HW6 - Logistic Regression and Multiple Linear Regression

Shiloh Bradley

6/17/2020

```
## Loading required package: Rcpp

## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.7.6, built: 2019-11-24)
## ## Copyright (C) 2005-2020 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##

## Loading required package: carData

##
## Attaching package: 'dplyr'

## The following object is masked from 'package:car':
##
##     recode

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

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

##
## Attaching package: 'gridExtra'

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

## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```r
r2fun <- function(m) {
  u <- LogRegR2(m)
  data.frame(McFadden = u$RL2, CoxSnell = u$CoxR2, Nagelkerke = u$NagelkerkeR2)
}
```

German Credit

```r
G <- read.csv("german_credit.csv", header = TRUE)
```

```r
head(G)
```

```
##   Creditability Account.Balance Duration.of.Credit..month.
## 1             1               1                         18
## 2             1               1                          9
## 3             1               2                         12
## 4             1               1                         12
## 5             1               1                         12
## 6             1               1                         10
##   Payment.Status.of.Previous.Credit Purpose Credit.Amount
## 1                                 4       2          1049
## 2                                 4       0          2799
## 3                                 2       9           841
## 4                                 4       0          2122
## 5                                 4       0          2171
## 6                                 4       0          2241
##   Value.Savings.Stocks Length.of.current.employment Instalment.per.cent
## 1                    1                            2                   4
## 2                    1                            3                   2
## 3                    2                            4                   2
## 4                    1                            3                   3
## 5                    1                            3                   4
## 6                    1                            2                   1
##   Sex...Marital.Status Guarantors Duration.in.Current.address
## 1                    2          1                           4
## 2                    3          1                           2
## 3                    2          1                           4
## 4                    3          1                           2
## 5                    3          1                           4
## 6                    3          1                           3
##   Most.valuable.available.asset Age..years. Concurrent.Credits
## 1                             2          21                  3
## 2                             1          36                  3
## 3                             1          23                  3
## 4                             1          39                  3
## 5                             2          38                  1
## 6                             1          48                  3
##   Type.of.apartment No.of.Credits.at.this.Bank Occupation No.of.dependents
## 1                 1                          1          3                1
## 2                 1                          2          3                2
## 3                 1                          1          2                1
## 4                 1                          2          2                2
## 5                 2                          2          2                1
## 6                 1                          2          2                2
##   Telephone Foreign.Worker
## 1         1              1
## 2         1              1
## 3         1              1
## 4         1              2
## 5         1              2
## 6         1              2
```

```r
tail(G)
```

```
##      Creditability Account.Balance Duration.of.Credit..month.
```

```
## 995              0               1                         12
## 996              0               1                         24
## 997              0               1                         24
## 998              0               4                         21
## 999              0               2                         12
## 1000             0               1                         30
##      Payment.Status.of.Previous.Credit Purpose Credit.Amount
## 995                                  0       3          6199
## 996                                  2       3          1987
## 997                                  2       0          2303
## 998                                  4       0         12680
## 999                                  2       3          6468
## 1000                                 2       2          6350
##      Value.Savings.Stocks Length.of.current.employment Instalment.per.cent
## 995                     1                            3                   4
## 996                     1                            3                   2
## 997                     1                            5                   4
## 998                     5                            5                   4
## 999                     5                            1                   2
## 1000                    5                            5                   4
##      Sex...Marital.Status Guarantors Duration.in.Current.address
## 995                     3          1                           2
## 996                     3          1                           4
## 997                     3          2                           1
## 998                     3          1                           4
## 999                     3          1                           1
## 1000                    3          1                           4
##      Most.valuable.available.asset Age..years. Concurrent.Credits
## 995                              2          28                  3
## 996                              1          21                  3
## 997                              1          45                  3
## 998                              4          30                  3
## 999                              4          52                  3
## 1000                             2          31                  3
##      Type.of.apartment No.of.Credits.at.this.Bank Occupation
## 995                  1                          2          3
## 996                  1                          1          2
## 997                  2                          1          3
## 998                  3                          1          4
## 999                  2                          1          4
## 1000                 2                          1          3
##      No.of.dependents Telephone Foreign.Worker
## 995                 1         2              1
## 996                 2         1              1
## 997                 1         1              1
## 998                 1         2              1
## 999                 1         2              1
## 1000                1         1              1
```

```r
dim(G)
```

```
## [1] 1000   21
```

```r
names(G)
```

```
##  [1] "Creditability"
```

```
##  [2] "Account.Balance"
##  [3] "Duration.of.Credit..month."
##  [4] "Payment.Status.of.Previous.Credit"
##  [5] "Purpose"
##  [6] "Credit.Amount"
##  [7] "Value.Savings.Stocks"
##  [8] "Length.of.current.employment"
##  [9] "Instalment.per.cent"
## [10] "Sex...Marital.Status"
## [11] "Guarantors"
## [12] "Duration.in.Current.address"
## [13] "Most.valuable.available.asset"
## [14] "Age..years."
## [15] "Concurrent.Credits"
## [16] "Type.of.apartment"
## [17] "No.of.Credits.at.this.Bank"
## [18] "Occupation"
## [19] "No.of.dependents"
## [20] "Telephone"
## [21] "Foreign.Worker"
```

```r
summary(G)
```

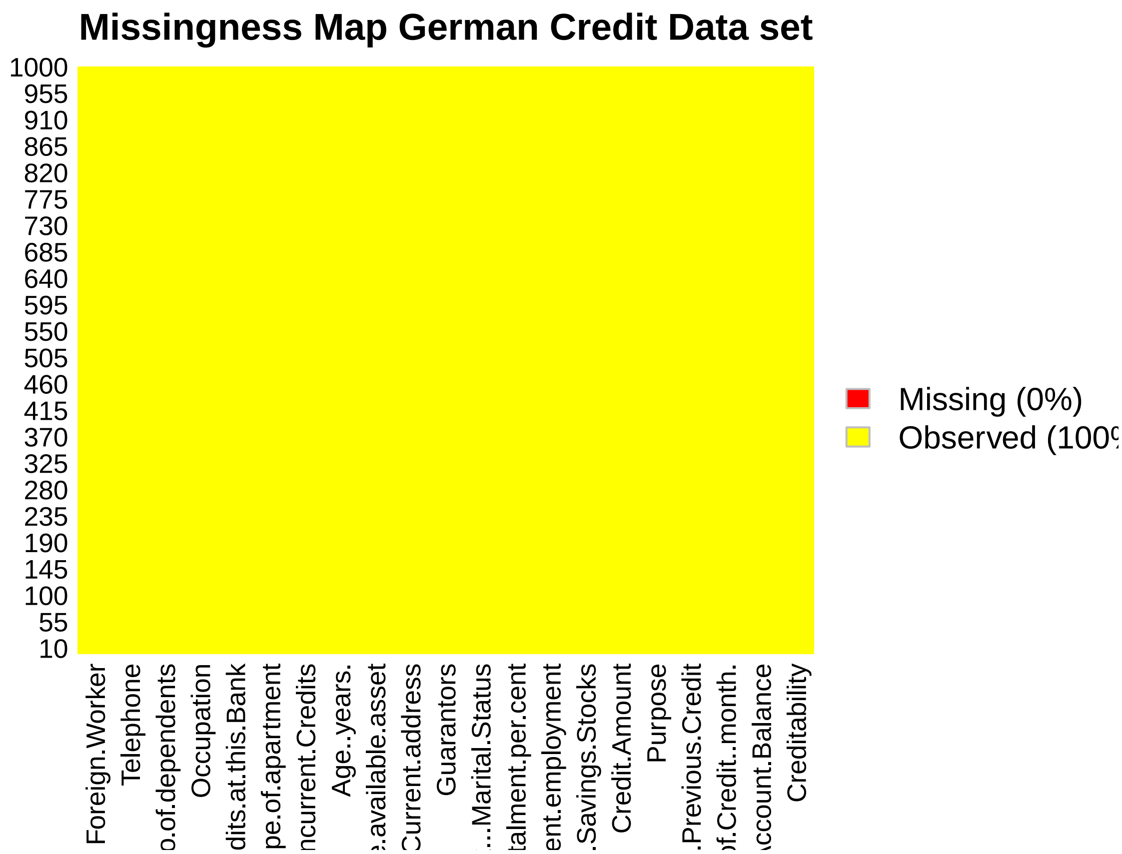
```
##  Creditability Account.Balance Duration.of.Credit..month.
##  Min.   :0.0   Min.   :1.000   Min.   : 4.0
##  1st Qu.:0.0   1st Qu.:1.000   1st Qu.:12.0
##  Median :1.0   Median :2.000   Median :18.0
##  Mean   :0.7   Mean   :2.577   Mean   :20.9
##  3rd Qu.:1.0   3rd Qu.:4.000   3rd Qu.:24.0
##  Max.   :1.0   Max.   :4.000   Max.   :72.0
##  Payment.Status.of.Previous.Credit    Purpose       Credit.Amount
##  Min.   :0.000                     Min.   : 0.000   Min.   :  250
##  1st Qu.:2.000                     1st Qu.: 1.000   1st Qu.: 1366
##  Median :2.000                     Median : 2.000   Median : 2320
##  Mean   :2.545                     Mean   : 2.828   Mean   : 3271
##  3rd Qu.:4.000                     3rd Qu.: 3.000   3rd Qu.: 3972
##  Max.   :4.000                     Max.   :10.000   Max.   :18424
##  Value.Savings.Stocks Length.of.current.employment Instalment.per.cent
##  Min.   :1.000        Min.   :1.000                Min.   :1.000
##  1st Qu.:1.000        1st Qu.:3.000                1st Qu.:2.000
##  Median :1.000        Median :3.000                Median :3.000
##  Mean   :2.105        Mean   :3.384                Mean   :2.973
##  3rd Qu.:3.000        3rd Qu.:5.000                3rd Qu.:4.000
##  Max.   :5.000        Max.   :5.000                Max.   :4.000
##  Sex...Marital.Status   Guarantors    Duration.in.Current.address
##  Min.   :1.000        Min.   :1.000   Min.   :1.000
##  1st Qu.:2.000        1st Qu.:1.000   1st Qu.:2.000
##  Median :3.000        Median :1.000   Median :3.000
##  Mean   :2.682        Mean   :1.145   Mean   :2.845
##  3rd Qu.:3.000        3rd Qu.:1.000   3rd Qu.:4.000
##  Max.   :4.000        Max.   :3.000   Max.   :4.000
##  Most.valuable.available.asset  Age..years.    Concurrent.Credits
##  Min.   :1.000                 Min.   :19.00   Min.   :1.000
##  1st Qu.:1.000                 1st Qu.:27.00   1st Qu.:3.000
##  Median :2.000                 Median :33.00   Median :3.000
```

```
##  Mean   :2.358                Mean   :35.54   Mean   :2.675
##  3rd Qu.:3.000                3rd Qu.:42.00   3rd Qu.:3.000
##  Max.   :4.000                Max.   :75.00   Max.   :3.000
##  Type.of.apartment No.of.Credits.at.this.Bank   Occupation
##  Min.   :1.000     Min.   :1.000             Min.   :1.000
##  1st Qu.:2.000     1st Qu.:1.000             1st Qu.:3.000
##  Median :2.000     Median :1.000             Median :3.000
##  Mean   :1.928     Mean   :1.407             Mean   :2.904
##  3rd Qu.:2.000     3rd Qu.:2.000             3rd Qu.:3.000
##  Max.   :3.000     Max.   :4.000             Max.   :4.000
##  No.of.dependents   Telephone     Foreign.Worker
##  Min.   :1.000    Min.   :1.000   Min.   :1.000
##  1st Qu.:1.000    1st Qu.:1.000   1st Qu.:1.000
##  Median :1.000    Median :1.000   Median :1.000
##  Mean   :1.155    Mean   :1.404   Mean   :1.037
##  3rd Qu.:1.000    3rd Qu.:2.000   3rd Qu.:1.000
##  Max.   :2.000    Max.   :2.000   Max.   :2.000
```

```
## Check for NA's in the data
sapply(G, function(x) sum(is.na(x)))
```

```
##                     Creditability                    Account.Balance
##                                 0                                  0
##        Duration.of.Credit..month. Payment.Status.of.Previous.Credit
##                                 0                                  0
##                           Purpose                      Credit.Amount
##                                 0                                  0
##              Value.Savings.Stocks      Length.of.current.employment
##                                 0                                  0
##               Instalment.per.cent               Sex...Marital.Status
##                                 0                                  0
##                        Guarantors       Duration.in.Current.address
##                                 0                                  0
##      Most.valuable.available.asset                       Age..years.
##                                 0                                  0
##                Concurrent.Credits                 Type.of.apartment
##                                 0                                  0
##        No.of.Credits.at.this.Bank                        Occupation
##                                 0                                  0
##                  No.of.dependents                         Telephone
##                                 0                                  0
##                    Foreign.Worker
##                                 0
```

```
missmap(G, col = c("red", "yellow"), main = "Missingness Map German Credit Data set")
```




```
## Checking more information about the variables and seeing what type of variables there are
table(G$Account.Balance)
```

```
##
##   1   2   3   4
## 274 269  63 394
```

```
summary(G$Duration.of.Credit..month.)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     4.0    12.0    18.0    20.9    24.0    72.0
```

```
#duration.of.Credit..month.  is a continuous predictor

table(G$Payment.Status.of.Previous.Credit) # categorical
```

```
##
##   0   1   2   3   4
##  40  49 530  88 293
```

```
table(G$Purpose)
```

```
##
##   0   1   2   3   4   5   6   8   9  10
## 234 103 181 280  12  22  50   9  97  12
```

```
#too many categories, so collapse at high end
G$Purpose[G$Purpose>=3] <- 4 #   (recoded)
table(G$Purpose)
```

```
##
##   0   1   2   4
## 234 103 181 482
```

```r
summary(G$Credit.Amount)  # - continuous
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     250    1366    2320    3271    3972   18424
```

```r
table(G$Value.Savings.Stocks)  # - categorical
```

```
##
##   1   2   3   4   5
## 603 103  63  48 183
```

```r
table(G$Length.of.current.employment) # - categorical
```

```
##
##   1   2   3   4   5
##  62 172 339 174 253
```

```r
table(G$Instalment.per.cent) # - categorical
```

```
##
##   1   2   3   4
## 136 231 157 476
```

```r
table(G$Guarantors)  # - categorical
```

```
##
##   1   2   3
## 907  41  52
```

```r
table(G$Duration.in.Current.address)  # - categorical
```

```
##
##   1   2   3   4
## 130 308 149 413
```

```r
table(G$Most.valuable.available.asse) # - categorical
```

```
##
##   1   2   3   4
## 282 232 332 154
```

```r
summary(G$Age..years.)  # - continuous
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   19.00   27.00   33.00   35.54   42.00   75.00
```

```r
table(G$Concurrent.Credits)  # - categorical
```

```
##
##   1   2   3
## 139  47 814
```

```r
table(G$Type.of.apartment)  # - categorical
```

```
##
##   1   2   3
## 179 714 107
```

```r
table(G$No.of.Credits.at.this.Bank)  # - categorical
```

```
##
##   1   2   3   4
## 633 333  28   6
```

```r
table(G$Occupation)  # - categorical
```

```
##
##   1   2   3   4
##  22 200 630 148
```

```r
table(G$No.of.dependents)  # - categorical
```

```
##
##   1   2
## 845 155
```

```r
table(G$Telephone)  #  categorical
```

```
##
##   1   2
## 596 404
```

```r
table(G$Foreign.Worker) # categorical
```

```
##
##   1   2
## 963  37
```

```r
G$Account.Balance <- factor(G$Account.Balance)
#Duration.of.Credit..month.  is a continuous predictor
G$Payment.Status.of.Previous.Credit <- factor(G$Payment.Status.of.Previous.Credit)
G$Purpose[G$Purpose>=4] <- 4
G$Purpose <- factor(G$Purpose)
table(G$Purpose)
```

```
##
##   0   1   2   4
## 234 103 181 482
```

```r
#G$Credit.Amount  - continuous
G$Value.Savings.Stocks <- factor(G$Value.Savings.Stocks)
G$Length.of.current.employment <- factor(G$Length.of.current.employment)
G$Instalment.per.cent <- factor(G$Instalment.per.cent)
G$Guarantors <- factor(G$Guarantors)
#table(G$Guarantors) - categorical
G$Duration.in.Current.address <- factor(G$Duration.in.Current.address)
G$Most.valuable.available.asset <- factor(G$Most.valuable.available.asset)
#G$Age..years. - continuous
G$Concurrent.Credits <- factor(G$Concurrent.Credits)
G$Type.of.apartment <- factor(G$Type.of.apartment)
G$No.of.Credits.at.this.Bank <- factor(G$No.of.Credits.at.this.Bank)
G$Occupation <- factor(G$Occupation)
G$No.of.dependents <- factor(G$No.of.dependents)
G$Telephone <- factor(G$Telephone)
G$Foreign.Worker <- factor(G$Foreign.Worker)
names(G)
```

```
##  [1] "Creditability"
##  [2] "Account.Balance"
##  [3] "Duration.of.Credit..month."
##  [4] "Payment.Status.of.Previous.Credit"
##  [5] "Purpose"
##  [6] "Credit.Amount"
##  [7] "Value.Savings.Stocks"
##  [8] "Length.of.current.employment"
##  [9] "Instalment.per.cent"
## [10] "Sex...Marital.Status"
## [11] "Guarantors"
## [12] "Duration.in.Current.address"
## [13] "Most.valuable.available.asset"
## [14] "Age..years."
## [15] "Concurrent.Credits"
## [16] "Type.of.apartment"
## [17] "No.of.Credits.at.this.Bank"
## [18] "Occupation"
## [19] "No.of.dependents"
## [20] "Telephone"
## [21] "Foreign.Worker"
```

```
dim(G) # 1000   21
```

```
## [1] 1000   21
```

```
M <- .25 * nrow(G)
## To be able to replicate the results,
## set initial seed for random number generator
set.seed(117317)
holdout <- sample(1:nrow(G), M, replace = F)

G.train <- G[-holdout, ] ## Training set
G.test <- G[holdout, ] ## Test set
dim(G.train) ## 1500 18
```

```
## [1] 750  21
```

```
dim(G.test) ## 500 18
```

```
## [1] 250  21
```

```
names(G.train)
```

```
##  [1] "Creditability"
##  [2] "Account.Balance"
##  [3] "Duration.of.Credit..month."
##  [4] "Payment.Status.of.Previous.Credit"
##  [5] "Purpose"
##  [6] "Credit.Amount"
##  [7] "Value.Savings.Stocks"
##  [8] "Length.of.current.employment"
##  [9] "Instalment.per.cent"
## [10] "Sex...Marital.Status"
## [11] "Guarantors"
## [12] "Duration.in.Current.address"
## [13] "Most.valuable.available.asset"
```

```
## [14] "Age..years."
## [15] "Concurrent.Credits"
## [16] "Type.of.apartment"
## [17] "No.of.Credits.at.this.Bank"
## [18] "Occupation"
## [19] "No.of.dependents"
## [20] "Telephone"
## [21] "Foreign.Worker"
```

```
# LR1 <- glm(Florence ~ ., family = binomial("logit"), data = G.train)
LR1 <- glm(Creditability ~ Account.Balance + Duration.of.Credit..month. + Payment.Status.of.Previous.Cre
                Purpose + Credit.Amount + Value.Savings.Stocks + Length.of.current.employment +
                Instalment.per.cent + Sex...Marital.Status + Guarantors + Duration.in.Current.address +
                Most.valuable.available.asset + Age..years. + Type.of.apartment,
                family = binomial("logit"), data = G.train)
smre1 <- summary(LR1)
smre1 ## As long as at least one category of a variable is significant, keep the variable
```

```
##
## Call:
## glm(formula = Creditability ~ Account.Balance + Duration.of.Credit..month. +
##     Payment.Status.of.Previous.Credit + Purpose + Credit.Amount +
##     Value.Savings.Stocks + Length.of.current.employment + Instalment.per.cent +
##     Sex...Marital.Status + Guarantors + Duration.in.Current.address +
##     Most.valuable.available.asset + Age..years. + Type.of.apartment,
##     family = binomial("logit"), data = G.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5763  -0.5748   0.3963   0.6687   2.3696
##
## Coefficients:
##                                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)                        -1.3212630  0.9278183  -1.424 0.154431
## Account.Balance2                    0.3693380  0.2541708   1.453 0.146193
## Account.Balance3                    1.6780264  0.4773243   3.515 0.000439
## Account.Balance4                    1.5136355  0.2623542   5.769 7.95e-09
## Duration.of.Credit..month.         -0.0151325  0.0108257  -1.398 0.162165
## Payment.Status.of.Previous.Credit1 -0.8768428  0.6749923  -1.299 0.193930
## Payment.Status.of.Previous.Credit2  0.9849756  0.4593298   2.144 0.032003
## Payment.Status.of.Previous.Credit3  1.0775043  0.5443081   1.980 0.047750
## Payment.Status.of.Previous.Credit4  1.7514385  0.4915705   3.563 0.000367
## Purpose1                            1.5661994  0.4186118   3.741 0.000183
## Purpose2                            0.4677357  0.2965934   1.577 0.114789
## Purpose4                            0.5668973  0.2527810   2.243 0.024920
## Credit.Amount                      -0.0001185  0.0000476  -2.490 0.012785
## Value.Savings.Stocks2               0.1610232  0.3494939   0.461 0.644991
## Value.Savings.Stocks3               0.6722555  0.4779691   1.406 0.159581
## Value.Savings.Stocks4               1.2262780  0.6185576   1.982 0.047426
## Value.Savings.Stocks5               1.0428556  0.3077501   3.389 0.000702
## Length.of.current.employment2      -0.0512996  0.4581254  -0.112 0.910842
## Length.of.current.employment3      -0.0730603  0.4216700  -0.173 0.862444
## Length.of.current.employment4       0.5002330  0.4686829   1.067 0.285829
## Length.of.current.employment5       0.1606859  0.4358458   0.369 0.712369
## Instalment.per.cent2               -0.3824222  0.3626534  -1.055 0.291649
```

```
## Instalment.per.cent3             -0.5197355  0.3964007  -1.311 0.189812
## Instalment.per.cent4             -1.1036970  0.3522736  -3.133 0.001730
## Sex...Marital.Status              0.1607236  0.1441873   1.115 0.264985
## Guarantors2                      -0.4517858  0.4496069  -1.005 0.314971
## Guarantors3                       1.2708722  0.5155660   2.465 0.013701
## Duration.in.Current.address2     -0.3793044  0.3415000  -1.111 0.266697
## Duration.in.Current.address3     -0.1148197  0.3814184  -0.301 0.763389
## Duration.in.Current.address4     -0.0662511  0.3504080  -0.189 0.850039
## Most.valuable.available.asset2   -0.2371995  0.2880830  -0.823 0.410297
## Most.valuable.available.asset3   -0.1388165  0.2763894  -0.502 0.615492
## Most.valuable.available.asset4   -0.8447166  0.4913385  -1.719 0.085575
## Age..years.                       0.0196074  0.0107245   1.828 0.067507
## Type.of.apartment2                0.4851480  0.2745135   1.767 0.077178
## Type.of.apartment3                0.1340627  0.5643591   0.238 0.812231
##
## (Intercept)
## Account.Balance2
## Account.Balance3                 ***
## Account.Balance4                 ***
## Duration.of.Credit..month.
## Payment.Status.of.Previous.Credit1
## Payment.Status.of.Previous.Credit2 *
## Payment.Status.of.Previous.Credit3 *
## Payment.Status.of.Previous.Credit4 ***
## Purpose1                         ***
## Purpose2
## Purpose4                         *
## Credit.Amount                    *
## Value.Savings.Stocks2
## Value.Savings.Stocks3
## Value.Savings.Stocks4            *
## Value.Savings.Stocks5            ***
## Length.of.current.employment2
## Length.of.current.employment3
## Length.of.current.employment4
## Length.of.current.employment5
## Instalment.per.cent2
## Instalment.per.cent3
## Instalment.per.cent4             **
## Sex...Marital.Status
## Guarantors2
## Guarantors3                      *
## Duration.in.Current.address2
## Duration.in.Current.address3
## Duration.in.Current.address4
## Most.valuable.available.asset2
## Most.valuable.available.asset3
## Most.valuable.available.asset4   .
## Age..years.                      .
## Type.of.apartment2               .
## Type.of.apartment3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 895.04  on 749  degrees of freedom
## Residual deviance: 657.29  on 714  degrees of freedom
## AIC: 729.29
##
## Number of Fisher Scoring iterations: 5
```

```r
      ## Need to keep all the coefficients for a significant categorical variable
## This comes from the car package
vif1 <- vif(LR1)
min(vif1)
```

