Untitled

Lê Tuấn Minh - 11224210

2025-06-12

library(readxl)

dat <- read\_excel("D:/Python/hmeq.xlsx")

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

dat <- dat %>%  
 select(-gDEBTINC)

str(dat)

## tibble [5,960 × 13] (S3: tbl\_df/tbl/data.frame)  
## $ REASON : chr [1:5960] NA "HomeImp" "DebtCon" "HomeImp" ...  
## $ JOB : chr [1:5960] NA "Other" NA "Mgr" ...  
## $ BAD : num [1:5960] 0 1 0 0 0 0 1 0 1 0 ...  
## $ LOAN : num [1:5960] 4400 2300 5400 3600 3800 5300 5000 5900 5000 6000 ...  
## $ MORTDUE: num [1:5960] NA 37626 53429 61327 51180 ...  
## $ VALUE : num [1:5960] 60250 46200 64505 76484 63459 ...  
## $ YOJ : num [1:5960] NA 3 5 9 20 7 3 2 17.5 1.6 ...  
## $ DEROG : num [1:5960] NA 0 NA NA 0 NA 0 NA 0 1 ...  
## $ DELINQ : num [1:5960] NA 1 0 2 0 NA 0 NA 2 NA ...  
## $ CLAGE : num [1:5960] NA 122.3 69.5 202.5 203.8 ...  
## $ NINQ : num [1:5960] NA 1 NA 0 0 NA 0 NA 0 NA ...  
## $ CLNO : num [1:5960] NA 14 16 25 20 18 14 NA 14 13 ...  
## $ DEBTINC: num [1:5960] NA NA 36.5 41.5 20.1 ...

head(dat)

## # A tibble: 6 × 13  
## REASON JOB BAD LOAN MORTDUE VALUE YOJ DEROG DELINQ CLAGE NINQ CLNO  
## <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 <NA> <NA> 0 4400 NA 60250 NA NA NA NA NA NA  
## 2 HomeImp Other 1 2300 37626 46200 3 0 1 122. 1 14  
## 3 DebtCon <NA> 0 5400 53429 64505 5 NA 0 69.5 NA 16  
## 4 HomeImp Mgr 0 3600 61327 76484 9 NA 2 203. 0 25  
## 5 HomeImp Office 0 3800 51180 63459 20 0 0 204. 0 20  
## 6 HomeImp Office 0 5300 44993 50673 7 NA NA 185. NA 18  
## # ℹ 1 more variable: DEBTINC <dbl>

# Chọn các cột số từ dataframe của bạn  
numeric\_cols <- unlist(lapply(dat, is.numeric))  
dat\_numeric <- dat[, numeric\_cols]

# Tính toán ma trận tương quan  
correlation\_matrix <- cor(dat\_numeric, use = "pairwise.complete.obs")  
  
# In ma trận tương quan ra console để xem  
print(correlation\_matrix)

## BAD LOAN MORTDUE VALUE YOJ  
## BAD 1.000000000 -0.075098920 -0.048219181 -0.029953689 -0.06023796  
## LOAN -0.075098920 1.000000000 0.228594664 0.335392924 0.10572770  
## MORTDUE -0.048219181 0.228594664 1.000000000 0.875665616 -0.08848027  
## VALUE -0.029953689 0.335392924 0.875665616 1.000000000 0.00775852  
## YOJ -0.060237964 0.105727701 -0.088480271 0.007758520 1.00000000  
## DEROG 0.276080986 -0.001301635 -0.049516042 -0.048689347 -0.06587886  
## DELINQ 0.354106876 -0.035144493 -0.001044278 -0.014102747 0.03815347  
## CLAGE -0.170499112 0.088539599 0.140046856 0.171238446 0.20171826  
## NINQ 0.174979676 0.044487492 0.031391983 -0.004397793 -0.07162702  
## CLNO -0.004157227 0.072631198 0.323631440 0.269148768 0.02483845  
## DEBTINC 0.199834555 0.084734740 0.154939092 0.132174426 -0.05589128  
## DEROG DELINQ CLAGE NINQ CLNO  
## BAD 0.276080986 0.354106876 -0.17049911 0.174979676 -0.004157227  
## LOAN -0.001301635 -0.035144493 0.08853960 0.044487492 0.072631198  
## MORTDUE -0.049516042 -0.001044278 0.14004686 0.031391983 0.323631440  
## VALUE -0.048689347 -0.014102747 0.17123845 -0.004397793 0.269148768  
## YOJ -0.065878855 0.038153475 0.20171826 -0.071627023 0.024838454  
## DEROG 1.000000000 0.211831861 -0.08304730 0.173933843 0.061883729  
## DELINQ 0.211831861 1.000000000 0.02248837 0.067812437 0.164638607  
## CLAGE -0.083047302 0.022488373 1.00000000 -0.116934575 0.237986998  
## NINQ 0.173933843 0.067812437 -0.11693457 1.000000000 0.088388713  
## CLNO 0.061883729 0.164638607 0.23798700 0.088388713 1.000000000  
## DEBTINC 0.017065206 0.052364467 -0.04647669 0.141343633 0.185538527  
## DEBTINC  
## BAD 0.19983456  
## LOAN 0.08473474  
## MORTDUE 0.15493909  
## VALUE 0.13217443  
## YOJ -0.05589128  
## DEROG 0.01706521  
## DELINQ 0.05236447  
## CLAGE -0.04647669  
## NINQ 0.14134363  
## CLNO 0.18553853  
## DEBTINC 1.00000000

dat <- dat %>%  
 select(-MORTDUE)

# Bước 0: Cài đặt và tải các gói cần thiết  
  
library(dplyr)  
library(caret) # Để chia train/test stratified

## Warning: package 'caret' was built under R version 4.4.2

## Loading required package: ggplot2

## Loading required package: lattice

library(scorecard) # Cho Optimal Binning và Scorecard

## Warning: package 'scorecard' was built under R version 4.4.3

library(ggplot2) # Cho biểu đồ (tùy chọn)  
  
# Kiểm tra lại cấu trúc dữ liệu sau khi loại bỏ biến  
str(dat)

## tibble [5,960 × 12] (S3: tbl\_df/tbl/data.frame)  
## $ REASON : chr [1:5960] NA "HomeImp" "DebtCon" "HomeImp" ...  
## $ JOB : chr [1:5960] NA "Other" NA "Mgr" ...  
## $ BAD : num [1:5960] 0 1 0 0 0 0 1 0 1 0 ...  
## $ LOAN : num [1:5960] 4400 2300 5400 3600 3800 5300 5000 5900 5000 6000 ...  
## $ VALUE : num [1:5960] 60250 46200 64505 76484 63459 ...  
## $ YOJ : num [1:5960] NA 3 5 9 20 7 3 2 17.5 1.6 ...  
## $ DEROG : num [1:5960] NA 0 NA NA 0 NA 0 NA 0 1 ...  
## $ DELINQ : num [1:5960] NA 1 0 2 0 NA 0 NA 2 NA ...  
## $ CLAGE : num [1:5960] NA 122.3 69.5 202.5 203.8 ...  
## $ NINQ : num [1:5960] NA 1 NA 0 0 NA 0 NA 0 NA ...  
## $ CLNO : num [1:5960] NA 14 16 25 20 18 14 NA 14 13 ...  
## $ DEBTINC: num [1:5960] NA NA 36.5 41.5 20.1 ...

summary(dat)

