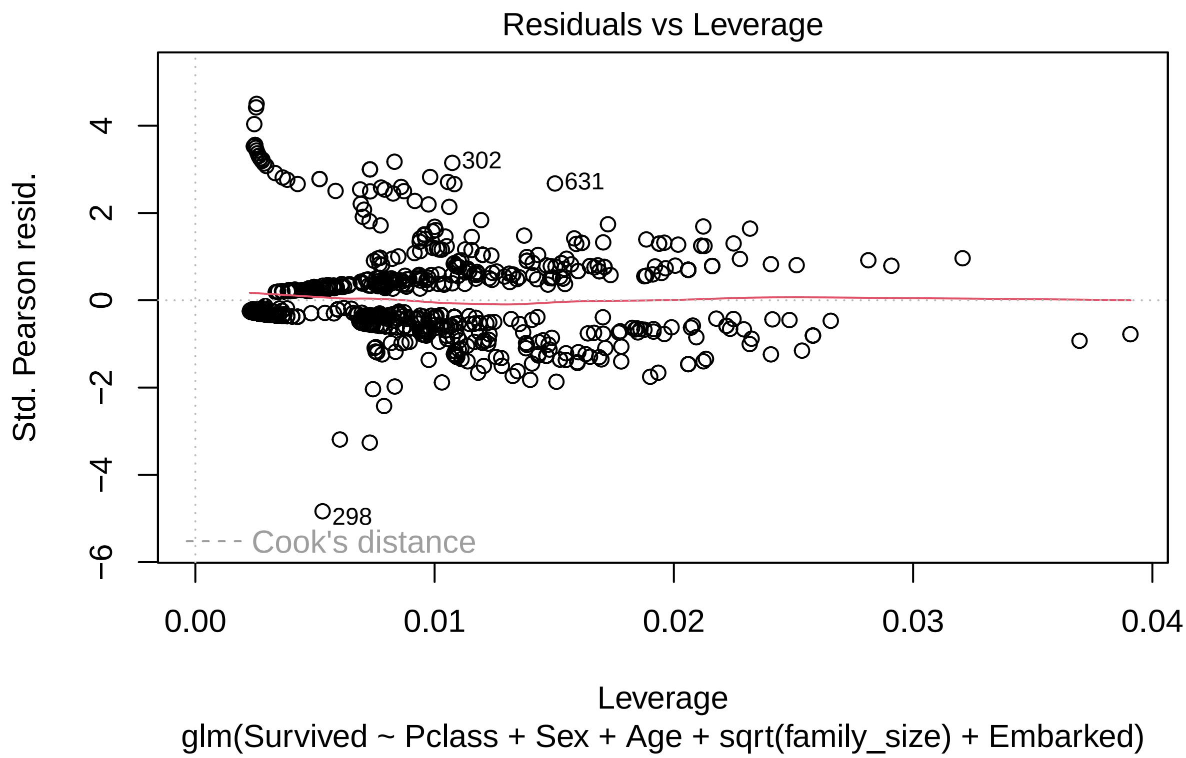
```
plot(model3)
```





Q−Q Residuals
|Std. Deviance resid.|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
Theoretical Quantiles
glm(Survived ~ Pclass + Sex + Age + sqrt(family_size) + Embarked)
415
339
298


Scale−Location
298
339
415
√|Std. Pearson resid.|
0.0
0.5
1.0
1.5
2.0
−4
−2
0
2
Predicted values
glm(Survived ~ Pclass + Sex + Age + sqrt(family_size) + Embarked)




```
varImp(model3, scale=F)
```

```
##                      Overall
## Pclass2            3.7418225
## Pclass3            9.1439806
## Sexmale           13.4860817
## Age                4.6942459
## sqrt(family_size)  1.8988974
## EmbarkedQ          0.2497199
## EmbarkedS          2.2920697
```

```
predict3 <- predict(model3, new_train, type = "response")
predict3[predict3<.5]=0
predict3[predict3>=.5]=1
predict3 <- table(predict3,train$Survived)
predict3
```

```
##
## predict3   0   1
##        0 477 101
##        1  72 241
```

```
accuracy3 <- (497+222)/891
accuracy3
```

```
## [1] 0.8069585
```

By choosing the highest accuracy rate, I chose model3 as my model to predict the test dataset.