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

```r
max(vif1) ## No multicollinearity since the max VIF is 4, which is less than 5
```

```
## [1] 4
```

```r
G$Creditability <- factor(G$Creditability)
## Use this to predict training set and also the test set
LR2 <- glm(Creditability ~ Account.Balance + Duration.of.Credit..month. + Payment.Status.of.Previous.Cre
                Value.Savings.Stocks + Instalment.per.cent + Sex...Marital.Status +
                Most.valuable.available.asset + Type.of.apartment,
                family = binomial("logit"), data = G.train)

smre2 <- summary(LR2)
smre2
```

```
##
## Call:
## glm(formula = Creditability ~ Account.Balance + Duration.of.Credit..month. +
##     Payment.Status.of.Previous.Credit + Value.Savings.Stocks +
##     Instalment.per.cent + Sex...Marital.Status + Most.valuable.available.asset +
##     Type.of.apartment, family = binomial("logit"), data = G.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7258  -0.6476   0.4560   0.7304   1.9895
##
## Coefficients:
##                                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)                        -0.747772   0.646764  -1.156 0.247610
## Account.Balance2                    0.343488   0.237143   1.448 0.147493
## Account.Balance3                    1.566820   0.463895   3.378 0.000731
## Account.Balance4                    1.435546   0.250615   5.728 1.02e-08
## Duration.of.Credit..month.         -0.023987   0.008289  -2.894 0.003806
## Payment.Status.of.Previous.Credit1 -0.580878   0.639554  -0.908 0.363744
## Payment.Status.of.Previous.Credit2  1.036821   0.443593   2.337 0.019422
## Payment.Status.of.Previous.Credit3  1.028939   0.527473   1.951 0.051093
## Payment.Status.of.Previous.Credit4  1.786845   0.471936   3.786 0.000153
## Value.Savings.Stocks2               0.135966   0.326054   0.417 0.676675
## Value.Savings.Stocks3               0.745557   0.451460   1.651 0.098650
## Value.Savings.Stocks4               1.299530   0.598644   2.171 0.029947
## Value.Savings.Stocks5               0.943306   0.289299   3.261 0.001112
## Instalment.per.cent2               -0.235645   0.337336  -0.699 0.484834
```

```
## Instalment.per.cent3                 -0.271688   0.365856  -0.743 0.457718
## Instalment.per.cent4                 -0.754000   0.308463  -2.444 0.014510
## Sex...Marital.Status                  0.210468   0.135462   1.554 0.120256
## Most.valuable.available.asset2       -0.362346   0.268595  -1.349 0.177323
## Most.valuable.available.asset3       -0.300505   0.259229  -1.159 0.246364
## Most.valuable.available.asset4       -1.131588   0.447335  -2.530 0.011419
## Type.of.apartment2                    0.538927   0.246946   2.182 0.029083
## Type.of.apartment3                    0.701937   0.506869   1.385 0.166099
##
## (Intercept)
## Account.Balance2
## Account.Balance3                     ***
## Account.Balance4                     ***
## Duration.of.Credit..month.           **
## Payment.Status.of.Previous.Credit1
## Payment.Status.of.Previous.Credit2 *
## Payment.Status.of.Previous.Credit3 .
## Payment.Status.of.Previous.Credit4 ***
## Value.Savings.Stocks2
## Value.Savings.Stocks3                .
## Value.Savings.Stocks4                *
## Value.Savings.Stocks5                **
## Instalment.per.cent2
## Instalment.per.cent3
## Instalment.per.cent4                 *
## Sex...Marital.Status
## Most.valuable.available.asset2
## Most.valuable.available.asset3
## Most.valuable.available.asset4       *
## Type.of.apartment2                   *
## Type.of.apartment3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 895.04  on 749  degrees of freedom
## Residual deviance: 701.64  on 728  degrees of freedom
## AIC: 745.64
##
## Number of Fisher Scoring iterations: 5
```

```
vif2 <- vif(LR2)
max(vif2)
```

```
## [1] 4
```

```
# write.csv(smre1$coefficients, "Final LR Model for German Credit Data.csv")
confint(LR2)
```

```
## Waiting for profiling to be done...

##                                           2.5 %       97.5 %
## (Intercept)                          -2.03331657  0.510620680
## Account.Balance2                     -0.12065367  0.810255175
## Account.Balance3                      0.70925921  2.549642580
```

```
## Account.Balance4                      0.95017858  1.934443214
## Duration.of.Credit..month.           -0.04032209 -0.007762539
## Payment.Status.of.Previous.Credit1 -1.85810507  0.664475191
## Payment.Status.of.Previous.Credit2  0.18437995  1.935777501
## Payment.Status.of.Previous.Credit3  0.01222900  2.089073766
## Payment.Status.of.Previous.Credit4  0.88009894  2.740471186
## Value.Savings.Stocks2              -0.49106080  0.791695344
## Value.Savings.Stocks3              -0.08921448  1.699042971
## Value.Savings.Stocks4               0.23356287  2.627941461
## Value.Savings.Stocks5               0.39212773  1.530451484
## Instalment.per.cent2               -0.90784192  0.418368527
## Instalment.per.cent3               -0.99611076  0.442422536
## Instalment.per.cent4               -1.37585854 -0.163081622
## Sex...Marital.Status               -0.05445402  0.477381869
## Most.valuable.available.asset2     -0.89167047  0.163195659
## Most.valuable.available.asset3     -0.81282277  0.205266729
## Most.valuable.available.asset4     -2.01171379 -0.252398664
## Type.of.apartment2                  0.05120326  1.021119146
## Type.of.apartment3                 -0.28619220  1.705478506
```

```
##Predict training data using the model LR1
observed.train <- G.train$Creditability
predicted.train <- predict(LR2, G.train, type = "response")
## predict.train consists of P(Y=1) for each observation in the training set
predicted.train <- round(predicted.train) ## Round to 0 or 1 to get the Y values

## Evaluate Performance of the LR Classifier on the training set
## Confusion Matrix of observed versus predicted Y values
CM.train <- table(observed.train, predicted.train)
CM.train
```

```
##               predicted.train
## observed.train   0   1
##              0  89 124
##              1  43 494
```

```
FP <- CM.train[1,2]/(CM.train[1,1]+CM.train[1,2]) ## false positive
FN <- CM.train[2,1]/(CM.train[2,1]+CM.train[2,2]) ## false negative
FP
```

```
## [1] 0.5821596
```

```
FN
```

```
## [1] 0.08007449
```

```
## Overall accuracy
## Calculated by summing the correct predictions divided by the total of the Confusion Matrix
OA.train <- sum(diag(CM.train)) / sum(CM.train)
OA.train
```

```
## [1] 0.7773333
```

```
## Precision, Recall, F1-Measure are performance measures for binary prediction
## F1-measure = geometric mean of the Precision and Recall

## Compute Precision, Recall, F1 for Category 1, model LR1
Recall1 <- CM.train[2,2] / (CM.train[2,1] + CM.train[2,2]) ## diag/row sum
```

```r
Precision1 <- CM.train[2,2] / (CM.train[1,2] + CM.train[2,2]) ## diag/column sum
F1.1 <- 2 / ((1 / Recall1) + (1 / Precision1)) ## The geometric mean

PRF1_train_1 <- (c(Precision1, Recall1, F1.1))
PRF1_train_1
```

```
## [1] 0.7993528 0.9199255 0.8554113
```

```r
# Category 0
## Repeat formulas, but use different positions in the Confusion Matrix
Recall0 <- CM.train[1,1] / (CM.train[1,1] + CM.train[1,2])
Precision0 <- CM.train[1,1] / (CM.train[1,1] + CM.train[2,1])
F1.0 <- 2 / ((1 / Recall0) + (1 / Precision0))

PRF1_train_0 <- c(Precision0, Recall0, F1.0)
PRF1_train_0
```

```
## [1] 0.6742424 0.4178404 0.5159420
```

```r
## Why is the LR2 model doing a better job of predicting 1's and so-so job for category 0?
## Because there are so many more 1's than 0's. This is an example of an unbalanced data set.
## There are methods for dealing with unbalanced data sets.
```