## REASON JOB BAD LOAN   
## Length:5960 Length:5960 Min. :0.0000 Min. : 1100   
## Class :character Class :character 1st Qu.:0.0000 1st Qu.:11100   
## Mode :character Mode :character Median :0.0000 Median :16300   
## Mean :0.1995 Mean :18608   
## 3rd Qu.:0.0000 3rd Qu.:23300   
## Max. :1.0000 Max. :89900   
##   
## VALUE YOJ DEROG DELINQ   
## Min. : 8000 Min. : 0.000 Min. : 0.0000 Min. : 0.0000   
## 1st Qu.: 66076 1st Qu.: 3.000 1st Qu.: 0.0000 1st Qu.: 0.0000   
## Median : 89236 Median : 7.000 Median : 0.0000 Median : 0.0000   
## Mean :101776 Mean : 8.922 Mean : 0.2546 Mean : 0.4494   
## 3rd Qu.:119824 3rd Qu.:13.000 3rd Qu.: 0.0000 3rd Qu.: 0.0000   
## Max. :855909 Max. :41.000 Max. :10.0000 Max. :15.0000   
## NA's :112 NA's :515 NA's :708 NA's :580   
## CLAGE NINQ CLNO DEBTINC   
## Min. : 0.0 Min. : 0.000 Min. : 0.0 Min. : 0.5245   
## 1st Qu.: 115.1 1st Qu.: 0.000 1st Qu.:15.0 1st Qu.: 29.1400   
## Median : 173.5 Median : 1.000 Median :20.0 Median : 34.8183   
## Mean : 179.8 Mean : 1.186 Mean :21.3 Mean : 33.7799   
## 3rd Qu.: 231.6 3rd Qu.: 2.000 3rd Qu.:26.0 3rd Qu.: 39.0031   
## Max. :1168.2 Max. :17.000 Max. :71.0 Max. :203.3121   
## NA's :308 NA's :510 NA's :222 NA's :1267

# ====================================================================  
# Bước 1: Chuẩn bị dữ liệu ban đầu và xử lý biến phân loại  
# ====================================================================  
  
# Chuyển biến mục tiêu BAD sang dạng factor  
dat$BAD <- as.factor(dat$BAD)  
  
# Chuyển các biến phân loại khác sang dạng factor  
# REASON và JOB là các biến ký tự, cần chuyển sang factor  
dat$REASON <- as.factor(dat$REASON)  
dat$JOB <- as.factor(dat$JOB)

# ====================================================================  
# Bước 2: Chia dữ liệu Train/Test (Stratified 80:20)  
# ====================================================================  
  
# Đặt seed để đảm bảo kết quả có thể tái lập  
set.seed(123)  
  
# Tạo chỉ mục cho tập huấn luyện sử dụng stratified sampling  
# p = 0.8 cho 80% train, list = FALSE để trả về vector chỉ mục  
trainIndex <- createDataPartition(dat$BAD, p = 0.8, list = FALSE, times = 1)  
  
# Tạo tập huấn luyện và tập kiểm tra  
train\_dat <- dat[trainIndex, ]  
test\_dat <- dat[-trainIndex, ]  
  
# Kiểm tra tỷ lệ BAD trong mỗi tập để đảm bảo stratified sampling hoạt động  
prop.table(table(train\_dat$BAD))

##   
## 0 1   
## 0.8003774 0.1996226

prop.table(table(test\_dat$BAD))

##   
## 0 1   
## 0.8010076 0.1989924

prop.table(table(dat$BAD)) # So sánh với tập gốc

##   
## 0 1   
## 0.8005034 0.1994966

cat("Kích thước tập huấn luyện:", nrow(train\_dat), "\n")

## Kích thước tập huấn luyện: 4769

cat("Kích thước tập kiểm tra:", nrow(test\_dat), "\n")

## Kích thước tập kiểm tra: 1191

# ====================================================================  
# Bước 3: Optimal Binning (WoE/IV) trên tập huấn luyện  
# ====================================================================  
  
# Hàm `woebin` từ gói `scorecard` sẽ thực hiện optimal binning  
# Nó tự động xử lý biến liên tục và phân loại, và xử lý NA  
# y: tên biến mục tiêu  
# positive: giá trị của biến mục tiêu đại diện cho "bad" (ví dụ: 1)  
# chk\_skip: các biến bỏ qua (ví dụ: ID nếu có, hoặc biến mục tiêu)  
# method: "tree" (sử dụng cây quyết định), "chimerge" (phân cụm dựa trên chi-square), "kmeans"  
# bin\_num\_limit: giới hạn số lượng bin (thường 5-10)  
# print\_info: có in thông tin quá trình binning không  
# stop\_limit: ngưỡng dừng cho cây quyết định (tùy chọn)  
# na.omit = FALSE: GIỮ NA VÀ ĐƯA VÀO 1 BIN RIÊNG (rất quan trọng cho yêu cầu của bạn)  
  
# Danh sách các biến dự đoán (loại bỏ biến mục tiêu BAD)  
predictor\_vars <- names(train\_dat)[!names(train\_dat) %in% c("BAD")]  
  
# Thực hiện binning  
# Hàm woebin sẽ trả về một list các data.frame, mỗi df là thông tin binning cho một biến  
bins <- woebin(train\_dat,  
 y = "BAD",  
 x = predictor\_vars, # Các biến dự đoán  
 positive = "1", # Chỉ ra rằng '1' là trường hợp "bad"  
 bin\_num\_limit = 6, # Giới hạn số lượng bin, bạn có thể điều chỉnh  
 na.omit = FALSE, # Giữ NA và tạo bin riêng cho chúng  
 print\_info = TRUE) # In thông tin binning

## ℹ Creating woe binning ...

## ✔ Binning on 4769 rows and 12 columns in 00:00:04

# Bạn có thể xem kết quả binning cho từng biến  
# Ví dụ: bins$LOAN  
# plot(bins$LOAN) # Để trực quan hóa các bin cho biến LOAN

# ====================================================================  
# Bước 4: Áp dụng Binning/WoE lên cả tập Train và Test  
# ====================================================================  
  
# Hàm woebin\_ply áp dụng các quy tắc binning đã học từ `woebin`  
# và chuyển đổi các biến gốc thành các giá trị WoE tương ứng.  
train\_woe <- woebin\_ply(train\_dat, bins)

## ℹ Converting into woe values ...

## ✔ Woe transformating on 4769 rows and 11 columns in 00:00:00

test\_woe <- woebin\_ply(test\_dat, bins)

## ℹ Converting into woe values ...

## ✔ Woe transformating on 1191 rows and 11 columns in 00:00:00

# Kiểm tra cấu trúc của dữ liệu sau khi chuyển đổi WoE  
str(train\_woe)

## Classes 'data.table' and 'data.frame': 4769 obs. of 12 variables:  
## $ BAD : Factor w/ 2 levels "0","1": 1 2 1 1 1 1 2 2 1 2 ...  
## $ REASON\_woe : num -0.0552 0.1245 -0.0552 0.1245 0.1245 ...  
## $ JOB\_woe : num -0.0449 0.1946 -0.0449 -0.0449 -0.4297 ...  
## $ LOAN\_woe : num 1.23 1.23 1.23 1.23 1.23 ...  
## $ VALUE\_woe : num -0.0387 0.9748 -0.0387 -0.0387 -0.0387 ...  
## $ YOJ\_woe : num -0.6428 0.2236 0.2236 -0.0654 0.1109 ...  
## $ DEROG\_woe : num -0.541 -0.228 -0.541 -0.541 -0.228 ...  
## $ DELINQ\_woe : num -0.59 0.718 -0.445 1.711 -0.445 ...  
## $ CLAGE\_woe : num 0.229 0.349 0.895 -0.217 -0.217 ...  
## $ NINQ\_woe : num -0.4388 -0.0345 -0.4388 -0.2996 -0.2996 ...  
## $ CLNO\_woe : num 0.485 -0.188 -0.188 -0.513 -0.188 ...  
## $ DEBTINC\_woe: num 1.884 1.884 -1.374 -0.872 -1.374 ...  
## - attr(\*, ".internal.selfref")=<externalptr>

str(test\_woe)

