

```
library(readxl)

data <- read_excel("Book1.xlsx")

library(mice)

#Multivariate Imputation using pmm method

data = data.frame(data)

#all information are true for the following variables
summary(data3)
```

```
##      Long_1      Long_2      Long_3      Long_4
## Min.   : 100   Min.   : 0.000   Min.   : 0   Min.   : 0.0
## 1st Qu.: 1100  1st Qu.: 1.000   1st Qu.: 0   1st Qu.: 161.0
## Median : 2300  Median : 2.000   Median : 0   Median : 231.0
## Mean   : 2745  Mean   : 2.823   Mean   : 1788  Mean   : 257.4
## 3rd Qu.: 4100  3rd Qu.: 4.000   3rd Qu.: 833  3rd Qu.: 323.0
## Max.   :10000  Max.   :34.000   Max.   :15352  Max.   :1174.0
##      Long_5      Long_6      Long_7      Long_8
## Min.   : 0.00   Min.   : 0.00   Min.   : 0.00   Min.   : 0
## 1st Qu.: 11.00  1st Qu.: 2.00   1st Qu.:11.00   1st Qu.: 3440
## Median : 39.00  Median : 4.00   Median :18.00   Median : 12015
## Mean   : 66.04  Mean   : 6.28   Mean   :24.76   Mean   : 28475
## 3rd Qu.:104.00  3rd Qu.: 8.00   3rd Qu.:35.00   3rd Qu.: 30468
## Max.   :686.00  Max.   :87.00   Max.   :97.00   Max.   :996223
##      Long_9      Long_10     Short_1     Short_2
## Min.   : 0.000   Min.   : -10000.0   Mode :logical   Mode :logical
## 1st Qu.: 2.000   1st Qu.: -346.0    FALSE:2657      FALSE:2527
## Median : 3.000   Median : 0.0       TRUE :1942       TRUE :2072
## Mean   : 5.746   Mean   : 78.1
## 3rd Qu.: 5.000   3rd Qu.: 403.0
## Max.   :223.000   Max.   : 20000.0
##      Short_3      Short_4      Short_5      Short_6      Short_7
## Mode:logical   Mode:logical   Mode :logical   Mode:logical   Mode :logical
## TRUE:1814      TRUE:1836      FALSE:2         TRUE:1836      FALSE:144
## NA's:2785      NA's:2763      TRUE :4597      NA's:2763      TRUE :4455
##
##
##
##      Short_8      Short_9      Short_10
## Mode :logical   Mode :logical   Mode :logical
## FALSE:50        FALSE:3138      FALSE:2815
## TRUE :4549      TRUE :1461      TRUE :1784
##
##
##
```

```
data3$Short_3 <- TRUE
data3$Short_4 <- TRUE
data3$Short_6 <- TRUE
```

```
#they are all true, so no useful for modeling
```

```
data_filled <- data.frame(data[,1:4], data3)
```

```
write.csv(data_filled, file = "data_filled.csv")
```

```
train_data <- data_filled[data_filled$TrainVal == "Train_60", ]
```

```
val_data <- data_filled[data_filled$TrainVal == "Val_40", ]
```

```
head(train_data)
```

```
## UniqueID submission_year target TrainVal Long_1 Long_2 Long_3 Long_4 Long_5
## 1 984TAH 2015 0 Train_60 1800 6 0 221 0
## 3 394ETK 2015 1 Train_60 700 1 0 147 17
## 4 036KQK 2015 0 Train_60 1700 2 0 461 187
## 5 996RNP 2015 0 Train_60 600 3 0 96 30
## 8 283LEL 2015 0 Train_60 200 3 0 218 190
## 9 695XKD 2015 0 Train_60 5000 7 8963 68 39
## Long_6 Long_7 Long_8 Long_9 Long_10 Short_1 Short_2 Short_3 Short_4 Short_5
## 1 15 6 1575 7 15 FALSE TRUE TRUE TRUE TRUE
## 3 10 12 9587 4 3279 FALSE TRUE TRUE TRUE TRUE
## 4 6 4 210 2 -2139 TRUE FALSE TRUE TRUE TRUE
## 5 11 5 43650 4 316 TRUE TRUE TRUE TRUE TRUE
## 8 6 17 34298 1 0 FALSE TRUE TRUE TRUE TRUE
## 9 26 34 6312 4 229 TRUE TRUE TRUE TRUE TRUE
## Short_6 Short_7 Short_8 Short_9 Short_10
## 1 TRUE TRUE TRUE FALSE FALSE
## 3 TRUE TRUE TRUE FALSE TRUE
## 4 TRUE TRUE TRUE TRUE TRUE
## 5 TRUE TRUE TRUE FALSE FALSE
## 8 TRUE TRUE TRUE FALSE TRUE
## 9 TRUE FALSE TRUE FALSE TRUE
```

```
modell1 <- glm(target~., data = train_data[, -c(1,4)], family = binomial(link="logit"))
```

```
summary(modell1)
```

```
##
```

```
## Call:
```

```
## glm(formula = target ~ ., family = binomial(link = "logit"),
```

```
## data = train_data[, -c(1, 4)])
```

```
##
```

```
## Deviance Residuals:
```

```
## Min 1Q Median 3Q Max
```

```
## -1.8183 -0.7291 -0.6409 -0.4807 2.2067
```

```
##
```

```
## Coefficients: (3 not defined because of singularities)
```

```
## Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept) -5.284e+01 3.519e+02 -0.150 0.88062
```

```
## submission_year 2.012e-02 6.718e-02 0.300 0.76454
```

```
## Long_1 -4.626e-05 2.689e-05 -1.720 0.08542 .
```

```
## Long_2          5.497e-02  1.692e-02   3.248  0.00116 **
## Long_3          1.008e-05  1.425e-05   0.707  0.47948
## Long_4         -1.005e-03  3.739e-04  -2.687  0.00722 **
## Long_5         -1.507e-03  7.274e-04  -2.072  0.03826 *
## Long_6          4.174e-02  7.674e-03   5.440  5.34e-08 ***
## Long_7         -2.046e-03  2.941e-03  -0.696  0.48658
## Long_8          4.194e-08  8.611e-07   0.049  0.96115
## Long_9         -1.638e-03  3.456e-03  -0.474  0.63545
## Long_10        -1.937e-06  1.343e-05  -0.144  0.88532
## Short_1TRUE    -1.859e-02  9.500e-02  -0.196  0.84488
## Short_2TRUE    -2.056e-02  9.503e-02  -0.216  0.82871
## Short_3TRUE           NA           NA           NA           NA
## Short_4TRUE           NA           NA           NA           NA
## Short_5TRUE     1.120e+01  3.247e+02   0.034  0.97248
## Short_6TRUE           NA           NA           NA           NA
## Short_7TRUE     5.366e-01  3.104e-01   1.729  0.08387 .
## Short_8TRUE    -5.451e-01  4.117e-01  -1.324  0.18546
## Short_9TRUE    -1.039e-01  1.064e-01  -0.977  0.32881
## Short_10TRUE   -1.368e-02  1.006e-01  -0.136  0.89188
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 2986.3  on 2767  degrees of freedom
## Residual deviance: 2854.2  on 2749  degrees of freedom
## AIC: 2892.2
##
## Number of Fisher Scoring iterations: 11
```

```
predict_train <- predict(model1, type = "response")

predict_class <- round(predict_train )

confusion_mat_train <- table(predict_class, train_data$target)
confusion_mat_train
```

```
##
## predict_class    0    1
##               0 2116  593
##               1   15   44
```

```
library(caret)

#show some measures
confusionMatrix(factor(predict_class), factor(train_data$target), positive = "1")
```