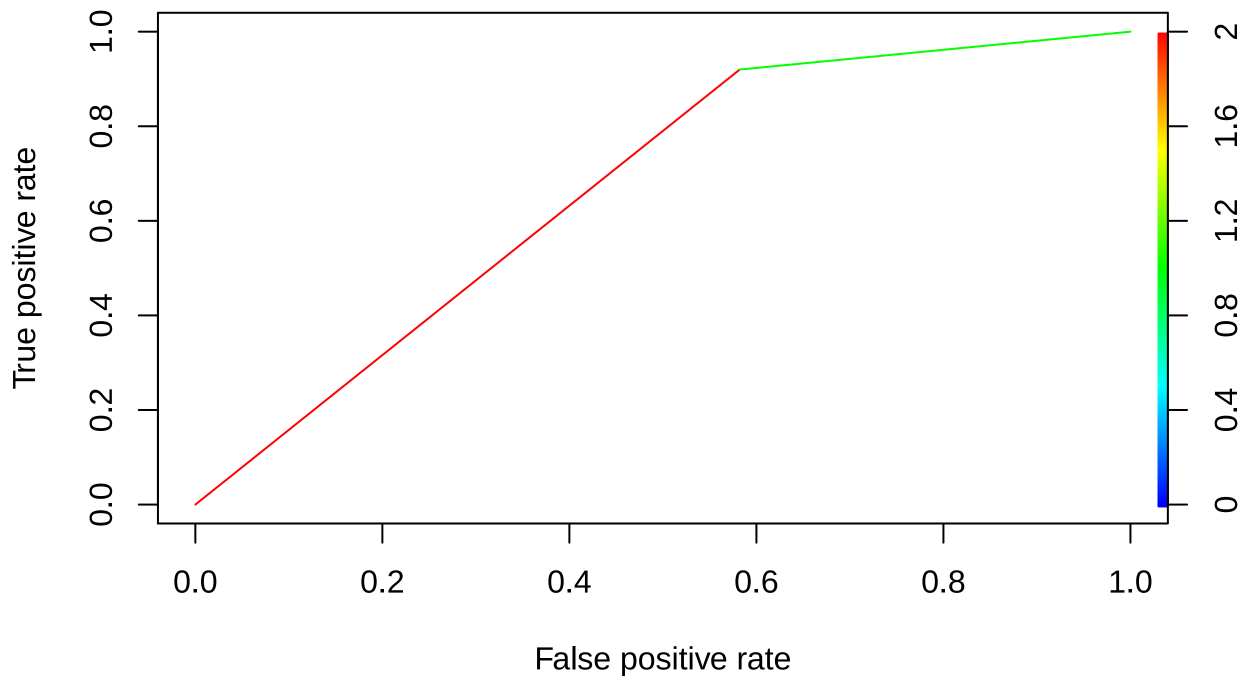
```r
r2fun(LR1)
```

```
##   McFadden  CoxSnell Nagelkerke
## 1 0.265628 0.2716663  0.3898735
```

```r
str(predicted.train)
```

```
##  Named num [1:750] 1 1 1 1 1 1 1 1 1 1 ...
##  - attr(*, "names")= chr [1:750] "2" "4" "5" "7" ...
```

```r
predicted.train <- as.numeric(predicted.train)

pred <- prediction(predicted.train, observed.train)
perf <- performance(pred, "tpr", "fpr")
plot(perf, colorize = TRUE)
```



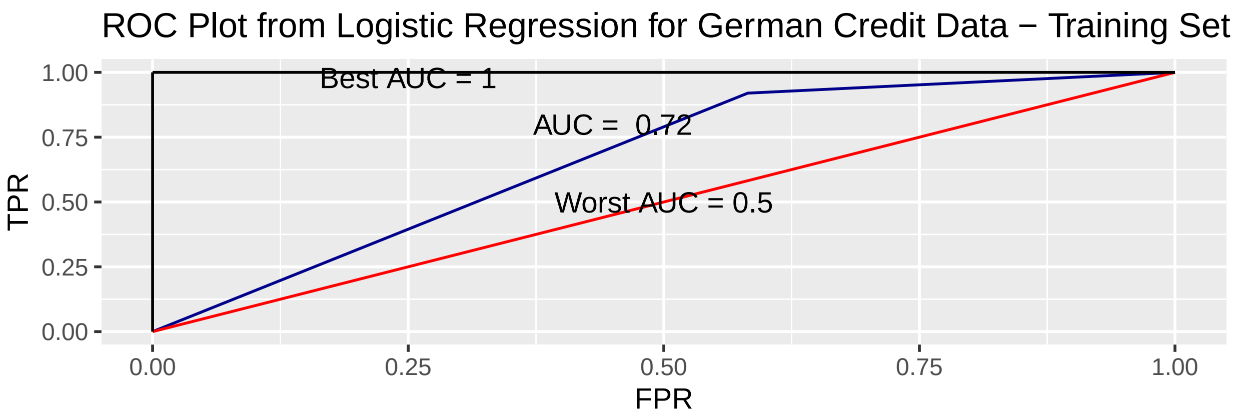

```
#calculate AUC
roc_obj <- roc(predicted.train, observed.train)
```

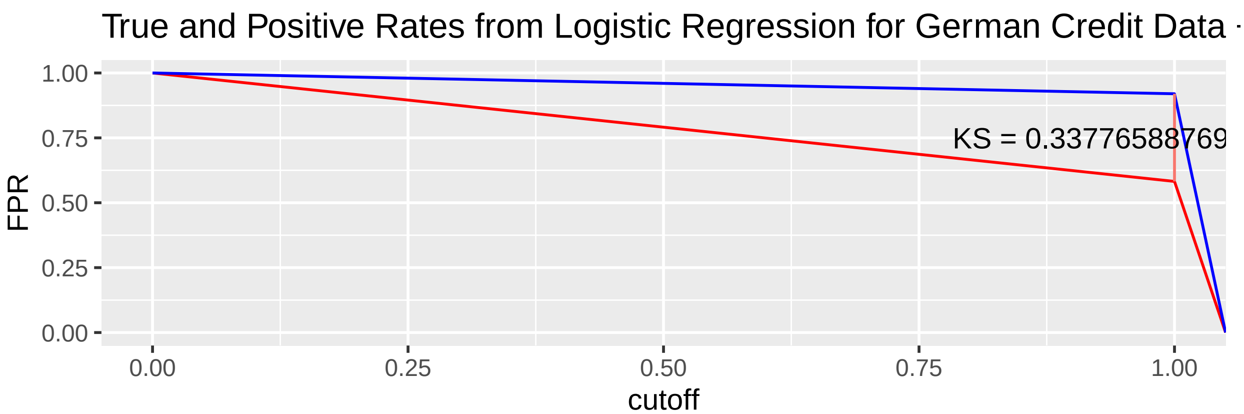
```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
AUC.train <- auc(roc_obj)
GINI.train <- 2 * AUC.train - 1

# ROC curve in ggplot2
DF.PR <- cbind.data.frame(perf@x.values[[1]], perf@y.values[[1]], perf@alpha.values[[1]])
colnames(DF.PR) <- c("FPR", "TPR", "cutoff")

#to add the 45 degree line to the plot
x <- c(0, 1)
y <- c(0, 1)
df2 <- cbind.data.frame(x, y)

# to add the AUC to the plot
x1 <- c(0, 1)
y1 <- c(1, 1)
df3 <- cbind.data.frame(x1, y1)

pROC.train <- ggplot() +
  geom_line(data = DF.PR, aes(x = FPR, y = TPR), color = "darkblue") +
  geom_line(data = df2, aes(x = x, y = y), color = "red") +
  geom_line(data = df3, aes(x = x1, y = y1), color = "black") +
  geom_segment(aes(x = 0, y = 0,xend = 0, yend = 1)) +
  annotate("text", x = 0.45, y = 0.80, label = "AUC =  0.72") +
  annotate("text", x = 0.25, y = 0.98, label = "Best AUC = 1") +
  annotate("text", x = 0.5, y = 0.5, label = "Worst AUC = 0.5") +
  ggtitle("ROC Plot from Logistic Regression for German Credit Data - Training Set")
```

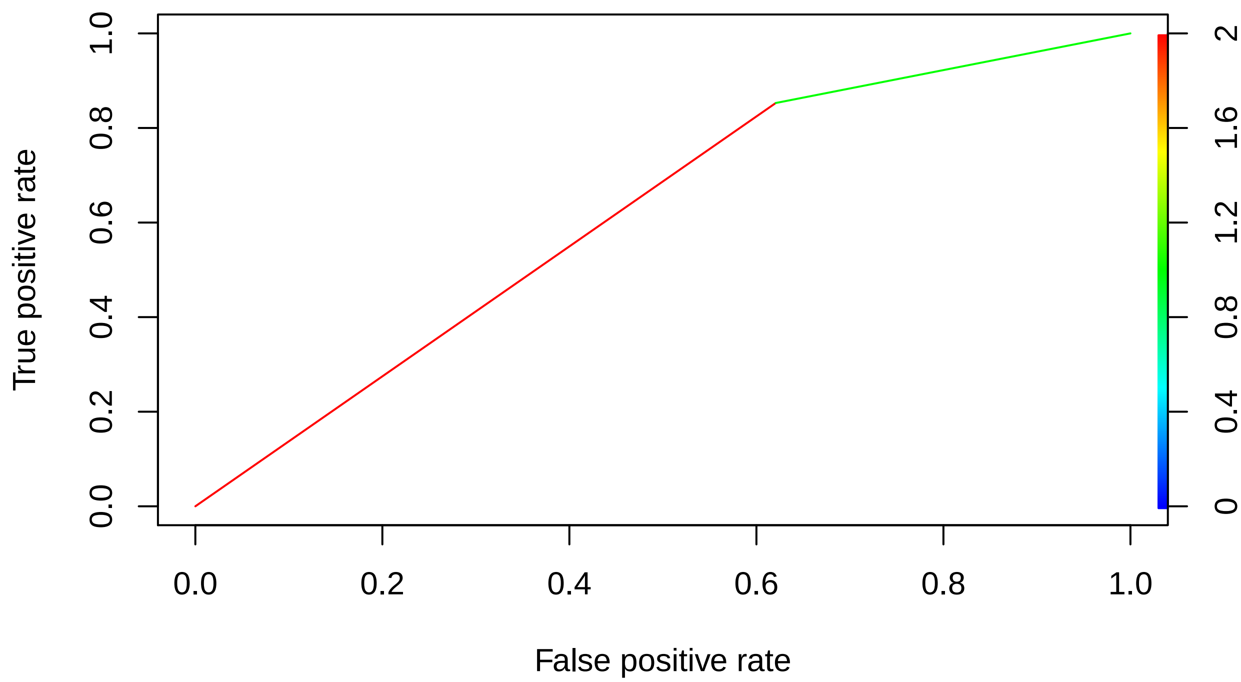
```r
## Kolmogorov-Smirnov Statistics (Performance Measure for Binary Classifiers)
## This is not the same as Kolmogorov-Smirnov Test Statistics for testing normality (of residuals in ML
# KS = maximum(TPR-FPR)

pK1 <- ggplot() +
       geom_line(data = DF.PR, aes(x = cutoff, y = FPR), color = "red") +
       geom_line(data = DF.PR, aes(x = cutoff, y = TPR), color = "blue")

DF.PR$diff <- DF.PR$TPR - DF.PR$FPR
KS.train <- max(DF.PR$diff)
i.m <- which.max(DF.PR$diff)
xM <- DF.PR$cutoff[i.m]
yML <- DF.PR$FPR[i.m]
yMU <- DF.PR$TPR[i.m]

pKS.train <- pK1 +
            geom_segment(aes(x = xM, y = yML, xend = xM, yend = yMU, colour = "black")) +
            annotate("text", x = 0.95, y = 0.75, label = paste0("KS = ", KS.train)) +
            theme(legend.position = "none") +
            ggtitle("True and Positive Rates from Logistic Regression for German Credit Data - Trainin

grid.arrange(pROC.train, pKS.train, nrow = 2)
```






```r
observed.test <- G.test$Creditability
predicted.test <- predict(LR2, G.test,type='response')
predicted.test <- round(predicted.test)
```

```r
#confusion matrix for Test set
CM.Test <- table(observed.test,predicted.test)
OA.Test <- sum(diag(CM.Test))/sum(CM.Test) # 0.7471264

#Precision, Recall, F1 for Test Data - Category 1
Recall.F <- CM.Test[2,2]/(CM.Test[2,1]+CM.Test[2,2])
Precision.F <- CM.Test[2,2]/(CM.Test[1,2]+CM.Test[2,2])
F1.F <- 2/((1/Recall.F)+(1/Precision.F))

#Precision, Recall, F1 for Test Data - Category 0
Recall.F0 <- CM.Test[1,1]/(CM.Test[1,1]+CM.Test[1,2])
Precision.F0 <- CM.Test[1,1]/(CM.Test[1,1]+CM.Test[2,1])
F1.F0 <- 2/((1/Recall.F0)+(1/Precision.F0))

PRF1_test_0 <- c(Precision.F0, Recall.F0, F1.F0)

# ROC Curve test set
pred <- prediction(predicted.test, observed.test)
perf <- performance(pred, "tpr", "fpr")
plot(perf, colorize = TRUE)
```