## Classes 'data.table' and 'data.frame': 1191 obs. of 12 variables:  
## $ BAD : Factor w/ 2 levels "0","1": 1 2 1 1 2 1 1 2 1 2 ...  
## $ REASON\_woe : num -0.0552 0.1245 -0.0552 0.1245 0.1245 ...  
## $ JOB\_woe : num -0.0609 0.1946 -0.0449 -0.4297 0.1946 ...  
## $ LOAN\_woe : num 1.23 1.23 1.23 1.23 1.23 ...  
## $ VALUE\_woe : num -0.4155 0.9748 -0.4155 -0.0387 -0.0387 ...  
## $ YOJ\_woe : num 0.2236 0.2236 0.1109 -0.0654 0.2236 ...  
## $ DEROG\_woe : num -0.541 -0.228 -0.228 -0.541 -0.228 ...  
## $ DELINQ\_woe : num -0.59 0.718 0.718 -0.59 -0.445 ...  
## $ CLAGE\_woe : num 0.229 0.349 -0.809 -0.217 0.349 ...  
## $ NINQ\_woe : num -0.4388 -0.2996 -0.0345 -0.4388 -0.2996 ...  
## $ CLNO\_woe : num 0.4846 0.4846 -0.5126 -0.1879 0.0789 ...  
## $ DEBTINC\_woe: num -1.37 1.88 1.88 -1.37 1.88 ...  
## - attr(\*, ".internal.selfref")=<externalptr>

# Các cột mới có hậu tố "\_WoE" sẽ xuất hiện.  
# Biến BAD vẫn còn nguyên.

# ====================================================================  
# Bước 5: Xây dựng mô hình Logistic Regression  
# ====================================================================  
  
# Chuẩn bị công thức mô hình  
# Biến mục tiêu là BAD, các biến dự đoán là tất cả các biến \_WoE  
# Lấy tên các cột WoE  
woe\_vars <- names(train\_woe)[grep("\_woe$", names(train\_woe))]  
formula\_str <- paste("BAD ~", paste(woe\_vars, collapse = " + "))  
model\_formula <- as.formula(formula\_str)  
  
cat("\nCông thức mô hình Logistic Regression:\n")

##   
## Công thức mô hình Logistic Regression:

print(model\_formula)

## BAD ~ REASON\_woe + JOB\_woe + LOAN\_woe + VALUE\_woe + YOJ\_woe +   
## DEROG\_woe + DELINQ\_woe + CLAGE\_woe + NINQ\_woe + CLNO\_woe +   
## DEBTINC\_woe

# Xây dựng mô hình  
logistic\_model <- glm(model\_formula, data = train\_woe, family = binomial(link = "logit"))  
  
# Xem tóm tắt mô hình  
summary(logistic\_model)

##   
## Call:  
## glm(formula = model\_formula, family = binomial(link = "logit"),   
## data = train\_woe)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.37925 0.05209 -26.477 < 2e-16 \*\*\*  
## REASON\_woe 0.12301 0.63521 0.194 0.846444   
## JOB\_woe 0.91304 0.24931 3.662 0.000250 \*\*\*  
## LOAN\_woe 0.46575 0.12306 3.785 0.000154 \*\*\*  
## VALUE\_woe 0.75245 0.10480 7.180 6.98e-13 \*\*\*  
## YOJ\_woe 1.03552 0.18088 5.725 1.04e-08 \*\*\*  
## DEROG\_woe 0.68774 0.08468 8.122 4.58e-16 \*\*\*  
## DELINQ\_woe 0.95428 0.06387 14.941 < 2e-16 \*\*\*  
## CLAGE\_woe 0.99041 0.10934 9.058 < 2e-16 \*\*\*  
## NINQ\_woe 0.38225 0.11945 3.200 0.001374 \*\*   
## CLNO\_woe 1.09089 0.17340 6.291 3.15e-10 \*\*\*  
## DEBTINC\_woe 0.93392 0.03440 27.147 < 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: 4767.8 on 4768 degrees of freedom  
## Residual deviance: 2682.2 on 4757 degrees of freedom  
## AIC: 2706.2  
##   
## Number of Fisher Scoring iterations: 6

# ====================================================================  
# Bước 6: Đánh giá mô hình (trên tập Test)  
# ====================================================================  
  
# Dự đoán xác suất "bad" trên tập kiểm tra  
test\_pred\_prob <- predict(logistic\_model, newdata = test\_woe, type = "response")  
  
# Chuyển đổi xác suất thành dự đoán lớp (ví dụ: ngưỡng 0.5)  
test\_pred\_class <- ifelse(test\_pred\_prob > 0.5, "1", "0")  
test\_pred\_class <- as.factor(test\_pred\_class)  
  
# Tạo confusion matrix  
confusionMatrix(test\_pred\_class, test\_woe$BAD, positive = "1")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 913 92  
## 1 41 145  
##   
## Accuracy : 0.8883   
## 95% CI : (0.8691, 0.9057)  
## No Information Rate : 0.801   
## P-Value [Acc > NIR] : 4.923e-16   
##   
## Kappa : 0.6189   
##   
## Mcnemar's Test P-Value : 1.454e-05   
##   
## Sensitivity : 0.6118   
## Specificity : 0.9570   
## Pos Pred Value : 0.7796   
## Neg Pred Value : 0.9085   
## Prevalence : 0.1990   
## Detection Rate : 0.1217   
## Detection Prevalence : 0.1562   
## Balanced Accuracy : 0.7844   
##   
## 'Positive' Class : 1   
##

# Tính toán ROC curve và AUC  
library(pROC)

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

##   
## Attaching package: 'pROC'

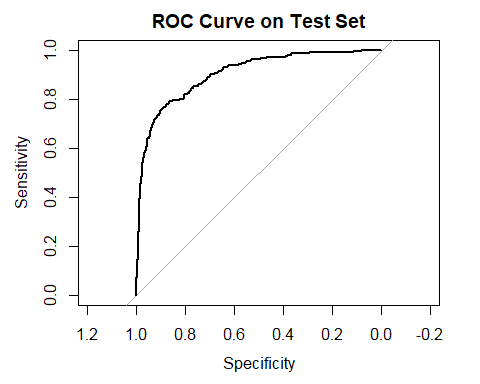
## The following objects are masked from 'package:stats':  
##   
## cov, smooth, var

roc\_obj <- roc(response = test\_woe$BAD, predictor = test\_pred\_prob)

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

## Setting direction: controls < cases

plot(roc\_obj, main = "ROC Curve on Test Set")



auc(roc\_obj)

## Area under the curve: 0.9025

# ====================================================================  
# Bước 6: Đánh giá mô hình (trên tập Test) - Tối ưu hóa Ngưỡng  
# ====================================================================  
  
# Dự đoán xác suất "bad" trên tập kiểm tra (như trước)  
test\_pred\_prob <- predict(logistic\_model, newdata = test\_woe, type = "response")  
  
# Tạo một sequence các ngưỡng để thử  
thresholds <- seq(0.05, 0.95, by = 0.01) # Thử từ 5% đến 95% với bước nhảy 1%  
  
# Khởi tạo dataframe để lưu kết quả Precision, Recall, F1-score cho mỗi ngưỡng  
results\_df <- data.frame(  
 threshold = numeric(),  
 accuracy = numeric(),  
 precision = numeric(),  
 recall = numeric(),  
 f1\_score = numeric(),  
 stringsAsFactors = FALSE  
)  
  
# Lặp qua từng ngưỡng để tính toán các độ đo  
for (thresh in thresholds) {  
 # Chuyển đổi xác suất thành dự đoán lớp với ngưỡng hiện tại  
 test\_pred\_class\_thresh <- ifelse(test\_pred\_prob > thresh, "1", "0")  
 test\_pred\_class\_thresh <- as.factor(test\_pred\_class\_thresh)  
  