```
test_predict <- predict(model3, new_test)
test_predict[test_predict<0.5]=0
test_predict[test_predict>=0.5]=1
new_test$survived <- as.numeric(test_predict >=0.5)
table(new_test$survived)
```

```
##
##   0   1
## 297 121
```

Write csv file for submission

```
Output <- data.frame(PassengerID = test$PassengerId, Survived = test_predict)
Output
```

```
##     PassengerID Survived
## 1           892        0
## 2           893        0
## 3           894        0
## 4           895        0
## 5           896        0
## 6           897        0
## 7           898        1
## 8           899        0
## 9           900        1
## 10          901        0
## 11          902        0
## 12          903        0
## 13          904        1
## 14          905        0
## 15          906        1
## 16          907        1
## 17          908        0
## 18          909        0
## 19          910        0
## 20          911        0
## 21          912        0
## 22          913        0
## 23          914        1
## 24          915        0
## 25          916        1
## 26          917        0
## 27          918        1
## 28          919        0
## 29          920        0
## 30          921        0
## 31          922        0
```

```
## 32     923   0
## 33     924   0
## 34     925   0
## 35     926   0
## 36     927   0
## 37     928   0
## 38     929   1
## 39     930   0
## 40     931   0
## 41     932   0
## 42     933   0
## 43     934   0
## 44     935   1
## 45     936   1
## 46     937   0
## 47     938   0
## 48     939   0
## 49     940   1
## 50     941   0
## 51     942   0
## 52     943   0
## 53     944   1
## 54     945   1
## 55     946   0
## 56     947   0
## 57     948   0
## 58     949   0
## 59     950   0
## 60     951   1
## 61     952   0
## 62     953   0
## 63     954   0
## 64     955   1
## 65     956   1
## 66     957   1
## 67     958   1
## 68     959   0
## 69     960   0
## 70     961   1
## 71     962   1
## 72     963   0
## 73     964   0
## 74     965   0
## 75     966   1
## 76     967   0
## 77     968   0
## 78     969   1
## 79     970   0
## 80     971   1
## 81     972   0
## 82     973   0
## 83     974   0
## 84     975   0
## 85     976   0
```

```
## 86           977        0
## 87           978        1
## 88           979        1
## 89           980        1
## 90           981        0
## 91           982        0
## 92           983        0
## 93           984        1
## 94           985        0
## 95           986        1
## 96           987        0
## 97           988        1
## 98           989        0
## 99           990        1
## 100          991        0
## 101          992        1
## 102          993        0
## 103          994        0
## 104          995        0
## 105          996        1
## 106          997        0
## 107          998        0
## 108          999        0
## 109         1000        0
## 110         1001        0
## 111         1002        0
## 112         1003        1
## 113         1004        1
## 114         1005        1
## 115         1006        1
## 116         1007        0
## 117         1008        0
## 118         1009        1
## 119         1010        0
## 120         1011        1
## 121         1012        1
## 122         1013        0
## 123         1014        1
## 124         1015        0
## 125         1016        0
## 126         1017        0
## 127         1018        0
## 128         1019        0
## 129         1020        0
## 130         1021        0
## 131         1022        0
## 132         1023        0
## 133         1024        0
## 134         1025        0
## 135         1026        0
## 136         1027        0
## 137         1028        0
## 138         1029        0
## 139         1030        0
```

```
## 140          1031          0
## 141          1032          0
## 142          1033          1
## 143          1034          0
## 144          1035          0
## 145          1036          0
## 146          1037          0
## 147          1038          0
## 148          1039          0
## 149          1040          0
## 150          1041          0
## 151          1042          1
## 152          1043          0
## 153          1044          0
## 154          1045          0
## 155          1046          0
## 156          1047          0
## 157          1048          1
## 158          1049          0
## 159          1050          0
## 160          1051          0
## 161          1052          1
## 162          1053          0
## 163          1054          1
## 164          1055          0
## 165          1056          0
## 166          1057          0
## 167          1058          0
## 168          1059          0
## 169          1060          1
## 170          1061          1
## 171          1062          0
## 172          1063          0
## 173          1064          0
## 174          1065          0
## 175          1066          0
## 176          1067          1
## 177          1068          1
## 178          1069          0
## 179          1070          1
## 180          1071          1
## 181          1072          0
## 182          1073          0
## 183          1074          1
## 184          1075          0
## 185          1076          1
## 186          1077          0
## 187          1078          1
## 188          1079          0
## 189          1080          0
## 190          1081          0
## 191          1082          0
## 192          1083          0
## 193          1084          0
```

```
## 194          1085          0
## 195          1086          0
## 196          1087          0
## 197          1088          1
## 198          1089          1
## 199          1090          0
## 200          1091          0
## 201          1092          1
## 202          1093          0
## 203          1094          0
## 204          1095          1
## 205          1096          0
## 206          1097          1
## 207          1098          0
## 208          1099          0
## 209          1100          1
## 210          1101          0
## 211          1102          0
## 212          1103          0
## 213          1104          0
## 214          1105          0
## 215          1106          0
## 216          1107          0
## 217          1108          1
## 218          1109          0
## 219          1110          1
## 220          1111          0
## 221          1112          1
## 222          1113          0
## 223          1114          1
## 224          1115          0
## 225          1116          1
## 226          1117          1
## 227          1118          0
## 228          1119          1
## 229          1120          0
## 230          1121          0
## 231          1122          0
## 232          1123          1
## 233          1124          0
## 234          1125          0
## 235          1126          0
## 236          1127          0
## 237          1128          0
## 238          1129          0
## 239          1130          1
## 240          1131          1
## 241          1132          1
## 242          1133          1
## 243          1134          0
## 244          1135          0
## 245          1136          0
## 246          1137          0
## 247          1138          1
```