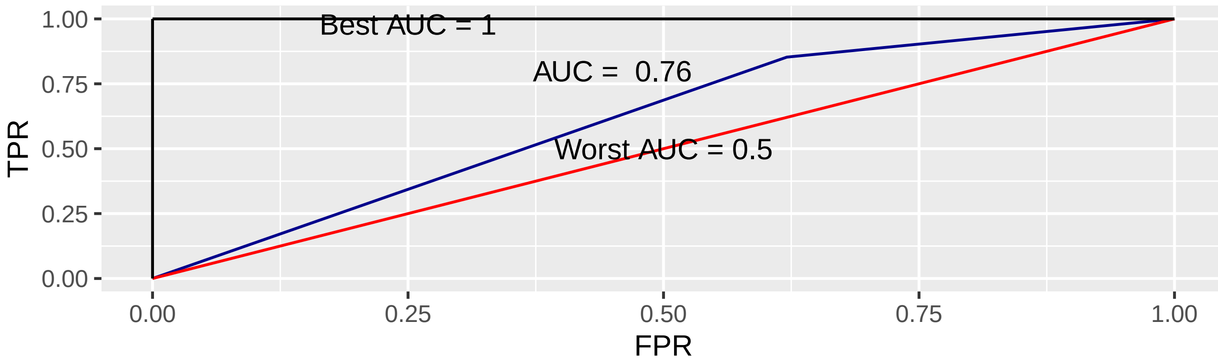

```r
#calculate AUC
roc_obj <- roc(predicted.test, observed.test)
```

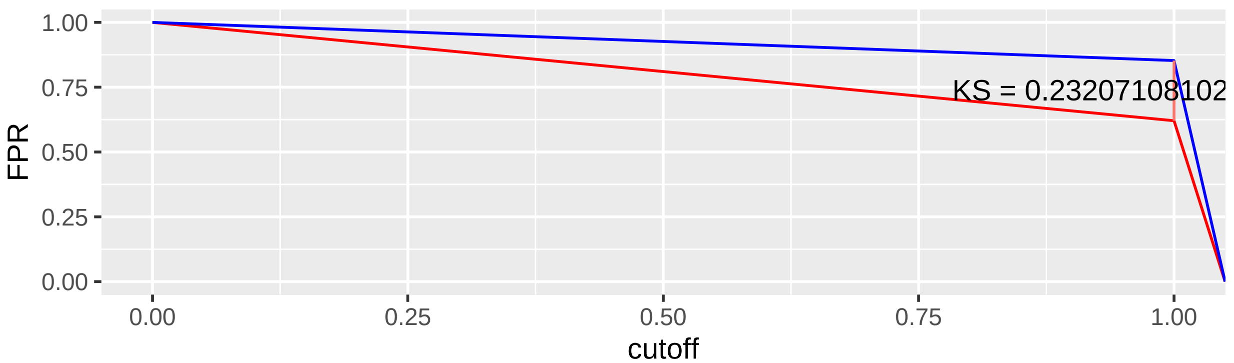
```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```r
AUC.test <- auc(roc_obj)
GINI.test <- 2 * AUC.test - 1

# OC curve in ggplot2
DF.PR <- cbind.data.frame(perf@x.values[[1]], perf@y.values[[1]], perf@alpha.values[[1]])
```

```r
colnames(DF.PR) <- c("FPR", "TPR", "cutoff")

# to add the 45 degree line to the plot
x <- c(0,1)
y <- c(0,1)
df2 <- cbind.data.frame(x,y)

#to add the AUC to the plot
x1 <- c(0,1)
y1 <- c(1,1)
df3 <- cbind.data.frame(x1,y1)

pROC.test <- ggplot() +
             geom_line(data = DF.PR, aes(x = FPR, y = TPR), color = "darkblue") +
             geom_line(data = df2, aes(x = x, y = y), color = "red") +
             geom_line(data = df3, aes(x = x1, y = y1), color = "black") +
             geom_segment(aes(x = 0, y = 0, xend = 0, yend = 1)) +
             annotate("text", x = 0.45, y = 0.80, label = "AUC =  0.76") +
             annotate("text", x = 0.25, y = 0.98, label = "Best AUC = 1") +
             annotate("text", x = 0.5, y = 0.5, label = "Worst AUC = 0.5") +
             ggtitle("ROC Plot from Logistic Regression for German Credit Data - Test Set")

#KS (Kolomogorov-Smirnov) Statistic
#KS = maximum(TPR-FPR)
pK1 <- ggplot() +
       geom_line(data = DF.PR, aes(x = cutoff, y = FPR), color = "red") +
       geom_line(data = DF.PR, aes(x = cutoff, y = TPR), color = "blue")

DF.PR$diff <- DF.PR$TPR - DF.PR$FPR
KS.test <- max(DF.PR$diff)
i.m <- which.max(DF.PR$diff)
xM <- DF.PR$cutoff[i.m]
yML <- DF.PR$FPR[i.m]
yMU <- DF.PR$TPR[i.m]

pKS.test <- pK1 +
       geom_segment(aes(x = xM, y = yML, xend = xM, yend = yMU, colour = "black")) +
       annotate("text", x = 0.95, y = 0.74, label = paste0("KS = ", KS.test)) +
       theme(legend.position = "none")+
       ggtitle("True and Positive Rates from Logistic Regression for German Credit Data - Test Set")

grid.arrange(pROC.test, pKS.test, nrow = 2)
```






```
OA <- c(OA.train, OA.Test)
names(OA) <- c("Overall accuracy_training", "Overall accuracy_test")

names(PRF1_train_1) <- c("Precision_train_1", "Recall_train_1", "F1_train_1")
names(PRF1_train_0) <- c("Precision_train_0", "Recall_train_0", "F1_train_0")
# names(PRF1_test_1) <- c("Precision_test_1", "Recall_test_1", "F1_test_1")
# names(PRF1_test_0) <- c("Precision_test_0", "Recall_test_0", "F1_test_0")

AUC  <- c(AUC.train, AUC.test)
GINI <- c(GINI.train, GINI.test)
names(AUC) <- c("AUC_train", "AUC_test")
names(GINI) <- c("GINI_train", "GINI_test")

# print performance results for both training and test sets
print("Logistic Regression Summary of Results for German Credit Data")
```

```
## [1] "Logistic Regression Summary of Results for German Credit Data"
```

```
print(PRF1_train_1)
```

```
## Precision_train_1    Recall_train_1        F1_train_1
##         0.7993528         0.9199255         0.8554113
```

```
print(PRF1_train_0)
```

```
## Precision_train_0    Recall_train_0        F1_train_0
##         0.6742424         0.4178404         0.5159420
```

```r
# print(PRF1_test_1)
# print(PRF1_test_0)

print(OA)
```

```
## Overall accuracy_training     Overall accuracy_test
##                 0.7773333                 0.6880000
```

```r
print(AUC)
```

```
## AUC_train  AUC_test
## 0.7367976 0.6495773
```

```r
print(GINI)
```

```
## GINI_train  GINI_test
##  0.4735952  0.2991546
```

Charles Book Club

```r
G <- read.csv("Charles_BookClub.csv", header = TRUE)
```

```r
head(G)
```

```
##   Seq. ID. Gender   M  R F FirstPurch ChildBks YouthBks CookBks DoltYBks
## 1    1   2      0 138 28 3         40        0        1       0        1
## 2    2  30      1 240 14 1         14        1        0       0        0
## 3    3  59      1  97  6 2         10        0        0       0        0
## 4    4  89      1 348  2 7         38        1        1       1        0
## 5    5  96      0 239 20 2         28        0        0       1        0
## 6    6 120      1 253 10 4         20        1        0       0        0
##   RefBks ArtBks GeogBks ItalCook ItalHAtlas ItalArt Florence
## 1      0      0       1        0          0       0        0
## 2      0      0       0        0          0       0        0
## 3      0      0       0        0          0       0        0
## 4      1      0       1        0          0       0        0
## 5      0      0       1        0          0       0        0
## 6      0      1       0        0          0       0        1
```

```r
tail(G)
```

```
##      Seq.   ID. Gender   M  R F FirstPurch ChildBks YouthBks CookBks
## 1995 1995 49781      1 192  8 1          8        0        0       0
## 1996 1996 49801      1 164 12 5         32        0        0       1
## 1997 1997 49866      0 294 10 1         10        0        0       0
## 1998 1998 49872      0 261  4 2         10        0        0       0
## 1999 1999 49914      1  41 32 1         32        0        0       1
## 2000 2000 49962      1 308 12 1         12        0        0       0
##      DoltYBks RefBks ArtBks GeogBks ItalCook ItalHAtlas ItalArt Florence
## 1995        0      0      0       0        0          0       0        0
## 1996        0      0      1       2        1          0       1        1
## 1997        0      0      0       0        0          0       0        0
## 1998        0      0      0       0        0          0       0        0
## 1999        0      0      0       0        0          0       0        0
## 2000        0      0      0       0        0          0       0        0
```

```r
dim(G)
```

```
## [1] 2000   18
```

```r
names(G)
```

```
##  [1] "Seq."       "ID."        "Gender"     "M"          "R"
##  [6] "F"          "FirstPurch" "ChildBks"   "YouthBks"   "CookBks"
## [11] "DoltYBks"   "RefBks"     "ArtBks"     "GeogBks"    "ItalCook"
## [16] "ItalHAtlas" "ItalArt"    "Florence"
```

```r
summary(G)
```