 # Đảm bảo các levels của dự đoán khớp với reference  
 # Nếu không có dự đoán nào là '1' (khi ngưỡng quá cao), caret sẽ báo lỗi  
 if (nlevels(test\_pred\_class\_thresh) < nlevels(test\_woe$BAD)) {  
 levels(test\_pred\_class\_thresh) <- levels(test\_woe$BAD)  
 }  
  
 # Tạo confusion matrix  
 cm <- confusionMatrix(test\_pred\_class\_thresh, test\_woe$BAD, positive = "1")  
  
 # Trích xuất Precision, Recall, F1-score  
 current\_accuracy <- cm$overall["Accuracy"]  
 current\_precision <- cm$byClass["Pos Pred Value"] # Precision  
 current\_recall <- cm$byClass["Sensitivity"] # Recall  
  
 # Tính F1-score  
 current\_f1 <- (2 \* current\_precision \* current\_recall) / (current\_precision + current\_recall)  
  
 # Thêm vào dataframe kết quả  
 results\_df <- rbind(results\_df, data.frame(  
 threshold = thresh,  
 accuracy = current\_accuracy,  
 precision = current\_precision,  
 recall = current\_recall,  
 f1\_score = current\_f1  
 ))  
}  
  
# Loại bỏ các hàng có NA (nếu F1-score không thể tính được do Precision/Recall là NA,  
# thường xảy ra khi không có dự đoán nào là "1" hoặc không có "1" thực sự trong tập test)  
results\_df <- na.omit(results\_df)  
  
# Tìm ngưỡng tối đa hóa F1-score  
optimal\_threshold\_f1 <- results\_df$threshold[which.max(results\_df$f1\_score)]  
max\_f1\_score <- max(results\_df$f1\_score)  
  
cat("\n------------------------------------------------\n")

##   
## ------------------------------------------------

cat("Phân tích Ngưỡng Tối ưu hóa F1-score:\n")

## Phân tích Ngưỡng Tối ưu hóa F1-score:

cat("Ngưỡng tối ưu F1-score:", optimal\_threshold\_f1, "\n")

## Ngưỡng tối ưu F1-score: 0.34

cat("F1-score tối đa:", max\_f1\_score, "\n")

## F1-score tối đa: 0.7127883

# Lấy các độ đo tại ngưỡng tối ưu  
optimal\_metrics\_f1 <- results\_df %>%  
 filter(threshold == optimal\_threshold\_f1)  
print(optimal\_metrics\_f1)

## threshold accuracy precision recall f1\_score  
## Accuracy29 0.34 0.8849706 0.7083333 0.7172996 0.7127883

cat("\n------------------------------------------------\n")

##   
## ------------------------------------------------

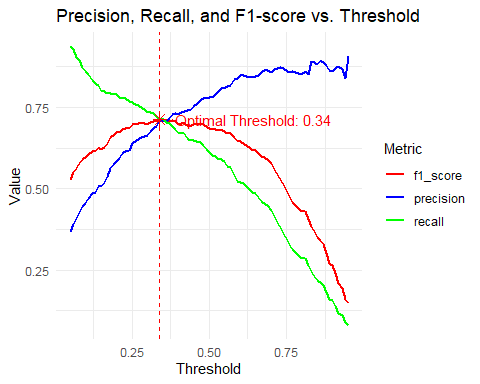
# ====================================================================  
# Trực quan hóa F1-score, Precision, Recall theo ngưỡng  
# ====================================================================  
library(ggplot2)  
library(tidyr) # Để sử dụng pivot\_longer

##   
## Attaching package: 'tidyr'

## The following object is masked from 'package:scorecard':  
##   
## replace\_na

# Chuyển đổi dataframe sang định dạng dài để dễ vẽ biểu đồ  
plot\_data <- results\_df %>%  
 pivot\_longer(  
 cols = c(precision, recall, f1\_score),  
 names\_to = "metric",  
 values\_to = "value"  
 )  
  
ggplot(plot\_data, aes(x = threshold, y = value, color = metric)) +  
 geom\_line(size = 1) +  
 geom\_point(aes(x = optimal\_threshold\_f1, y = max\_f1\_score), color = "red", size = 3, shape = 8,  
 data = data.frame(optimal\_threshold\_f1, max\_f1\_score, metric = "f1\_score", value = max\_f1\_score)) +  
 labs(  
 title = "Precision, Recall, and F1-score vs. Threshold",  
 x = "Threshold",  
 y = "Value",  
 color = "Metric"  
 ) +  
 scale\_color\_manual(values = c("precision" = "blue", "recall" = "green", "f1\_score" = "red")) +  
 theme\_minimal() +  
 geom\_vline(xintercept = optimal\_threshold\_f1, linetype = "dashed", color = "red") +  
 annotate("text", x = optimal\_threshold\_f1 + 0.05, y = max\_f1\_score,  
 label = paste0("Optimal Threshold: ", round(optimal\_threshold\_f1, 2)),  
 color = "red", hjust = 0)

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.  
## ℹ Please use `linewidth` instead.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.



# ====================================================================  
# Áp dụng ngưỡng tối ưu và xem Confusion Matrix cuối cùng  
# ====================================================================  
test\_pred\_class\_optimal <- ifelse(test\_pred\_prob > optimal\_threshold\_f1, "1", "0")  
test\_pred\_class\_optimal <- as.factor(test\_pred\_class\_optimal)  
  
# Đảm bảo các levels của dự đoán khớp với reference  
if (nlevels(test\_pred\_class\_optimal) < nlevels(test\_woe$BAD)) {  
 levels(test\_pred\_class\_optimal) <- levels(test\_woe$BAD)  
}  
  
cat("\nConfusion Matrix với ngưỡng tối ưu F1-score (", optimal\_threshold\_f1, "):\n")

##   
## Confusion Matrix với ngưỡng tối ưu F1-score ( 0.34 ):

confusionMatrix(test\_pred\_class\_optimal, test\_woe$BAD, positive = "1")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 884 67  
## 1 70 170  
##   
## Accuracy : 0.885   
## 95% CI : (0.8655, 0.9025)  
## No Information Rate : 0.801   
## P-Value [Acc > NIR] : 7.173e-15   
##   
## Kappa : 0.6409   
##   
## Mcnemar's Test P-Value : 0.8643   
##   
## Sensitivity : 0.7173   
## Specificity : 0.9266   
## Pos Pred Value : 0.7083   
## Neg Pred Value : 0.9295   
## Prevalence : 0.1990   
## Detection Rate : 0.1427   
## Detection Prevalence : 0.2015   
## Balanced Accuracy : 0.8220   
##   
## 'Positive' Class : 1   
##

# Bước 7: Tính điểm Scorecard (CẬP NHẬT CHÍNH XÁC DỰA TRÊN TÀI LIỆU CỦA BẠN)  
# ====================================================================  
  
# Hàm `scorecard` trong phiên bản 0.4.4 có các tham số points0, odds0, pdo.  
# Chúng ta sẽ sử dụng các tên tham số chính xác này.  
# Ví dụ: đặt base\_point = 600, pdo = 20, odds = 50  
card <- scorecard(bins,  
 logistic\_model,  
 points0 = 600, # Tên tham số đúng là points0  
 pdo = 50, # Tên tham số đúng là pdo  
 odds0 = 1/19) # Tên tham số đúng là odds0 (ví dụ 1/50 để có odds là 50)  
 # Lưu ý: odds0 = p/(1-p). Nếu bạn muốn odds = 50, thì odds0 = 50.  
 # Nếu odds = 1/19 là mặc định, và bạn muốn odds là 50, thì chỉ cần đặt odds0 = 50.  
  