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    0    1
##              0 2116  593
```

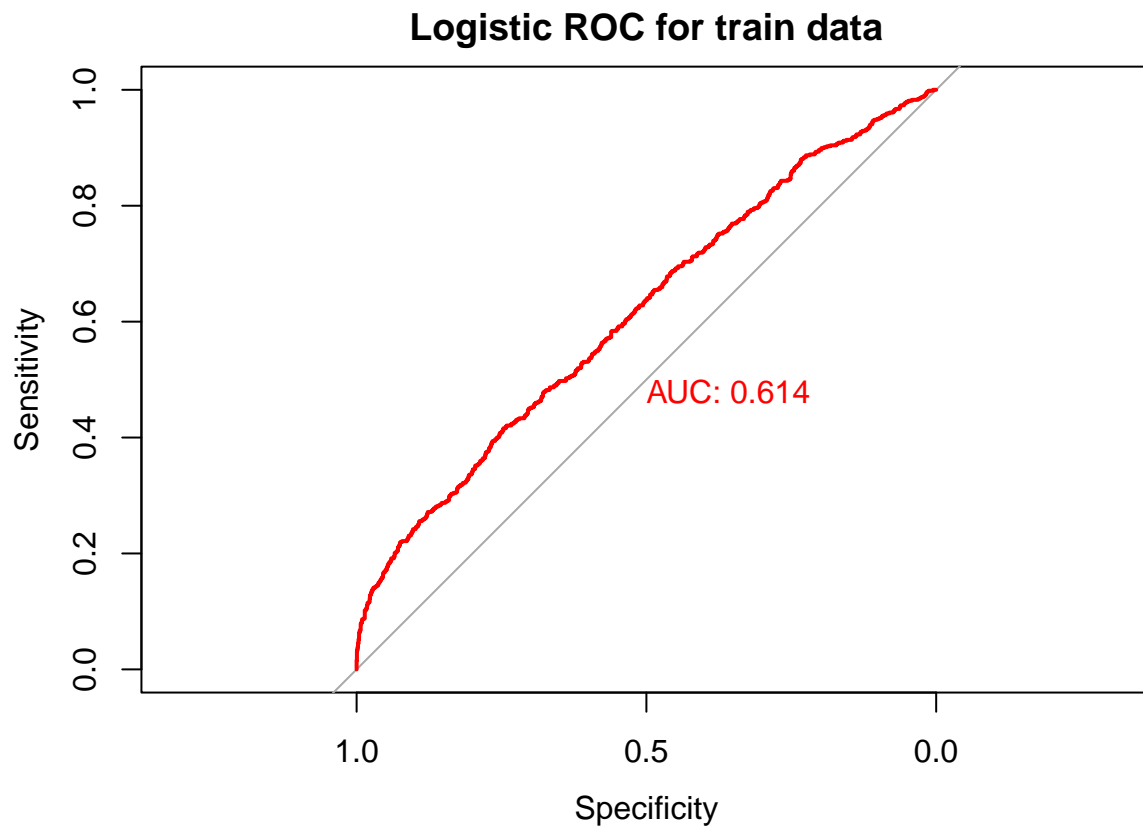
```
##          1   15   44
##
##          Accuracy : 0.7803
##          95% CI : (0.7645, 0.7956)
##    No Information Rate : 0.7699
##    P-Value [Acc > NIR] : 0.09857
##
##          Kappa : 0.091
##
## Mcnemar's Test P-Value : < 2e-16
##
##          Sensitivity : 0.06907
##          Specificity : 0.99296
##    Pos Pred Value : 0.74576
##    Neg Pred Value : 0.78110
##          Prevalence : 0.23013
##    Detection Rate : 0.01590
##    Detection Prevalence : 0.02132
##    Balanced Accuracy : 0.53102
##
##    'Positive' Class : 1
##
```

#ROC curve with auc