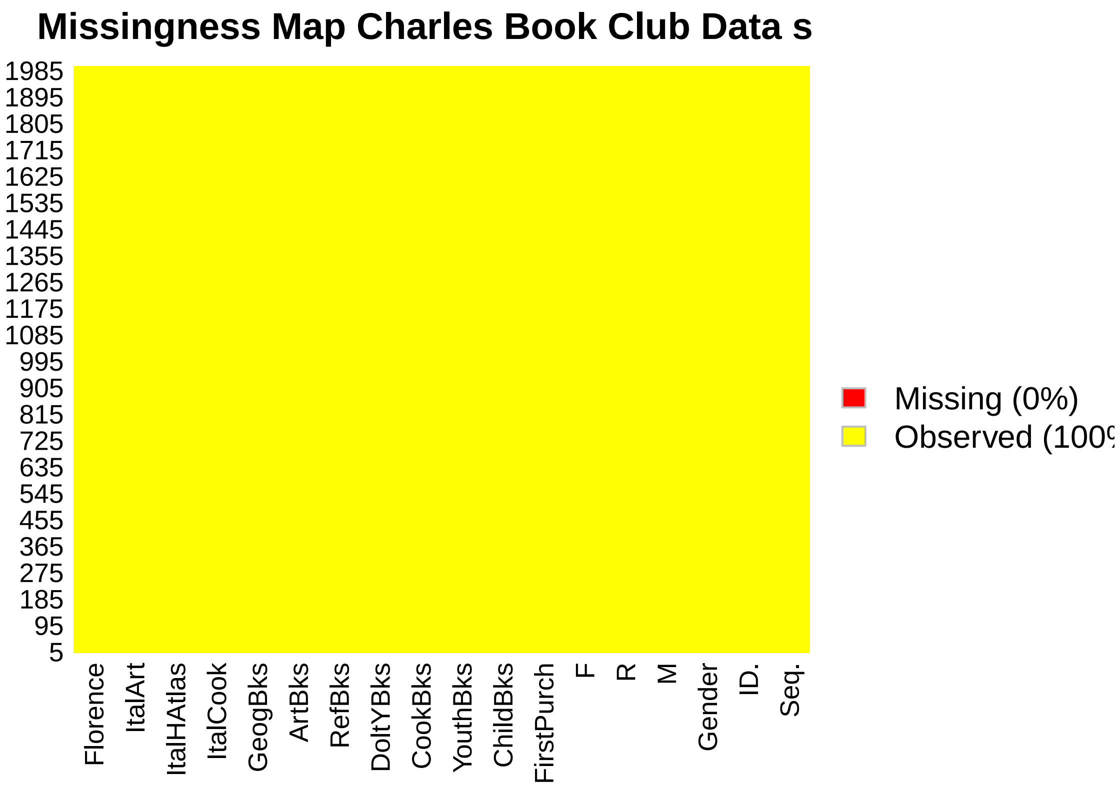
```
##       Seq.              ID.            Gender             M
##  Min.   :   1.0   Min.   :    2   Min.   :0.0000   Min.   : 15.0
##  1st Qu.: 500.8   1st Qu.:12699   1st Qu.:0.0000   1st Qu.:126.8
##  Median :1000.5   Median :24201   Median :1.0000   Median :207.0
##  Mean   :1000.5   Mean   :24753   Mean   :0.7085   Mean   :206.8
##  3rd Qu.:1500.2   3rd Qu.:37300   3rd Qu.:1.0000   3rd Qu.:281.2
##  Max.   :2000.0   Max.   :49962   Max.   :1.0000   Max.   :477.0
##        R               F            FirstPurch       ChildBks
##  Min.   : 2.00   Min.   : 1.000   Min.   : 2.00   Min.   :0.000
##  1st Qu.: 8.00   1st Qu.: 1.000   1st Qu.:14.00   1st Qu.:0.000
##  Median :12.00   Median : 2.000   Median :22.00   Median :0.000
##  Mean   :13.52   Mean   : 4.005   Mean   :27.42   Mean   :0.711
##  3rd Qu.:16.00   3rd Qu.: 6.000   3rd Qu.:38.00   3rd Qu.:1.000
##  Max.   :36.00   Max.   :12.000   Max.   :99.00   Max.   :6.000
##     YouthBks        CookBks          DoltYBks         RefBks
##  Min.   :0.000   Min.   :0.0000   Min.   :0.000   Min.   :0.0000
##  1st Qu.:0.000   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:0.0000
##  Median :0.000   Median :0.0000   Median :0.000   Median :0.0000
##  Mean   :0.314   Mean   :0.7385   Mean   :0.391   Mean   :0.2705
##  3rd Qu.:0.000   3rd Qu.:1.0000   3rd Qu.:1.000   3rd Qu.:0.0000
##  Max.   :5.000   Max.   :8.0000   Max.   :5.000   Max.   :4.0000
##      ArtBks          GeogBks          ItalCook        ItalHAtlas
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.0000   Median :0.0000   Median :0.0000   Median :0.0000
##  Mean   :0.3145   Mean   :0.4115   Mean   :0.1285   Mean   :0.0395
##  3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:0.0000
##  Max.   :5.0000   Max.   :5.0000   Max.   :2.0000   Max.   :2.0000
##     ItalArt         Florence
##  Min.   :0.000   Min.   :0.0000
##  1st Qu.:0.000   1st Qu.:0.0000
##  Median :0.000   Median :0.0000
##  Mean   :0.052   Mean   :0.1085
##  3rd Qu.:0.000   3rd Qu.:0.0000
##  Max.   :2.000   Max.   :1.0000
```

```r
## Check for NA's in the data
sapply(G, function(x) sum(is.na(x)))
```

```
##       Seq.        ID.     Gender          M          R          F
##          0          0          0          0          0          0
## FirstPurch   ChildBks   YouthBks    CookBks   DoltYBks     RefBks
##          0          0          0          0          0          0
```

```
##     ArtBks     GeogBks    ItalCook ItalHAtlas     ItalArt    Florence
##          0           0           0           0           0           0
```

```
missmap(G, col = c("red", "yellow"), main = "Missingness Map Charles Book Club Data set")
```




```
M <- .25 * nrow(G)
## To be able to replicate the results,
## set initial seed for random number generator
set.seed(117317)
holdout <- sample(1:nrow(G), M, replace = F)

G.train <- G[-holdout, ] ## Training set
G.test <- G[holdout, ] ## Test set
dim(G.train) ## 1500 18
```

```
## [1] 1500   18
```

```
dim(G.test) ## 500 18
```

```
## [1] 500  18
```

```
names(G.train)
```

```
##  [1] "Seq."       "ID."        "Gender"     "M"          "R"
##  [6] "F"          "FirstPurch" "ChildBks"   "YouthBks"   "CookBks"
## [11] "DoltYBks"   "RefBks"     "ArtBks"     "GeogBks"    "ItalCook"
## [16] "ItalHAtlas" "ItalArt"    "Florence"
```

```
# LR1 <- glm(Florence ~ ., family = binomial("logit"), data = G.train)
LR1 <- glm(Florence ~ Seq. + ID. + Gender + M + R + F + FirstPurch + ChildBks + YouthBks + CookBks + Do
smre1 <- summary(LR1)
smre1 ## As long as at least one category of a variable is significant, keep the variable
```

```
##
## Call:
## glm(formula = Florence ~ Seq. + ID. + Gender + M + R + F + FirstPurch +
##     ChildBks + YouthBks + CookBks + DoItYBks + RefBks + ArtBks +
##     GeogBks + ItalCook + ItalHAtlas + ItalArt, family = binomial("logit"),
##     data = G.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8113  -0.4842  -0.3323  -0.1977   2.8681
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.5114266  0.3991748  -3.786 0.000153 ***
## Seq.         0.0069440  0.0095583   0.726 0.467542
## ID.         -0.0002789  0.0003825  -0.729 0.465974
## Gender      -0.8735560  0.1846272  -4.731 2.23e-06 ***
## M            0.0006768  0.0010484   0.646 0.518557
## R           -0.0849292  0.0210404  -4.036 5.43e-05 ***
## F            0.3149815  0.0990338   3.181 0.001470 **
## FirstPurch   0.0108011  0.0129893   0.832 0.405670
## ChildBks    -0.5562163  0.1548258  -3.593 0.000327 ***
## YouthBks    -0.5658717  0.2031475  -2.786 0.005344 **
## CookBks     -0.6414943  0.1648676  -3.891 9.98e-05 ***
## DoItYBks    -0.7295431  0.1935028  -3.770 0.000163 ***
## RefBks      -0.0507961  0.2126410  -0.239 0.811197
## ArtBks       0.5639761  0.1724793   3.270 0.001076 **
## GeogBks      0.1384277  0.1663847   0.832 0.405424
## ItalCook     0.0875561  0.2983574   0.293 0.769170
## ItalHAtlas   0.2418714  0.4677487   0.517 0.605088
## ItalArt      0.6852723  0.3780944   1.812 0.069919 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1052.03  on 1499  degrees of freedom
## Residual deviance:  857.32  on 1482  degrees of freedom
## AIC: 893.32
##
## Number of Fisher Scoring iterations: 6
```

```
      ## Need to keep all the coefficients for a significant categorical variable
## This comes from the car package
vif1 <- vif(LR1)
min(vif1)
```

```
## [1] 1.029525
```

```
max(vif1) ## No multicollinearity since the max VIF is 4, which is less than 5
```

```
## [1] 3756.699
```

```
## Use this to predict training set and also the test set
LR2 <- glm(Florence ~ . -Seq. -ID. -M -ItalCook -RefBks -GeogBks -FirstPurch -ItalHAtlas, family = bino
```

```r
# LR2 <- glm(Florence ~ Gender + R + F + ChildBks + YouthBks + CookBks + DoltYBks + ArtBks + ItalArt, f

smre2 <- summary(LR2)
smre2
```

```
##
## Call:
## glm(formula = Florence ~ . - Seq. - ID. - M - ItalCook - RefBks -
##     GeogBks - FirstPurch - ItalHAtlas, family = binomial("logit"),
##     data = G.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7830  -0.4828  -0.3332  -0.2017   2.8625
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.47851    0.26302  -5.621 1.90e-08 ***
## Gender      -0.86896    0.18361  -4.733 2.22e-06 ***
## R           -0.06934    0.01645  -4.215 2.50e-05 ***
## F            0.39544    0.06808   5.808 6.32e-09 ***
## ChildBks    -0.54988    0.14245  -3.860 0.000113 ***
## YouthBks    -0.56334    0.19317  -2.916 0.003543 **
## CookBks     -0.63896    0.14304  -4.467 7.93e-06 ***
## DoltYBks    -0.74587    0.17856  -4.177 2.95e-05 ***
## ArtBks       0.53063    0.15289   3.471 0.000519 ***
## ItalArt      0.77201    0.30184   2.558 0.010539 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1052.03  on 1499  degrees of freedom
## Residual deviance:  860.55  on 1490  degrees of freedom
## AIC: 880.55
##
## Number of Fisher Scoring iterations: 6
```

```r
vif2 <- vif(LR2)
max(vif2)
```

```
## [1] 8.11661
```

```r
# write.csv(smre1$coefficients, "Final LR Model for Charles Book Club Data.csv")
confint(LR2)
```

```
## Waiting for profiling to be done...
```

```
##                  2.5 %      97.5 %
## (Intercept) -1.9969367 -0.96482647
## Gender      -1.2293630 -0.50849317
## R           -0.1027597 -0.03821673
## F            0.2624852  0.52979920
## ChildBks    -0.8345154 -0.27524048
## YouthBks    -0.9545385 -0.19591292
## CookBks     -0.9268042 -0.36543627
```

```
## DoltYBks    -1.1087753 -0.40783580
## ArtBks       0.2315499  0.83347087
## ItalArt      0.1710879  1.36062325
```

```r
# #Predict training data using the model LR1
observed.train <- G.train$Florence
predicted.train <- predict(LR2, G.train, type = 'response')
## predict.train consists of P(Y=1) for each observation in the training set
predicted.train <- round(predicted.train) ## Round to 0 or 1 to get the Y values

## Evaluate Performance of the LR Classifier on the training set
## Confusion Matrix of observed versus predicted Y values
CM.train <- table(observed.train, predicted.train)
CM.train
```

```
##                predicted.train
## observed.train    0    1
##              0 1319   13
##              1  149   19
```

```r
FP <- CM.train[1,2]/(CM.train[1,1]+CM.train[1,2]) ## false positive
FN <- CM.train[2,1]/(CM.train[2,1]+CM.train[2,2]) ## false negative
FP
```

```
## [1] 0.00975976
```

```r
FN
```

```
## [1] 0.8869048
```

```r
## Overall accuracy
## Calculated by summing the correct predictions divided by the total of the Confusion Matrix
OA.train <- sum(diag(CM.train)) / sum(CM.train)
OA.train
```

```
## [1] 0.892
```

```r
## Precision, Recall, F1-Measure are performance measures for binary prediction
## F1-measure = geometric mean of the Precision and Recall

## Compute Precision, Recall, F1 for Category 1, model LR1
Recall1 <- CM.train[2,2] / (CM.train[2,1] + CM.train[2,2]) ## diag/row sum
Precision1 <- CM.train[2,2] / (CM.train[1,2] + CM.train[2,2]) ## diag/column sum
F1.1 <- 2 / ((1 / Recall1) + (1 / Precision1)) ## The geometric mean

PRF1_train_1 <- (c(Precision1, Recall1, F1.1))
PRF1_train_1
```

```
## [1] 0.5937500 0.1130952 0.1900000
```

```r
# Category 0
## Repeat formulas, but use different positions in the Confusion Matrix
Recall0 <- CM.train[1,1] / (CM.train[1,1] + CM.train[1,2])
Precision0 <- CM.train[1,1] / (CM.train[1,1] + CM.train[2,1])
F1.0 <- 2 / ((1 / Recall0) + (1 / Precision0))

PRF1_train_0 <- c(Precision0, Recall0, F1.0)
PRF1_train_0
```

```
## [1] 0.8985014 0.9902402 0.9421429
```

```
## Why is the LR2 model doing a better job of predicting 1's and so-so job for category 0?
## Because there are so many more 1's than 0's. This is an example of an unbalanced data set.
## There are methods for dealing with unbalanced data sets.
```

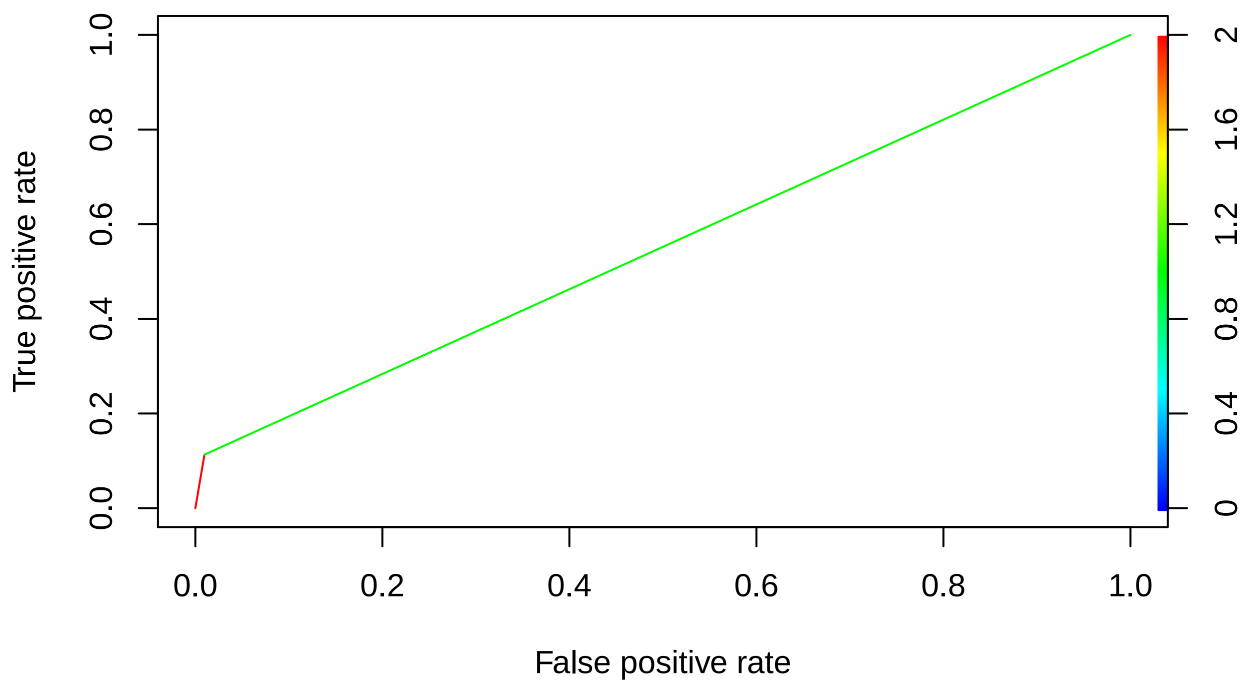
```
r2fun(LR1)
```

```
##   McFadden  CoxSnell Nagelkerke
## 1 0.185076 0.1217321  0.2414907
```

```
str(predicted.train)
```

```
##  Named num [1:1500] 0 0 0 0 0 0 0 0 0 0 ...
##  - attr(*, "names")= chr [1:1500] "2" "3" "4" "5" ...
```

```
predicted.train <- as.numeric(predicted.train)

pred <- prediction(predicted.train, observed.train)
perf <- performance(pred, "tpr", "fpr")
plot(perf, colorize = TRUE)
```




```
#calculate AUC
roc_obj <- roc(predicted.train, observed.train)
```

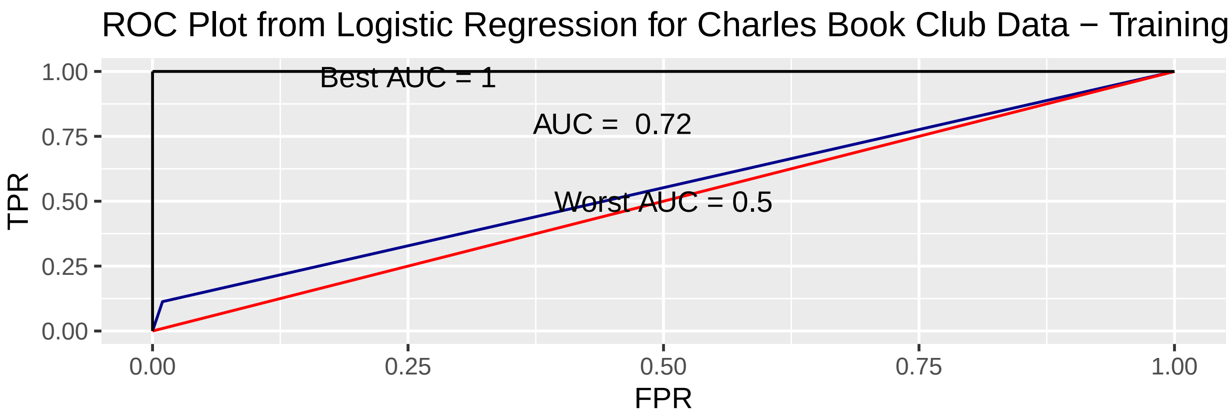
```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

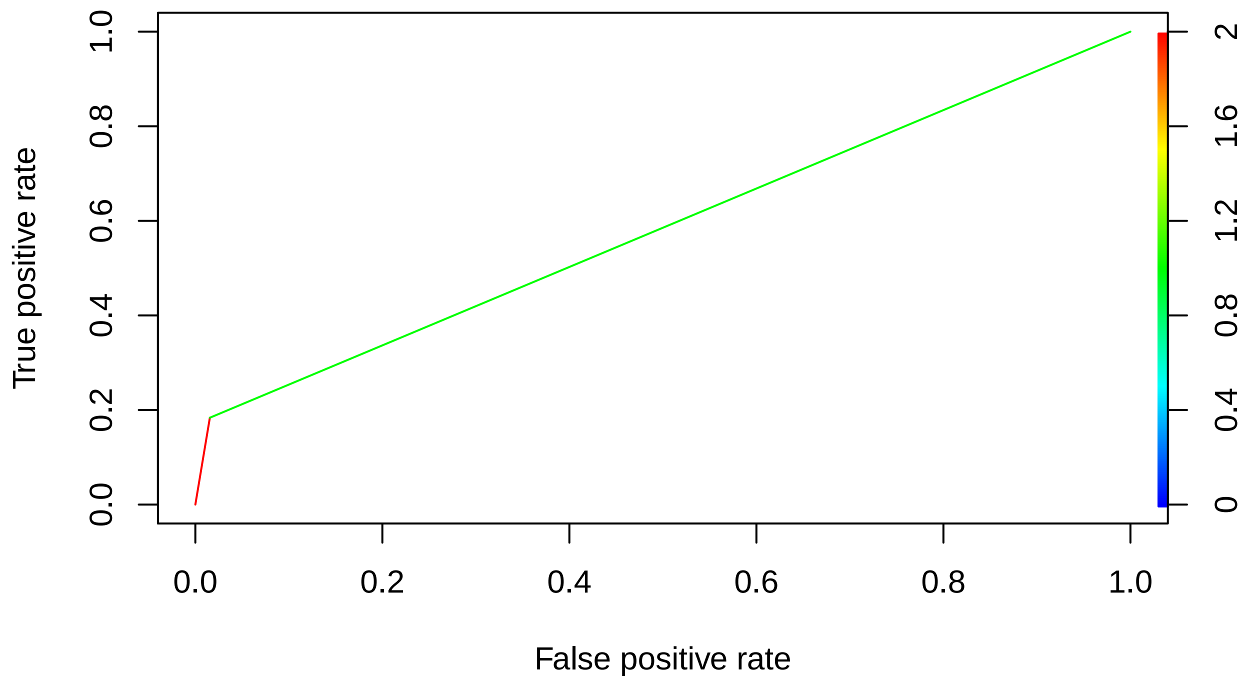
```
AUC.train <- auc(roc_obj)
GINI.train <- 2 * AUC.train - 1