# In bảng điểm scorecard (giờ đây sẽ phản ánh các giá trị points0, odds0, pdo bạn đã đặt)  
print(card)

## $basepoints  
## variable bin woe points  
## <char> <lgcl> <lgcl> <num>  
## 1: basepoints NA NA 487  
##   
## $REASON  
## variable bin count count\_distr neg pos posprob  
## <char> <char> <int> <num> <int> <int> <num>  
## 1: REASON DebtCon%,%missing 3357 0.7039212 2716 641 0.1909443  
## 2: REASON HomeImp 1412 0.2960788 1101 311 0.2202550  
## woe bin\_iv total\_iv breaks is\_special\_values  
## <num> <num> <num> <char> <lgcl>  
## 1: -0.05523101 0.002111716 0.006870878 DebtCon%,%missing FALSE  
## 2: 0.12447379 0.004759162 0.006870878 HomeImp FALSE  
## points  
## <num>  
## 1: 0  
## 2: -1  
##   
## $JOB  
## variable bin count count\_distr neg pos posprob  
## <char> <char> <int> <num> <int> <int> <num>  
## 1: JOB Mgr%,%missing 831 0.1742504 671 160 0.1925391  
## 2: JOB Office 752 0.1576850 647 105 0.1396277  
## 3: JOB Other 1918 0.4021808 1472 446 0.2325339  
## 4: JOB ProfExe%,%Sales%,%Self 1268 0.2658838 1027 241 0.1900631  
## woe bin\_iv total\_iv breaks is\_special\_values  
## <num> <num> <num> <char> <lgcl>  
## 1: -0.04494030 0.0003471764 0.0428826 Mgr%,%missing FALSE  
## 2: -0.42973093 0.0254446815 0.0428826 Office FALSE  
## 3: 0.19459667 0.0161212093 0.0428826 Other FALSE  
## 4: -0.06094526 0.0009695300 0.0428826 ProfExe%,%Sales%,%Self FALSE  
## points  
## <num>  
## 1: 3  
## 2: 28  
## 3: -13  
## 4: 4  
##   
## $LOAN  
## variable bin count count\_distr neg pos posprob woe  
## <char> <char> <int> <num> <int> <int> <num> <num>  
## 1: LOAN [-Inf,6000) 247 0.05179283 133 114 0.4615385 1.23450434  
## 2: LOAN [6000,15000) 1795 0.37638918 1430 365 0.2033426 0.02312265  
## 3: LOAN [15000,16000) 281 0.05892221 200 81 0.2882562 0.48478681  
## 4: LOAN [16000,38000) 2171 0.45523170 1846 325 0.1497006 -0.34829621  
## 5: LOAN [38000, Inf) 275 0.05766408 208 67 0.2436364 0.25580956  
## bin\_iv total\_iv breaks is\_special\_values points  
## <num> <num> <char> <lgcl> <num>  
## 1: 0.1048140856 0.1744679 6000 FALSE -41  
## 2: 0.0002026375 0.1744679 15000 FALSE -1  
## 3: 0.0158461599 0.1744679 16000 FALSE -16  
## 4: 0.0495414201 0.1744679 38000 FALSE 12  
## 5: 0.0040635588 0.1744679 Inf FALSE -9  
##   
## $VALUE  
## variable bin count count\_distr neg pos posprob  
## <char> <char> <int> <num> <int> <int> <num>  
## 1: VALUE [-Inf,50000)%,%missing 593 0.12434473 357 236 0.39797639  
## 2: VALUE [50000,85000) 1602 0.33591948 1292 310 0.19350811  
## 3: VALUE [85000,125000) 1514 0.31746697 1300 214 0.14134742  
## 4: VALUE [125000,170000) 532 0.11155378 417 115 0.21616541  
## 5: VALUE [170000,200000) 258 0.05409939 236 22 0.08527132  
## 6: VALUE [200000, Inf) 270 0.05661564 215 55 0.20370370  
## woe bin\_iv total\_iv breaks is\_special\_values points  
## <num> <num> <num> <char> <lgcl> <num>  
## 1: 0.97475104 1.504725e-01 0.238383 50000%,%missing FALSE -53  
## 2: -0.03871937 4.977557e-04 0.238383 85000 FALSE 2  
## 3: -0.41548851 4.811012e-02 0.238383 125000 FALSE 23  
## 4: 0.10050092 1.160808e-03 0.238383 170000 FALSE -5  
## 5: -0.98413433 3.810511e-02 0.238383 200000 FALSE 53  
## 6: 0.02535018 3.666018e-05 0.238383 Inf FALSE -1  
##   
## $YOJ  
## variable bin count count\_distr neg pos posprob woe  
## <char> <char> <int> <num> <int> <int> <num> <num>  
## 1: YOJ missing 414 0.08681065 366 48 0.1159420 -0.64277730  
## 2: YOJ [-Inf,6) 1817 0.38100231 1385 432 0.2377545 0.22362519  
## 3: YOJ [6,8) 445 0.09331097 375 70 0.1573034 -0.28977577  
## 4: YOJ [8,10) 433 0.09079472 351 82 0.1893764 -0.06541196  
## 5: YOJ [10,21) 1216 0.25498008 951 265 0.2179276 0.11087078  
## 6: YOJ [21, Inf) 444 0.09310128 389 55 0.1238739 -0.56759114  
## bin\_iv total\_iv breaks is\_special\_values points  
## <num> <num> <char> <lgcl> <num>  
## 1: 0.0292249333 0.08539415 missing TRUE 48  
## 2: 0.0203344858 0.08539415 6 FALSE -17  
## 3: 0.0071618901 0.08539415 8 FALSE 22  
## 4: 0.0003808664 0.08539415 10 FALSE 5  
## 5: 0.0032388458 0.08539415 21 FALSE -8  
## 6: 0.0250531253 0.08539415 Inf FALSE 42  
##   
## $DEROG  
## variable bin count count\_distr neg pos posprob woe  
## <char> <char> <int> <num> <int> <int> <num> <num>  
## 1: DEROG missing 552 0.1157475 482 70 0.1268116 -0.5407939  
## 2: DEROG [-Inf,1) 3622 0.7594884 3022 600 0.1656543 -0.2280895  
## 3: DEROG [1, Inf) 595 0.1247641 313 282 0.4739496 1.2843589  
## bin\_iv total\_iv breaks is\_special\_values points  
## <num> <num> <char> <lgcl> <num>  
## 1: 0.02852567 0.3404865 missing TRUE 27  
## 2: 0.03682941 0.3404865 1 FALSE 11  
## 3: 0.27513140 0.3404865 Inf FALSE -64  
##   
## $DELINQ  
## variable bin count count\_distr neg pos posprob woe  
## <char> <char> <int> <num> <int> <int> <num> <num>  
## 1: DELINQ missing 453 0.09498847 398 55 0.1214128 -0.5904638  
## 2: DELINQ [-Inf,1) 3337 0.69972741 2877 460 0.1378484 -0.4446219  
## 3: DELINQ [1,2) 541 0.11344097 358 183 0.3382625 0.7176082  
## 4: DELINQ [2, Inf) 438 0.09184315 184 254 0.5799087 1.7110535  
## bin\_iv total\_iv breaks is\_special\_values points  
## <num> <num> <char> <lgcl> <num>  
## 1: 0.02745495 0.59242 missing TRUE 41  
## 2: 0.12028801 0.59242 1 FALSE 31  
## 3: 0.07063845 0.59242 2 FALSE -49  
## 4: 0.37403857 0.59242 Inf FALSE -118  
##   
## $CLAGE  
## variable bin count count\_distr neg pos posprob woe  
## <char> <char> <int> <num> <int> <int> <num> <num>  
## 1: CLAGE missing 243 0.05095408 185 58 0.2386831 0.2287422  
## 2: CLAGE [-Inf,70) 248 0.05200252 154 94 0.3790323 0.8949972  
## 3: CLAGE [70,150) 1631 0.34200042 1205 426 0.2611895 0.3488595  
## 4: CLAGE [150,240) 1627 0.34116167 1355 272 0.1671789 -0.2170996  
## 5: CLAGE [240, Inf) 1020 0.21388132 918 102 0.1000000 -0.8085696  
## bin\_iv total\_iv breaks is\_special\_values points  
## <num> <num> <char> <lgcl> <num>  
## 1: 0.002849439 0.2239573 missing TRUE -16  
## 2: 0.052262175 0.2239573 70 FALSE -64  
## 3: 0.045974814 0.2239573 150 FALSE -25  
## 4: 0.015039914 0.2239573 240 FALSE 16  
## 5: 0.107830962 0.2239573 Inf FALSE 58  
##   
## $NINQ  
## variable bin count count\_distr neg pos posprob woe  
## <char> <char> <int> <num> <int> <int> <num> <num>  
## 1: NINQ missing 397 0.08324596 342 55 0.1385390 -0.43882253  
## 2: NINQ [-Inf,1) 2032 0.42608513 1715 317 0.1560039 -0.29961157  
## 3: NINQ [1,2) 1097 0.23002726 884 213 0.1941659 -0.03450988  
## 4: NINQ [2,3) 594 0.12455441 460 134 0.2255892 0.15526833  
## 5: NINQ [3,4) 317 0.06647096 226 91 0.2870662 0.47897953  
## 6: NINQ [4, Inf) 332 0.06961627 190 142 0.4277108 1.09745800  
## bin\_iv total\_iv breaks is\_special\_values points  
## <num> <num> <char> <lgcl> <num>  
## 1: 0.0139659890 0.1787247 missing TRUE 12  
## 2: 0.0348515798 0.1787247 1 FALSE 8  
## 3: 0.0002711095 0.1787247 2 FALSE 1  
## 4: 0.0031430673 0.1787247 3 FALSE -4  
## 5: 0.0174250033 0.1787247 4 FALSE -13  
## 6: 0.1090679574 0.1787247 Inf FALSE -30  
##   
## $CLNO  
## variable bin count count\_distr neg pos posprob  
## <char> <char> <int> <num> <int> <int> <num>  
## 1: CLNO [-Inf,10)%,%missing 628 0.13168379 447 181 0.2882166  
## 2: CLNO [10,22) 2096 0.43950514 1737 359 0.1712786  
## 3: CLNO [22,24) 367 0.07695534 289 78 0.2125341  
## 4: CLNO [24,27) 554 0.11616691 482 72 0.1299639  
## 5: CLNO [27, Inf) 1124 0.23568882 862 262 0.2330961  
## woe bin\_iv total\_iv breaks is\_special\_values points  
## <num> <num> <num> <char> <lgcl> <num>  
## 1: 0.48459345 0.0353842262 0.08625539 10%,%missing FALSE -38  
## 2: -0.18793736 0.0146532102 0.08625539 22 FALSE 15  
## 3: 0.07893716 0.0004908993 0.08625539 24 FALSE -6  
## 4: -0.51262298 0.0259627795 0.08625539 27 FALSE 40  
## 5: 0.19774425 0.0097642708 0.08625539 Inf FALSE -16  
##   
## $DEBTINC  
## variable bin count count\_distr neg pos posprob woe  
## <char> <char> <int> <num> <int> <int> <num> <num>  
## 1: DEBTINC missing 1017 0.21325225 385 632 0.62143559 1.8843011  
## 2: DEBTINC [-Inf,40) 3032 0.63577270 2852 180 0.05936675 -1.3741639  
## 3: DEBTINC [40,42) 434 0.09100440 393 41 0.09447005 -0.8715825  
## 4: DEBTINC [42, Inf) 286 0.05997064 187 99 0.34615385 0.7526663  
## bin\_iv total\_iv breaks is\_special\_values points  
## <num> <num> <char> <lgcl> <num>  
## 1: 1.06086338 1.921394 missing TRUE -127  
## 2: 0.76693190 1.921394 40 FALSE 93  
## 3: 0.05220188 1.921394 42 FALSE 59  
## 4: 0.04139683 1.921394 Inf FALSE -51