```
## 248          1139          0
## 249          1140          1
## 250          1141          1
## 251          1142          1
## 252          1143          0
## 253          1144          0
## 254          1145          0
## 255          1146          0
## 256          1147          0
## 257          1148          0
## 258          1149          0
## 259          1150          1
## 260          1151          0
## 261          1152          0
## 262          1153          0
## 263          1154          1
## 264          1155          1
## 265          1156          0
## 266          1157          0
## 267          1158          0
## 268          1159          0
## 269          1160          0
## 270          1161          0
## 271          1162          0
## 272          1163          0
## 273          1164          1
## 274          1165          1
## 275          1166          0
## 276          1167          1
## 277          1168          0
## 278          1169          0
## 279          1170          0
## 280          1171          0
## 281          1172          0
## 282          1173          0
## 283          1174          1
## 284          1175          1
## 285          1176          1
## 286          1177          0
## 287          1178          0
## 288          1179          0
## 289          1180          0
## 290          1181          0
## 291          1182          0
## 292          1183          1
## 293          1184          0
## 294          1185          0
## 295          1186          0
## 296          1187          0
## 297          1188          1
## 298          1189          0
## 299          1190          0
## 300          1191          0
## 301          1192          0
```

```
## 302           1193            0
## 303           1194            0
## 304           1195            0
## 305           1196            1
## 306           1197            1
## 307           1198            0
## 308           1199            0
## 309           1200            0
## 310           1201            0
## 311           1202            0
## 312           1203            0
## 313           1204            0
## 314           1205            0
## 315           1206            1
## 316           1207            1
## 317           1208            0
## 318           1209            0
## 319           1210            0
## 320           1211            0
## 321           1212            0
## 322           1213            0
## 323           1214            0
## 324           1215            0
## 325           1216            1
## 326           1217            0
## 327           1218            1
## 328           1219            0
## 329           1220            0
## 330           1221            0
## 331           1222            1
## 332           1223            0
## 333           1224            0
## 334           1225            1
## 335           1226            0
## 336           1227            0
## 337           1228            0
## 338           1229            0
## 339           1230            0
## 340           1231            0
## 341           1232            0
## 342           1233            0
## 343           1234            0
## 344           1235            1
## 345           1236            0
## 346           1237            1
## 347           1238            0
## 348           1239            0
## 349           1240            0
## 350           1241            1
## 351           1242            1
## 352           1243            0
## 353           1244            0
## 354           1245            0
## 355           1246            1
```

```
## 356           1247          0
## 357           1248          1
## 358           1249          0
## 359           1250          0
## 360           1251          0
## 361           1252          0
## 362           1253          1
## 363           1254          1
## 364           1255          0
## 365           1256          1
## 366           1257          0
## 367           1258          0
## 368           1259          1
## 369           1260          1
## 370           1261          0
## 371           1262          0
## 372           1263          1
## 373           1264          0
## 374           1265          0
## 375           1266          1
## 376           1267          1
## 377           1268          0
## 378           1269          0
## 379           1270          0
## 380           1271          0
## 381           1272          0
## 382           1273          0
## 383           1274          0
## 384           1275          0
## 385           1276          0
## 386           1277          1
## 387           1278          0
## 388           1279          0
## 389           1280          0
## 390           1281          0
## 391           1282          0
## 392           1283          1
## 393           1284          0
## 394           1285          0
## 395           1286          0
## 396           1287          1
## 397           1288          0
## 398           1289          1
## 399           1290          0
## 400           1291          0
## 401           1292          1
## 402           1293          0
## 403           1294          1
## 404           1295          0
## 405           1296          0
## 406           1297          0
## 407           1298          0
## 408           1299          0
## 409           1300          1
```

```
## 410        1301        1
## 411        1302        1
## 412        1303        1
## 413        1304        0
## 414        1305        0
## 415        1306        1
## 416        1307        0
## 417        1308        0
## 418        1309        0
```

```
write.csv(Output, file= 'lgm_titanic_output.csv', row.names = F)
```