# ROC curve in ggplot2
DF.PR <- cbind.data.frame(perf@x.values[[1]], perf@y.values[[1]], perf@alpha.values[[1]])
colnames(DF.PR) <- c("FPR", "TPR", "cutoff")

#to add the 45 degree line to the plot
```

```r
x <- c(0, 1)
y <- c(0, 1)
df2 <- cbind.data.frame(x, y)

# to add the AUC to the plot
x1 <- c(0, 1)
y1 <- c(1, 1)
df3 <- cbind.data.frame(x1, y1)

pROC.train <- ggplot() +
  geom_line(data = DF.PR, aes(x = FPR, y = TPR), color = "darkblue") +
  geom_line(data = df2, aes(x = x, y = y), color = "red") +
  geom_line(data = df3, aes(x = x1, y = y1), color = "black") +
  geom_segment(aes(x = 0, y = 0,xend = 0, yend = 1)) +
  annotate("text", x = 0.45, y = 0.80, label = "AUC =  0.72") +
  annotate("text", x = 0.25, y = 0.98, label = "Best AUC = 1") +
  annotate("text", x = 0.5, y = 0.5, label = "Worst AUC = 0.5") +
  ggtitle("ROC Plot from Logistic Regression for Charles Book Club Data - Training Set")
```

```r
## Kolmogorov-Smirnov Statistics (Performance Measure for Binary Classifiers)
## This is not the same as Kolmogorov-Smirnov Test Statistics for testing normality (of residuals in MLR
# KS = maximum(TPR-FPR)

pK1 <- ggplot() +
       geom_line(data = DF.PR, aes(x = cutoff, y = FPR), color = "red") +
       geom_line(data = DF.PR, aes(x = cutoff, y = TPR), color = "blue")

DF.PR$diff <- DF.PR$TPR - DF.PR$FPR
KS.train <- max(DF.PR$diff)
i.m <- which.max(DF.PR$diff)
xM <- DF.PR$cutoff[i.m]
yML <- DF.PR$FPR[i.m]
yMU <- DF.PR$TPR[i.m]

pKS.train <- pK1 +
             geom_segment(aes(x = xM, y = yML, xend = xM, yend = yMU, colour = "black")) +
             annotate("text", x = 0.95, y = 0.75, label = paste0("KS = ", KS.train)) +
             theme(legend.position = "none") +
             ggtitle("True and Positive Rates from Logistic Regression for Charles Book Club Data - Trai

grid.arrange(pROC.train, pKS.train, nrow = 2)
```




```
observed.test <- G.test$Florence
predicted.test <- predict(LR2, G.test,type='response')
predicted.test <- round(predicted.test)

#confusion matrix for Test set
CM.Test <- table(observed.test,predicted.test)
OA.Test <- sum(diag(CM.Test))/sum(CM.Test) # 0.7471264

#Precision, Recall, F1 for Test Data - Category 1
Recall.F <- CM.Test[2,2]/(CM.Test[2,1]+CM.Test[2,2])
Precision.F <- CM.Test[2,2]/(CM.Test[1,2]+CM.Test[2,2])
F1.F <- 2/((1/Recall.F)+(1/Precision.F))

#Precision, Recall, F1 for Test Data - Category 0
Recall.F0 <- CM.Test[1,1]/(CM.Test[1,1]+CM.Test[1,2])
Precision.F0 <- CM.Test[1,1]/(CM.Test[1,1]+CM.Test[2,1])
F1.F0 <- 2/((1/Recall.F0)+(1/Precision.F0))

PRF1_test_0 <- c(Precision.F0, Recall.F0, F1.F0)

# ROC Curve test set
pred <- prediction(predicted.test, observed.test)
perf <- performance(pred, "tpr", "fpr")
plot(perf, colorize = TRUE)
```



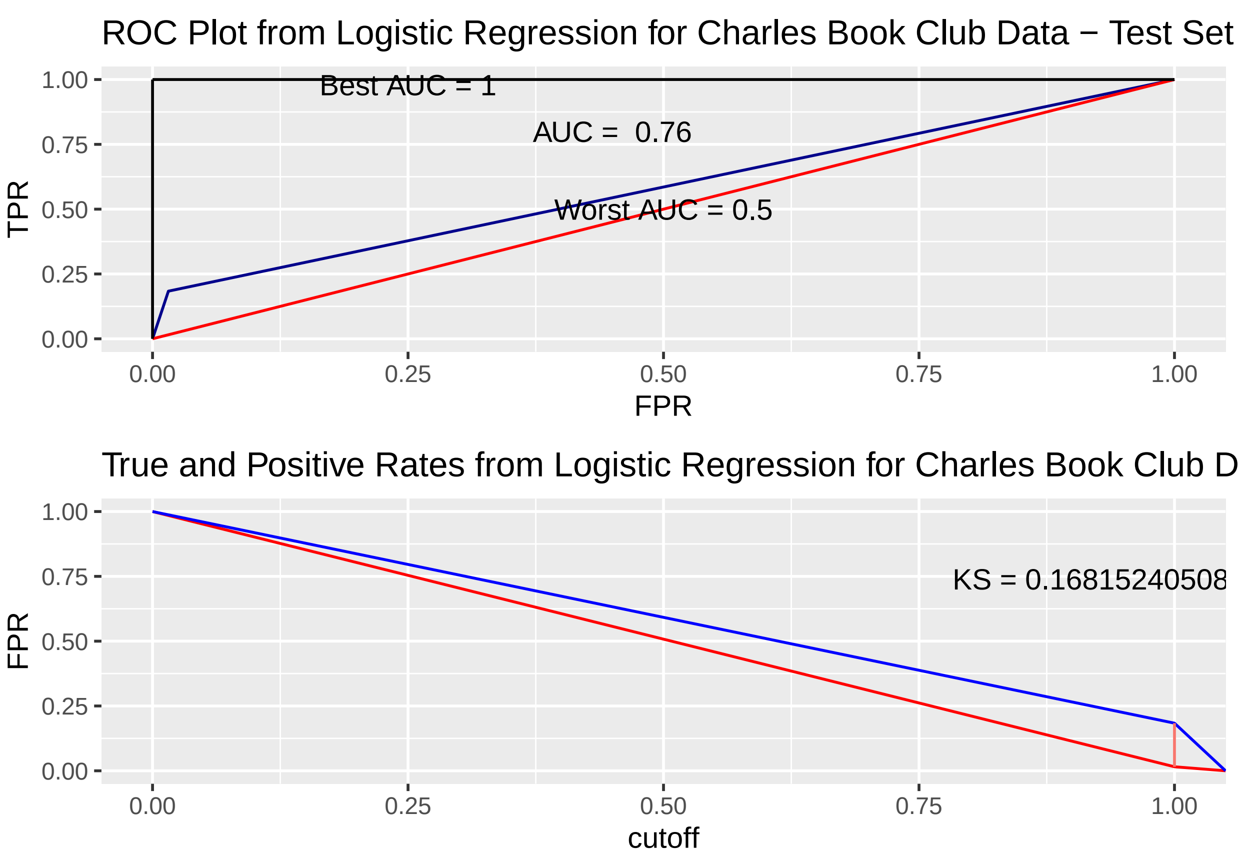

```
#calculate AUC
roc_obj <- roc(predicted.test, observed.test)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
AUC.test <- auc(roc_obj)
GINI.test <- 2 * AUC.test - 1

# OC curve in ggplot2
DF.PR <- cbind.data.frame(perf@x.values[[1]], perf@y.values[[1]], perf@alpha.values[[1]])
colnames(DF.PR) <- c("FPR", "TPR", "cutoff")

# to add the 45 degree line to the plot
x <- c(0,1)
y <- c(0,1)
df2 <- cbind.data.frame(x,y)

#to add the AUC to the plot
x1 <- c(0,1)
y1 <- c(1,1)
df3 <- cbind.data.frame(x1,y1)

pROC.test <- ggplot() +
             geom_line(data = DF.PR, aes(x = FPR, y = TPR), color = "darkblue") +
             geom_line(data = df2, aes(x = x, y = y), color = "red") +
             geom_line(data = df3, aes(x = x1, y = y1), color = "black") +
             geom_segment(aes(x = 0, y = 0, xend = 0, yend = 1)) +
             annotate("text", x = 0.45, y = 0.80, label = "AUC =  0.76") +
             annotate("text", x = 0.25, y = 0.98, label = "Best AUC = 1") +
             annotate("text", x = 0.5, y = 0.5, label = "Worst AUC = 0.5") +
             ggtitle("ROC Plot from Logistic Regression for Charles Book Club Data - Test Set")
```

```
#KS (Kolomogorov-Smirnov) Statistic
#KS = maximum(TPR-FPR)
pK1 <- ggplot() +
       geom_line(data = DF.PR, aes(x = cutoff, y = FPR), color = "red") +
       geom_line(data = DF.PR, aes(x = cutoff, y = TPR), color = "blue")

DF.PR$diff <- DF.PR$TPR - DF.PR$FPR
KS.test <- max(DF.PR$diff)
i.m <- which.max(DF.PR$diff)
xM <- DF.PR$cutoff[i.m]
yML <- DF.PR$FPR[i.m]
yMU <- DF.PR$TPR[i.m]

pKS.test <- pK1 +
       geom_segment(aes(x = xM, y = yML, xend = xM, yend = yMU, colour = "black")) +
       annotate("text", x = 0.95, y = 0.74, label = paste0("KS = ", KS.test)) +
       theme(legend.position = "none")+
       ggtitle("True and Positive Rates from Logistic Regression for Charles Book Club Data - Test Set")

grid.arrange(pROC.test, pKS.test, nrow = 2)
```




```
OA <- c(OA.train, OA.Test)
names(OA) <- c("Overall accuracy_training", "Overall accuracy_test")

names(PRF1_train_1) <- c("Precision_train_1", "Recall_train_1", "F1_train_1")
names(PRF1_train_0) <- c("Precision_train_0", "Recall_train_0", "F1_train_0")
```

```r
# names(PRF1_test_1) <- c("Precision_test_1", "Recall_test_1", "F1_test_1")
# names(PRF1_test_0) <- c("Precision_test_0", "Recall_test_0", "F1_test_0")

AUC  <- c(AUC.train, AUC.test)
GINI <- c(GINI.train, GINI.test)
names(AUC) <- c("AUC_train", "AUC_test")
names(GINI) <- c("GINI_train", "GINI_test")

# print performance results for both training and test sets
print("Logistic Regression Summary of Results for Charles Book Club Data")
```

```
## [1] "Logistic Regression Summary of Results for Charles Book Club Data"
```

```r
print(PRF1_train_1)
```

```
## Precision_train_1    Recall_train_1        F1_train_1
##         0.5937500         0.1130952         0.1900000
```

```r
print(PRF1_train_0)
```

```
## Precision_train_0    Recall_train_0        F1_train_0
##         0.8985014         0.9902402         0.9421429
```

```r
# print(PRF1_test_1)
# print(PRF1_test_0)

print(OA)
```

```
## Overall accuracy_training     Overall accuracy_test
##                     0.892                     0.906
```

```r
print(AUC)
```

```
## AUC_train  AUC_test
## 0.7461257 0.7399277
```

```r
print(GINI)
```

```
## GINI_train  GINI_test
##  0.4922514  0.4798554
```

Titanic

```r
dt <- read.csv("titanic3.csv", header = TRUE) %>%
  select(survived, pclass, sex, age, sibsp, parch) %>%
  filter(!is.na(pclass) & !is.na(sex) & !is.na(age) & !is.na(sibsp) & !is.na(parch)) %>%
  mutate(survived = as.numeric(survived))
```

```r
head(dt)
```

```
##   survived pclass sex     age sibsp parch
## 1        1      1   1 29.0000     0     0
## 2        1      1   0  0.9167     1     2
## 3        0      1   1  2.0000     1     2
## 4        0      1   0 30.0000     1     2
## 5        0      1   1 25.0000     1     2
## 6        1      1   0 48.0000     0     0
```

```r
tail(dt)
```

```
##      survived pclass sex  age sibsp parch
## 1041        1      3   1 15.0     1     0
## 1042        0      3   0 45.5     0     0
## 1043        0      3   1 14.5     1     0
## 1044        0      3   0 26.5     0     0
## 1045        0      3   0 27.0     0     0
## 1046        0      3   0 29.0     0     0
```