# Áp dụng bảng điểm lên dữ liệu để tính điểm cho từng khách hàng  
# (scorecard\_ply không có các tham số này trong v0.4.4)  
train\_score <- scorecard\_ply(train\_dat, card)  
test\_score <- scorecard\_ply(test\_dat, card)  
  
# BỔ SUNG: THÊM BIẾN BAD VÀO train\_score và test\_score THỦ CÔNG  
train\_score$BAD <- train\_dat$BAD  
test\_score$BAD <- test\_dat$BAD  
  
# Xem một vài khách hàng và điểm của họ  
head(train\_score)

## score BAD  
## <num> <fctr>  
## 1: 398 0  
## 2: 188 1  
## 3: 548 0  
## 4: 487 0  
## 5: 641 0  
## 6: 701 0

head(test\_score)

## score BAD  
## <num> <fctr>  
## 1: 575 0  
## 2: 142 1  
## 3: 398 0  
## 4: 684 0  
## 5: 309 1  
## 6: 426 0

# Kiểm tra lại tên cột trong test\_score để đảm bảo có BAD  
names(test\_score)

## [1] "score" "BAD"

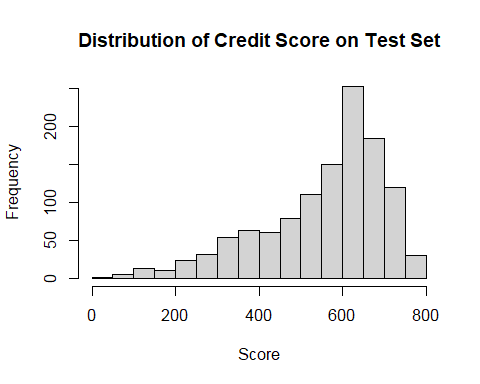
# Phân phối điểm  
summary(train\_score$score)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 26.0 462.0 591.0 550.1 659.0 808.0

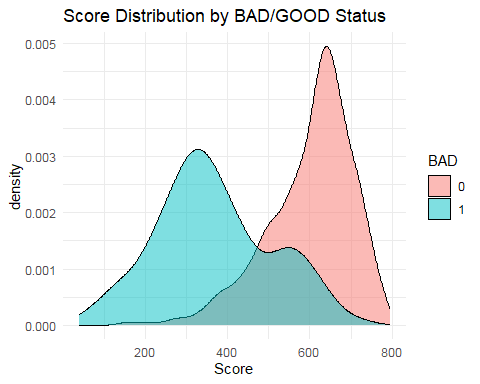
summary(test\_score$score)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 39.0 472.5 597.0 554.6 659.0 796.0

# Bạn có thể vẽ histogram của điểm  
hist(test\_score$score, main = "Distribution of Credit Score on Test Set", xlab = "Score")



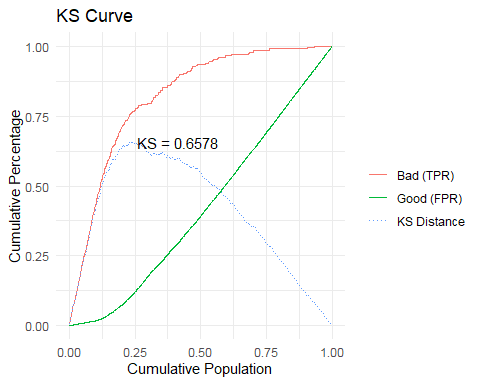
# Hoặc so sánh điểm giữa khách hàng BAD và GOOD  
library(ggplot2)  
ggplot(test\_score, aes(x = score, fill = BAD)) +  
 geom\_density(alpha = 0.5) +  
 labs(title = "Score Distribution by BAD/GOOD Status", x = "Score", fill = "BAD") +  
 theme\_minimal()



# Bước 8: Các chỉ tiêu đánh giá  
# ====================================================================  
# Chuyển BAD về số  
actual <- as.numeric(as.character(test\_woe$BAD)) # 1 = bad, 0 = good  
predicted <- test\_pred\_prob  
  