```
library(pROC)

r = roc(train_data$target, predict_train)

plot(r, col = "red", print.auc = TRUE, main = "Logistic ROC for train data")
```



```
predict_val <- predict(model1, val_data, type = "response")
predict_class <- round(predict_val )

confusion_mat_val <- table(predict_class, val_data$target)
confusion_mat_val
```

```
##
## predict_class    0    1
##               0 1403  401
##               1   11  16
```

```
#show some measures
confusionMatrix(factor(predict_class), factor(val_data$target), positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##           0 1403  401
##           1   11  16
##
##           Accuracy : 0.775
##           95% CI : (0.7552, 0.7939)
```

```
##      No Information Rate : 0.7723
##      P-Value [Acc > NIR] : 0.4028
##
##              Kappa : 0.0456
##
##      McNemar's Test P-Value : <2e-16
##
##              Sensitivity : 0.038369
##              Specificity : 0.992221
##              Pos Pred Value : 0.592593
##              Neg Pred Value : 0.777716
##              Prevalence : 0.227744
##              Detection Rate : 0.008738
##      Detection Prevalence : 0.014746
##      Balanced Accuracy : 0.515295
##
##      'Positive' Class : 1
##
```

#ROC curve with auc

```
r2 = roc(val_data$target, predict_val)
```

```
plot(r2, col = "red", print.auc = TRUE, main = "Logistic ROC for validation data")
```