```r
dim(dt)
```

```
## [1] 1046    6
```

```r
names(dt)
```

```
## [1] "survived" "pclass"   "sex"      "age"      "sibsp"    "parch"
```

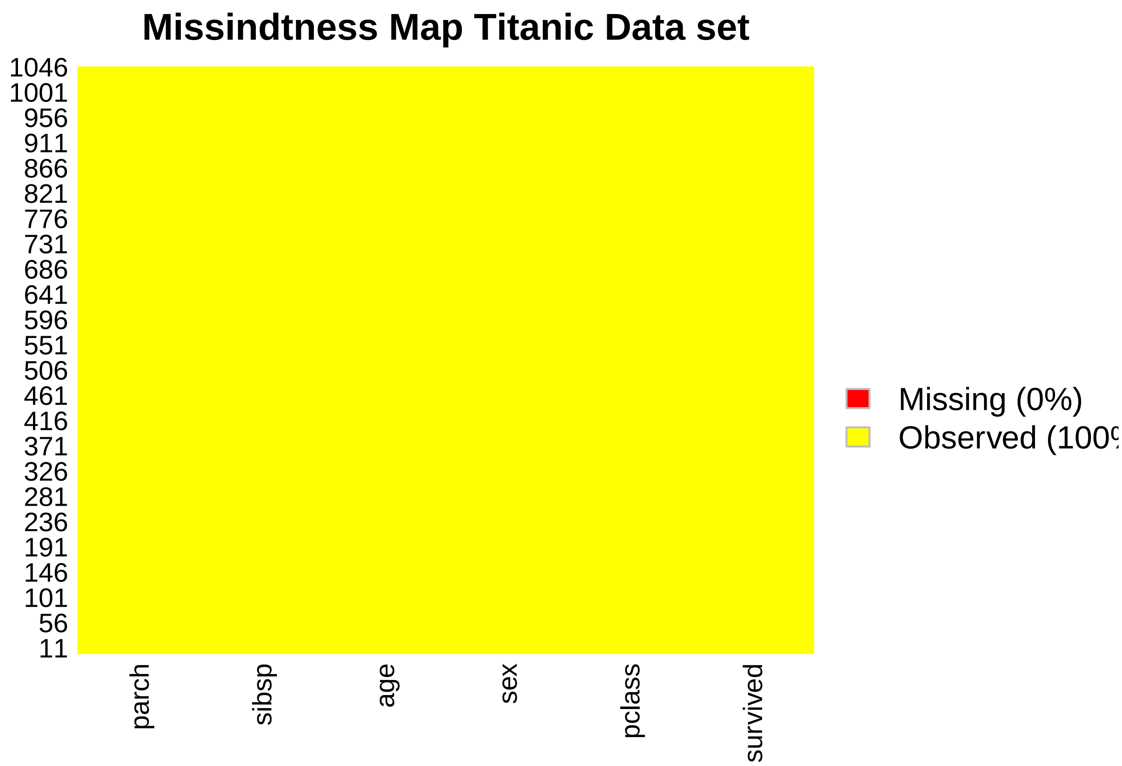
```r
summary(dt)
```

```
##     survived          pclass           sex              age         
##  Min.   :0.0000   Min.   :1.000   Min.   :0.0000   Min.   : 0.1667  
##  1st Qu.:0.0000   1st Qu.:1.000   1st Qu.:0.0000   1st Qu.:21.0000  
##  Median :0.0000   Median :2.000   Median :0.0000   Median :28.0000  
##  Mean   :0.4082   Mean   :2.207   Mean   :0.3709   Mean   :29.8811  
##  3rd Qu.:1.0000   3rd Qu.:3.000   3rd Qu.:1.0000   3rd Qu.:39.0000  
##  Max.   :1.0000   Max.   :3.000   Max.   :1.0000   Max.   :80.0000  
##      sibsp            parch       
##  Min.   :0.0000   Min.   :0.0000  
##  1st Qu.:0.0000   1st Qu.:0.0000  
##  Median :0.0000   Median :0.0000  
##  Mean   :0.5029   Mean   :0.4207  
##  3rd Qu.:1.0000   3rd Qu.:1.0000  
##  Max.   :8.0000   Max.   :6.0000
```

```r
## Check for NA's in the data
sapply(dt, function(x) sum(is.na(x)))
```

```
## survived   pclass      sex      age    sibsp    parch 
##        0        0        0        0        0        0
```

```r
missmap(dt, col = c("red", "yellow"), main = "Missindtness Map Titanic Data set")
```




```r
M <- .25 * nrow(dt)
## To be able to replicate the results,
## set initial seed for random number generator
set.seed(117317)
holdout <- sample(1:nrow(dt), M, replace = F)

dt.train <- dt[-holdout, ] ## Training set
dt.test <- dt[holdout, ] ## Test set
dim(dt.train) ## 982 14
```

```
## [1] 785   6
```

```r
dim(dt.test) ## 327 14
```

```
## [1] 261   6
```

```r
names(dt.train)
```

```
## [1] "survived" "pclass"   "sex"      "age"      "sibsp"    "parch"
```

```r
LR1 <- glm(survived ~ pclass + sex + age + sibsp + parch, family = binomial("logit"), data = dt.train)
smre1 <- summary(LR1)
smre1 ## As long as at least one category of a variable is significant, keep the variable
```

```
##
## Call:
## glm(formula = survived ~ pclass + sex + age + sibsp + parch,
##     family = binomial("logit"), data = dt.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3443  -0.6762  -0.4251   0.6418   2.4894
```

```
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.290760   0.444088   5.158 2.49e-07 ***
## pclass      -1.073843   0.128952  -8.327  < 2e-16 ***
## sex          2.620261   0.199008  13.167  < 2e-16 ***
## age         -0.040253   0.007533  -5.343 9.13e-08 ***
## sibsp       -0.351841   0.122658  -2.868  0.00412 **
## parch        0.101272   0.117570   0.861  0.38903
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1068.25  on 784  degrees of freedom
## Residual deviance:  723.19  on 779  degrees of freedom
## AIC: 735.19
##
## Number of Fisher Scoring iterations: 4
```

```
      ## Need to keep all the coefficients for a significant categorical variable
## This comes from the car package
vif1 <- vif(LR1)
min(vif1)
```

```
## [1] 1.092409
```

```
max(vif1) ## No multicollinearity since the max VIF is 4, which is less than 5
```

```
## [1] 1.356767
```

```
## Use this to predict training set and also the test set
LR2 <- glm(survived ~ pclass + sex + age + sibsp, family = binomial("logit"), data = dt.train)

smre2 <- summary(LR2)
smre2
```

```
##
## Call:
## glm(formula = survived ~ pclass + sex + age + sibsp, family = binomial("logit"),
##     data = dt.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2948  -0.6745  -0.4237   0.6478   2.4828
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.321826   0.441742   5.256 1.47e-07 ***
## pclass      -1.074954   0.128794  -8.346  < 2e-16 ***
## sex          2.645167   0.197457  13.396  < 2e-16 ***
## age         -0.040622   0.007517  -5.404 6.52e-08 ***
## sibsp       -0.318025   0.115572  -2.752  0.00593 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 1068.25  on 784  degrees of freedom
## Residual deviance:  723.93  on 780  degrees of freedom
## AIC: 733.93
##
## Number of Fisher Scoring iterations: 4
```

```
vif2 <- vif(LR2)
max(vif2)
```

```
## [1] 1.350517
```

```
# write.csv(smre1$coefficients, "Final LR Model for Titanic Data.csv")
confint(LR2)
```

```
## Waiting for profiling to be done...
```

```
##                   2.5 %      97.5 %
## (Intercept)  1.46780193  3.20236971
## pclass      -1.33275391 -0.82713846
## sex          2.26603456  3.04107927
## age         -0.05563581 -0.02612462
## sibsp       -0.55260354 -0.09878319
```

```
# #Predict training data using the model LR1
observed.train <- dt.train$survived
predicted.train <- predict(LR2, dt.train, type = 'response')
## predict.train consists of P(Y=1) for each observation in the training set
predicted.train <- round(predicted.train) ## Round to 0 or 1 to get the Y values

## Evaluate Performance of the LR Classifier on the training set
## Confusion Matrix of observed versus predicted Y values
CM.train <- table(observed.train, predicted.train)
CM.train
```

```
##                 predicted.train
## observed.train   0   1
##              0 384  71
##              1  95 235
```

```
FP <- CM.train[1,2]/(CM.train[1,1]+CM.train[1,2]) ## false positive
FN <- CM.train[2,1]/(CM.train[2,1]+CM.train[2,2]) ## false negative
FP
```

```
## [1] 0.156044
```

```
FN
```

```
## [1] 0.2878788
```

```
## Overall accuracy
## Calculated by summing the correct predictions divided by the total of the Confusion Matrix
OA.train <- sum(diag(CM.train)) / sum(CM.train)
OA.train
```

```
## [1] 0.788535
```

```
## Precision, Recall, F1-Measure are performance measures for binary prediction
## F1-measure = geometric mean of the Precision and Recall
```

```r
## Compute Precision, Recall, F1 for Category 1, model LR1
Recall1 <- CM.train[2,2] / (CM.train[2,1] + CM.train[2,2]) ## diag/row sum
Precision1 <- CM.train[2,2] / (CM.train[1,2] + CM.train[2,2]) ## diag/column sum
F1.1 <- 2 / ((1 / Recall1) + (1 / Precision1)) ## The geometric mean

PRF1_train_1 <- (c(Precision1, Recall1, F1.1))
PRF1_train_1
```

```
## [1] 0.7679739 0.7121212 0.7389937
```

```r
# Category 0
## Repeat formulas, but use different positions in the Confusion Matrix
Recall0 <- CM.train[1,1] / (CM.train[1,1] + CM.train[1,2])
Precision0 <- CM.train[1,1] / (CM.train[1,1] + CM.train[2,1])
F1.0 <- 2 / ((1 / Recall0) + (1 / Precision0))

PRF1_train_0 <- c(Precision0, Recall0, F1.0)
PRF1_train_0
```

```
## [1] 0.8016701 0.8439560 0.8222698
```

```r
## Why is the LR2 model doing a better job of predicting 1's and so-so job for category 0?
## Because there are so many more 1's than 0's. This is an example of an unbalanced data set.
## There are methods for dealing with unbalanced data sets.
```

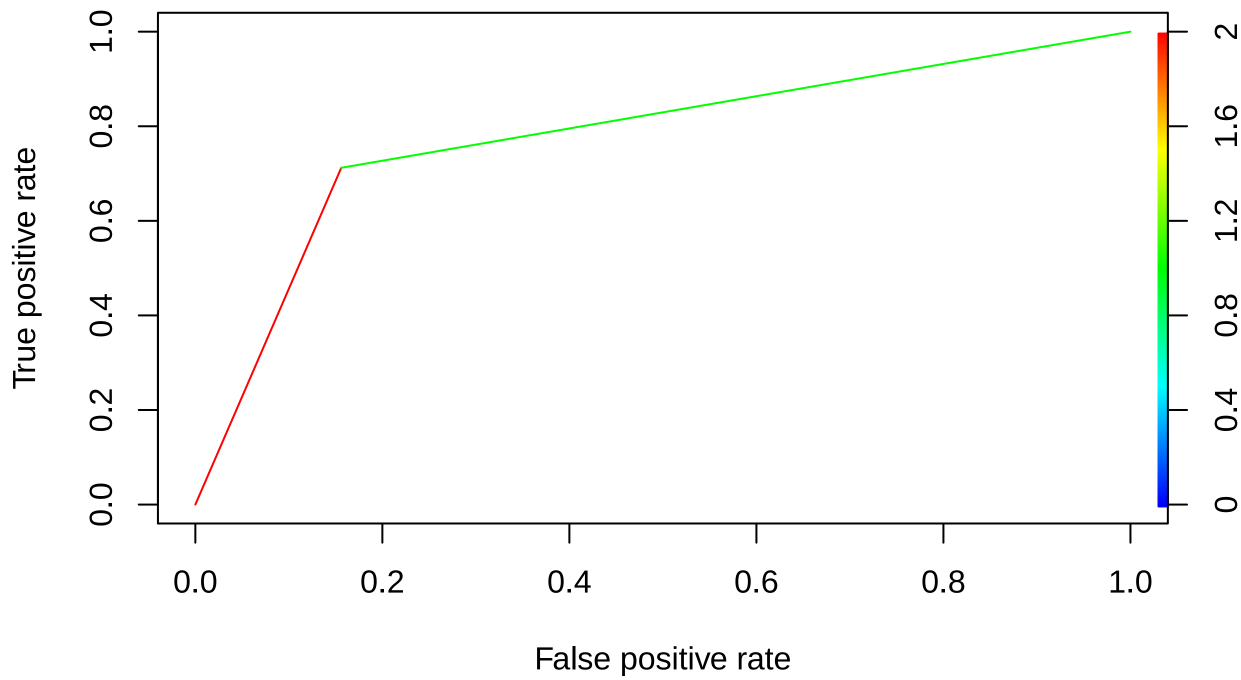
```r
r2fun(LR1)
```

```
##    McFadden  CoxSnell Nagelkerke
## 1 0.3230182 0.3556885  0.4783638
```

```r
str(predicted.train)
```

```
##  Named num [1:785] 1 0 1 1 0 1 0 1 1 0 ...
##  - attr(*, "names")= chr [1:785] "2" "4" "5" "7" ...
```

```r
predicted.train <- as.numeric(predicted.train)

pred <- prediction(predicted.train, observed.train)
perf <- performance(pred, "tpr", "fpr")
plot(perf, colorize = TRUE)
```



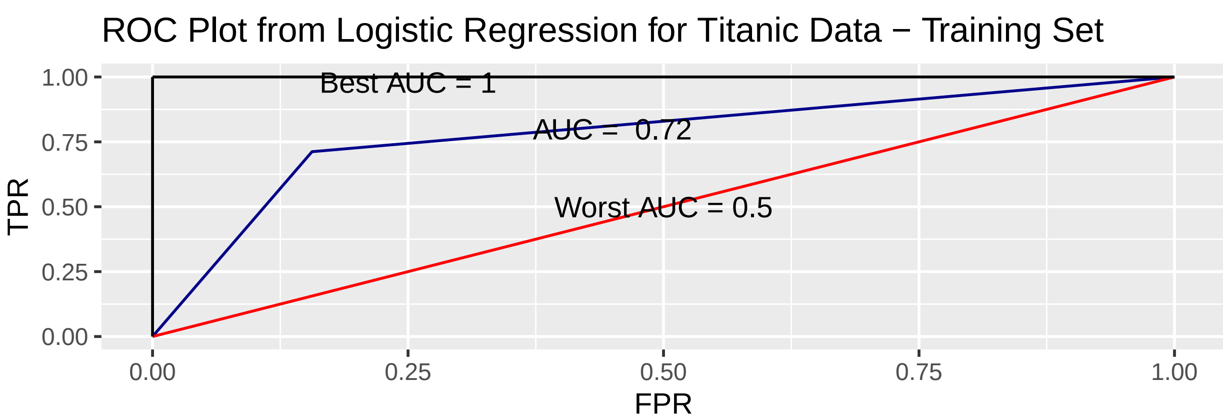