# Tạo dataframe  
ks\_df <- data.frame(actual = actual, predicted = predicted)  
  
# Sắp xếp theo xác suất giảm dần  
ks\_df <- ks\_df[order(-ks\_df$predicted), ]  
  
# Tính tỷ lệ tích lũy  
ks\_df$bad <- ifelse(ks\_df$actual == 1, 1, 0)  
ks\_df$good <- ifelse(ks\_df$actual == 0, 1, 0)  
ks\_df$bad\_cum\_pct <- cumsum(ks\_df$bad) / sum(ks\_df$bad)  
ks\_df$good\_cum\_pct <- cumsum(ks\_df$good) / sum(ks\_df$good)  
  
# KS là độ lệch lớn nhất giữa TPR và FPR  
ks\_df$ks\_stat <- abs(ks\_df$bad\_cum\_pct - ks\_df$good\_cum\_pct)  
ks\_val <- max(ks\_df$ks\_stat)  
cat("KS Statistic:", round(ks\_val, 4), "\n")

## KS Statistic: 0.6578

library(ggplot2)  
  
ks\_df$row\_id <- seq\_len(nrow(ks\_df)) / nrow(ks\_df) # Xác suất tích lũy theo rank  
  
ggplot(ks\_df, aes(x = row\_id)) +  
 geom\_line(aes(y = bad\_cum\_pct, color = "Bad (TPR)")) +  
 geom\_line(aes(y = good\_cum\_pct, color = "Good (FPR)")) +  
 geom\_line(aes(y = ks\_stat, color = "KS Distance"), linetype = "dotted") +  
 labs(  
 title = "KS Curve",  
 x = "Cumulative Population",  
 y = "Cumulative Percentage",  
 color = ""  
 ) +  
 annotate("text", x = ks\_df$row\_id[which.max(ks\_df$ks\_stat)],  
 y = ks\_val,  
 label = paste0("KS = ", round(ks\_val, 4)),  
 hjust = -0.1, vjust = 0.5, color = "black") +  
 theme\_minimal()



# Cắt score thành các bins giống nhau  
cut\_bins <- woebin\_ply(data.frame(score = c(train\_score$score, test\_score$score)),   
 woebin(data.frame(score = c(train\_score$score, test\_score$score),  
 BAD = c(train\_score$BAD, test\_score$BAD)),  
 y = "BAD", x = "score", bin\_num\_limit = 10))

## ℹ Converting into woe values ...  
## ℹ Creating woe binning ...

## ✔ Binning on 5960 rows and 2 columns in 00:00:00  
## ✔ Woe transformating on 5960 rows and 1 columns in 00:00:00

# Phân phối tỷ lệ trong từng bin  
train\_score$bin <- cut\_bins$score\_woe[1:nrow(train\_score)]  
test\_score$bin <- cut\_bins$score\_woe[(nrow(train\_score)+1):(nrow(train\_score) + nrow(test\_score))]  
  
psi\_table <- data.frame(table(train\_score$bin) / nrow(train\_score))  
names(psi\_table) <- c("bin", "train\_pct")  
psi\_table$test\_pct <- table(test\_score$bin) / nrow(test\_score)  
psi\_table$psi <- (psi\_table$train\_pct - psi\_table$test\_pct) \* log(psi\_table$train\_pct / psi\_table$test\_pct)  
  
# Tổng PSI  
psi\_val <- sum(psi\_table$psi, na.rm = TRUE)  
cat("PSI (Population Stability Index):", round(psi\_val, 4), "\n")

## PSI (Population Stability Index): 0.0022

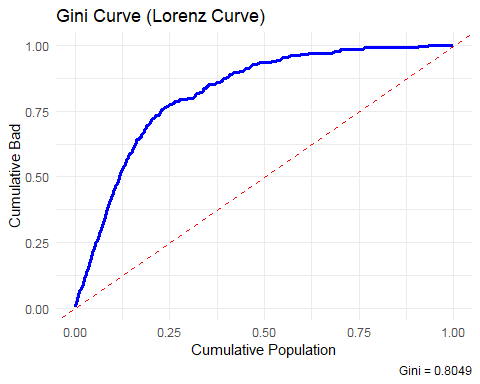
# In bảng PSI  
print(psi\_table)

## bin train\_pct test\_pct psi  
## 1 -2.78909334461712 0.40721325 0.42821159 1.055805e-03  
## 2 -0.673702380663279 0.34640386 0.34089001 8.847218e-05  
## 3 1.11390362319927 0.11427972 0.10663308 5.295709e-04  
## 4 2.34376143628872 0.07863284 0.07472712 1.989819e-04  
## 5 4.1319796859987 0.05347033 0.04953820 3.003466e-04

# Dữ liệu đã có: test\_pred\_prob, test\_woe$BAD  
# Tạo dataframe  
gini\_df <- data.frame(actual = as.numeric(as.character(test\_woe$BAD)),  
 predicted = test\_pred\_prob)  
  
# Sắp xếp theo xác suất giảm dần  
gini\_df <- gini\_df[order(-gini\_df$predicted), ]  
  
# Thêm cột thứ tự tích lũy (cumulative population)  
gini\_df$cum\_pop <- seq\_along(gini\_df$actual) / nrow(gini\_df)  
  
# Tính tỷ lệ bad tích lũy  
gini\_df$cum\_bad <- cumsum(gini\_df$actual) / sum(gini\_df$actual)  
  
# Đường lý tưởng (perfect model): 100% bad nằm trong top đầu  
# Đường ngẫu nhiên (random model): đường chéo  
  
# Vẽ Gini (Lorenz) Curve  
library(ggplot2)  
ggplot(gini\_df, aes(x = cum\_pop, y = cum\_bad)) +  
 geom\_line(color = "blue", size = 1.2) +  
 geom\_abline(slope = 1, intercept = 0, color = "red", linetype = "dashed") +  
 labs(  
 title = "Gini Curve (Lorenz Curve)",  
 x = "Cumulative Population",  
 y = "Cumulative Bad",  
 caption = paste("Gini =", round(2 \* auc(roc(response = test\_woe$BAD, predictor = test\_pred\_prob)) - 1, 4))  
 ) +  
 theme\_minimal()

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

## Setting direction: controls < cases



# ====================================================================

# Bước 9: In ra Score và logodds  
  
test\_prob\_bad <- predict(logistic\_model, newdata = test\_woe, type = "response")  
test\_prob\_good <- 1 - test\_prob\_bad  
  
# Odds\_Good\_Bad = P(Good) / P(Bad)  
odds\_good\_bad <- test\_prob\_good / test\_prob\_bad  
  
# Odds\_Bad\_Good = P(Bad) / P(Good)  
odds\_bad\_good <- test\_prob\_bad / test\_prob\_good  
  
# Gộp dữ liệu lại để tiện phân tích  
analysis\_df <- data.frame(  
 score = test\_score$score, # Điểm số từ scorecard\_ply  
 prob\_bad = test\_prob\_bad,  
 prob\_good = test\_prob\_good,  
 odds\_good\_bad = odds\_good\_bad,  
 odds\_bad\_good = odds\_bad\_good  
)  
  
# Sắp xếp theo điểm số tăng dần  
analysis\_df\_sorted <- analysis\_df %>%  
 arrange(score)  
  
# In ra một vài hàng đầu và cuối để xem xu hướng  
cat("\n--- Dữ liệu sắp xếp theo Score tăng dần ---\n")

##   
## --- Dữ liệu sắp xếp theo Score tăng dần ---

print(head(analysis\_df\_sorted))