```
#calculate AUC
roc_obj <- roc(predicted.train, observed.train)
```

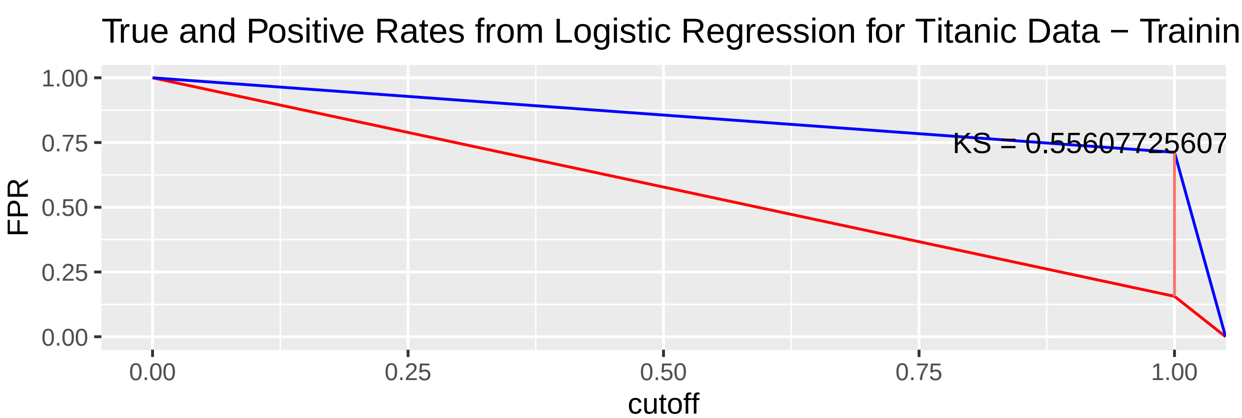
```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
AUC.train <- auc(roc_obj)
GINI.train <- 2 * AUC.train - 1

# ROC curve in ggplot2
DF.PR <- cbind.data.frame(perf@x.values[[1]], perf@y.values[[1]], perf@alpha.values[[1]])
colnames(DF.PR) <- c("FPR", "TPR", "cutoff")

#to add the 45 degree line to the plot
x <- c(0, 1)
y <- c(0, 1)
df2 <- cbind.data.frame(x, y)

# to add the AUC to the plot
x1 <- c(0, 1)
y1 <- c(1, 1)
df3 <- cbind.data.frame(x1, y1)

pROC.train <- ggplot() +
  geom_line(data = DF.PR, aes(x = FPR, y = TPR), color = "darkblue") +
  geom_line(data = df2, aes(x = x, y = y), color = "red") +
  geom_line(data = df3, aes(x = x1, y = y1), color = "black") +
  geom_segment(aes(x = 0, y = 0,xend = 0, yend = 1)) +
  annotate("text", x = 0.45, y = 0.80, label = "AUC =  0.72") +
  annotate("text", x = 0.25, y = 0.98, label = "Best AUC = 1") +
  annotate("text", x = 0.5, y = 0.5, label = "Worst AUC = 0.5") +
  ggtitle("ROC Plot from Logistic Regression for Titanic Data - Training Set")
```

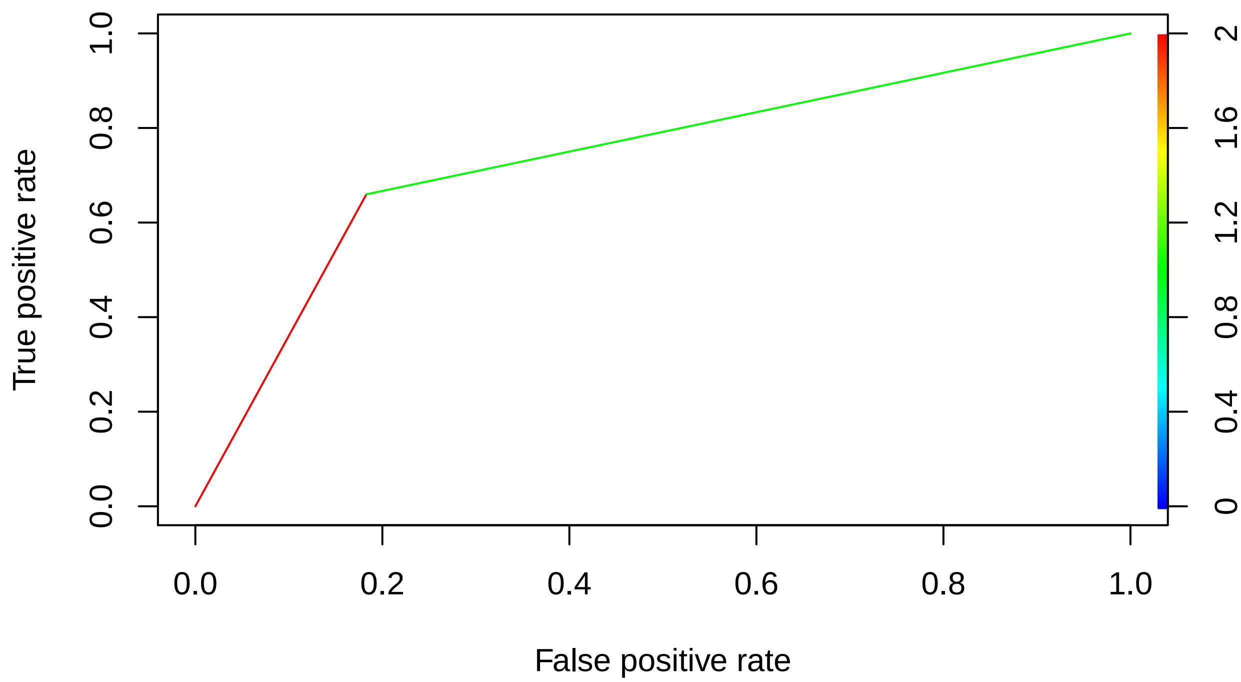
```r
## Kolmogorov-Smirnov Statistics (Performance Measure for Binary Classifiers)
## This is not the same as Kolmogorov-Smirnov Test Statistics for testing normality (of residuals in MLR
# KS = maximum(TPR-FPR)

pK1 <- ggplot() +
       geom_line(data = DF.PR, aes(x = cutoff, y = FPR), color = "red") +
       geom_line(data = DF.PR, aes(x = cutoff, y = TPR), color = "blue")

DF.PR$diff <- DF.PR$TPR - DF.PR$FPR
KS.train <- max(DF.PR$diff)
i.m <- which.max(DF.PR$diff)
xM <- DF.PR$cutoff[i.m]
yML <- DF.PR$FPR[i.m]
yMU <- DF.PR$TPR[i.m]

pKS.train <- pK1 +
             geom_segment(aes(x = xM, y = yML, xend = xM, yend = yMU, colour = "black")) +
             annotate("text", x = 0.95, y = 0.75, label = paste0("KS = ", KS.train)) +
             theme(legend.position = "none") +
             ggtitle("True and Positive Rates from Logistic Regression for Titanic Data - Training Set")

grid.arrange(pROC.train, pKS.train, nrow = 2)
```






```r
observed.test <- dt.test$survived
predicted.test <- predict(LR2, dt.test,type = "response")
predicted.test <- round(predicted.test)
```

```r
#confusion matrix for Test set
CM.Test <- table(observed.test,predicted.test)
OA.Test <- sum(diag(CM.Test))/sum(CM.Test) # 0.7471264

#Precision, Recall, F1 for Test Data - Category 1
Recall.F <- CM.Test[2,2]/(CM.Test[2,1]+CM.Test[2,2])
Precision.F <- CM.Test[2,2]/(CM.Test[1,2]+CM.Test[2,2])
F1.F <- 2/((1/Recall.F)+(1/Precision.F))

#Precision, Recall, F1 for Test Data - Category 0
Recall.F0 <- CM.Test[1,1]/(CM.Test[1,1]+CM.Test[1,2])
Precision.F0 <- CM.Test[1,1]/(CM.Test[1,1]+CM.Test[2,1])
F1.F0 <- 2/((1/Recall.F0)+(1/Precision.F0))

PRF1_test_0 <- c(Precision.F0, Recall.F0, F1.F0)

# ROC Curve test set
pred <- prediction(predicted.test, observed.test)
perf <- performance(pred, "tpr", "fpr")
plot(perf, colorize = TRUE)
```



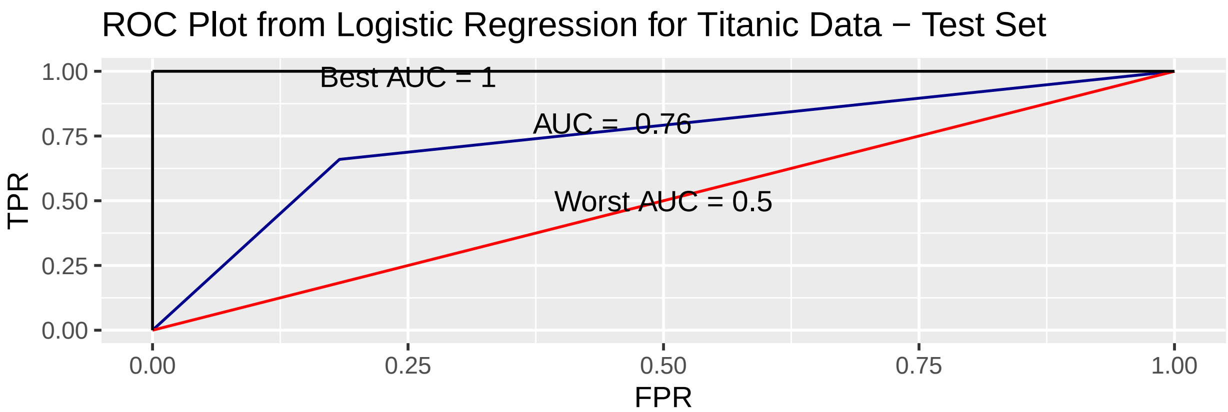

```r
#calculate AUC
roc_obj <- roc(predicted.test, observed.test)
```

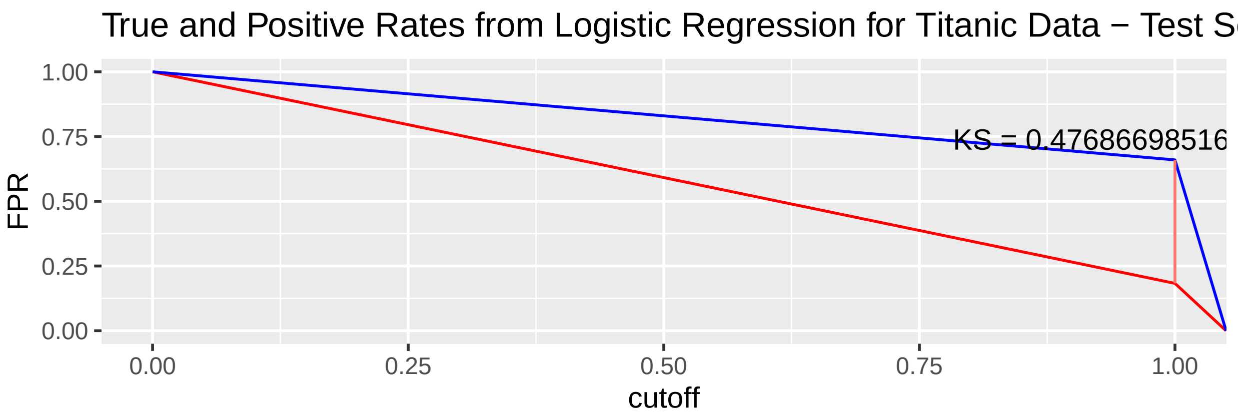
```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```r
AUC.test <- auc(roc_obj)
GINI.test <- 2 * AUC.test - 1

# OC curve in ggplot2
DF.PR <- cbind.data.frame(perf@x.values[[1]], perf@y.values[[1]], perf@alpha.values[[1]])
```

```r
colnames(DF.PR) <- c("FPR", "TPR", "cutoff")

# to add the 45 degree line to the plot
x <- c(0,1)
y <- c(0,1)
df2 <- cbind.data.frame(x,y)

#to add the AUC to the plot
x1 <- c(0,1)
y1 <- c(1,1)
df3 <- cbind.data.frame(x1,y1)

pROC.test <- ggplot() +
             geom_line(data = DF.PR, aes(x = FPR, y = TPR), color = "darkblue") +
             geom_line(data = df2, aes(x = x, y = y), color = "red") +
             geom_line(data = df3, aes(x = x1, y = y1), color = "black") +
             geom_segment(aes(x = 0, y = 0, xend = 0, yend = 1)) +
             annotate("text", x = 0.45, y = 0.80, label = "AUC =  0.76") +
             annotate("text", x = 0.25, y = 0.98, label = "Best AUC = 1") +
             annotate("text", x = 0.5, y = 0.5, label = "Worst AUC = 0.5") +
             ggtitle("ROC Plot from Logistic Regression for Titanic Data - Test Set")

#KS (Kolomogorov-Smirnov) Statistic
#KS = maximum(TPR-FPR)
pK1 <- ggplot() +
       geom_line(data = DF.PR, aes(x = cutoff, y = FPR), color = "red") +
       geom_line(data = DF.PR, aes(x = cutoff, y = TPR), color = "blue")

DF.PR$diff <- DF.PR$TPR - DF.PR$FPR
KS.test <- max(DF.PR$diff)
i.m <- which.max(DF.PR$diff)
xM <- DF.PR$cutoff[i.m]
yML <- DF.PR$FPR[i.m]
yMU <- DF.PR$TPR[i.m]

pKS.test <- pK1 +
       geom_segment(aes(x = xM, y = yML, xend = xM, yend = yMU, colour = "black")) +
       annotate("text", x = 0.95, y = 0.74, label = paste0("KS = ", KS.test)) +
       theme(legend.position = "none") +
       ggtitle("True and Positive Rates from Logistic Regression for Titanic Data - Test Set")

grid.arrange(pROC.test, pKS.test, nrow = 2)
```






```r
OA <- c(OA.train, OA.Test)
names(OA) <- c("Overall accuracy_training", "Overall accuracy_test")

names(PRF1_train_1) <- c("Precision_train_1", "Recall_train_1", "F1_train_1")
names(PRF1_train_0) <- c("Precision_train_0", "Recall_train_0", "F1_train_0")
# names(PRF1_test_1) <- c("Precision_test_1", "Recall_test_1", "F1_test_1")
# names(PRF1_test_0) <- c("Precision_test_0", "Recall_test_0", "F1_test_0")

AUC  <- c(AUC.train, AUC.test)
GINI <- c(GINI.train, GINI.test)
names(AUC) <- c("AUC_train", "AUC_test")
names(GINI) <- c("GINI_train", "GINI_test")

# print performance results for both training and test sets
print("Logistic Regression Summary of Results for Titanic Data")
```

```
## [1] "Logistic Regression Summary of Results for Titanic Data"
```

```r
print(PRF1_train_1)
```

```
## Precision_train_1    Recall_train_1        F1_train_1
##         0.7679739         0.7121212         0.7389937
```

```r
print(PRF1_train_0)
```

```
## Precision_train_0    Recall_train_0        F1_train_0
##         0.8016701         0.8439560         0.8222698
```

```r
# print(PRF1_test_1)
# print(PRF1_test_0)

print(OA)
```

```
## Overall accuracy_training     Overall accuracy_test
##                 0.7885350                 0.7586207
```

```r
print(AUC)
```

```
## AUC_train  AUC_test
## 0.7848220 0.7416231
```

```r
print(GINI)
```

```
## GINI_train  GINI_test
##  0.5696440  0.4832463
```