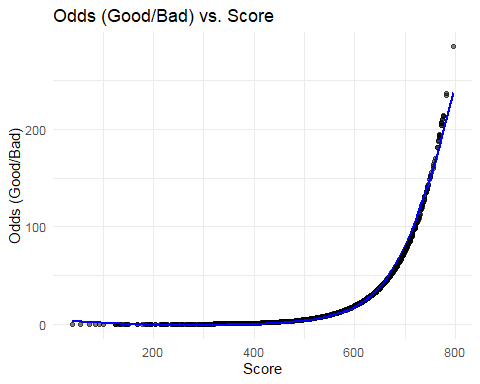
## score prob\_bad prob\_good odds\_good\_bad odds\_bad\_good  
## 10 39 0.9920614 0.007938621 0.008002147 124.96647  
## 1183 54 0.9900809 0.009919087 0.010018461 99.81573  
## 1137 72 0.9875239 0.012476074 0.012633693 79.15342  
## 1138 85 0.9850684 0.014931558 0.015157888 65.97225  
## 1141 92 0.9836151 0.016384945 0.016657884 60.03163  
## 40 100 0.9817637 0.018236258 0.018574997 53.83581

print(tail(analysis\_df\_sorted))

## score prob\_bad prob\_good odds\_good\_bad odds\_bad\_good  
## 176 776 0.004679618 0.9953204 212.6927 0.004701619  
## 206 776 0.004679618 0.9953204 212.6927 0.004701619  
## 1094 776 0.004645564 0.9953544 214.2591 0.004667246  
## 523 781 0.004204757 0.9957952 236.8259 0.004222511  
## 107 782 0.004245480 0.9957545 234.5446 0.004263581  
## 1093 796 0.003498198 0.9965018 284.8615 0.003510478

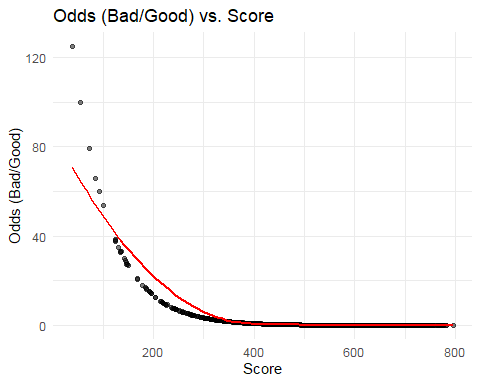
# Plot Odds\_Good\_Bad vs Score  
ggplot(analysis\_df, aes(x = score, y = odds\_good\_bad)) +  
 geom\_point(alpha = 0.5) +  
 geom\_smooth(method = "loess", se = FALSE, color = "blue") +  
 labs(title = "Odds (Good/Bad) vs. Score", x = "Score", y = "Odds (Good/Bad)") +  
 theme\_minimal()

## `geom\_smooth()` using formula = 'y ~ x'



# Plot Odds\_Bad\_Good vs Score  
ggplot(analysis\_df, aes(x = score, y = odds\_bad\_good)) +  
 geom\_point(alpha = 0.5) +  
 geom\_smooth(method = "loess", se = FALSE, color = "red") +  
 labs(title = "Odds (Bad/Good) vs. Score", x = "Score", y = "Odds (Bad/Good)") +  
 theme\_minimal()

## `geom\_smooth()` using formula = 'y ~ x'



analysis\_df\_sorted

## score prob\_bad prob\_good odds\_good\_bad odds\_bad\_good  
## 10 39 0.992061379 0.007938621 8.002147e-03 1.249665e+02  
## 1183 54 0.990080913 0.009919087 1.001846e-02 9.981573e+01  
## 1137 72 0.987523926 0.012476074 1.263369e-02 7.915342e+01  
## 1138 85 0.985068442 0.014931558 1.515789e-02 6.597225e+01  
## 1141 92 0.983615055 0.016384945 1.665788e-02 6.003163e+01  
## 40 100 0.981763742 0.018236258 1.857500e-02 5.383581e+01  
## 1146 124 0.974264442 0.025735558 2.641537e-02 3.785674e+01  
## 1150 124 0.974694494 0.025305506 2.596250e-02 3.851709e+01  
## 64 125 0.974377325 0.025622675 2.629646e-02 3.802793e+01  
## 54 130 0.972313315 0.027686685 2.847507e-02 3.511844e+01  
## 686 134 0.970840327 0.029159673 3.003550e-02 3.329394e+01  
## 849 135 0.970530892 0.029469108 3.036391e-02 3.293384e+01  
## 41 136 0.970576793 0.029423207 3.031518e-02 3.298678e+01  
## 2 142 0.967737870 0.032262130 3.333767e-02 2.999609e+01  
## 983 144 0.966800426 0.033199574 3.433964e-02 2.912087e+01  
## 592 146 0.965636132 0.034363868 3.558677e-02 2.810033e+01  
## 675 147 0.964836738 0.035163262 3.644478e-02 2.743877e+01  
## 1187 148 0.964924018 0.035075982 3.635103e-02 2.750954e+01  
## 870 150 0.964002055 0.035997945 3.734219e-02 2.677936e+01  
## 15 167 0.954931399 0.045068601 4.719564e-02 2.118840e+01  
## 60 168 0.954110528 0.045889472 4.809660e-02 2.079149e+01  
## 1052 177 0.947573591 0.052426409 5.532701e-02 1.807436e+01  
## 1149 183 0.944283688 0.055716312 5.900379e-02 1.694807e+01  
## 1007 185 0.942418722 0.057581278 6.109946e-02 1.636676e+01  
## 1128 186 0.941406390 0.058593610 6.224051e-02 1.606671e+01  
## 22 191 0.938621090 0.061378910 6.539264e-02 1.529224e+01  
## 13 192 0.937971078 0.062028922 6.613095e-02 1.512151e+01  
## 1181 193 0.936778044 0.063221956 6.748873e-02 1.481729e+01  
## 30 195 0.935287362 0.064712638 6.919011e-02 1.445293e+01  
## 670 195 0.934345436 0.065654564 7.026798e-02 1.423123e+01  
## 271 203 0.927321822 0.072678178 7.837428e-02 1.275929e+01  
## 313 203 0.926727721 0.073272279 7.906560e-02 1.264773e+01  
## 894 213 0.915841941 0.084158059 9.189147e-02 1.088240e+01  
## 32 216 0.913809805 0.086190195 9.431962e-02 1.060225e+01  
## 1021 217 0.912744955 0.087255045 9.559631e-02 1.046066e+01  
## 1035 217 0.912482284 0.087517716 9.591169e-02 1.042626e+01  
## 676 218 0.910079395 0.089920605 9.880523e-02 1.012092e+01  
## 885 218 0.911398572 0.088601428 9.721480e-02 1.028650e+01  
## 819 219 0.909795628 0.090204372 9.914795e-02 1.008594e+01  
## 35 224 0.905480102 0.094519898 1.043865e-01 9.579783e+00  
## 170 224 0.905745366 0.094254634 1.040631e-01 9.609558e+00  
## 264 226 0.901171536 0.098828464 1.096667e-01 9.118542e+00  
## 49 228 0.902802978 0.097197022 1.076614e-01 9.288381e+00  
## 1081 236 0.891380445 0.108619555 1.218554e-01 8.206445e+00  
## 56 238 0.886484415 0.113515585 1.280514e-01 7.809363e+00  
## 467 240 0.884339455 0.115660545 1.307875e-01 7.645991e+00  
## 31 241 0.882803764 0.117196236 1.327546e-01 7.532697e+00  
## 1153 241 0.882674947 0.117325053 1.329199e-01 7.523329e+00  
## 201 243 0.880949784 0.119050216 1.351385e-01 7.399817e+00  
## 629 243 0.879392610 0.120607390 1.371485e-01 7.291366e+00  
## 55 244 0.879549698 0.120450302 1.369454e-01 7.302179e+00  
## 846 246 0.876580116 0.123419884 1.407970e-01 7.102422e+00  
## 1172 246 0.874954063 0.125045937 1.429171e-01 6.997061e+00  
## 1174 246 0.876898092 0.123101908 1.403834e-01 7.123351e+00  
## 909 251 0.869474550 0.130525450 1.501199e-01 6.661341e+00  
## 247 252 0.865721609 0.134278391 1.551057e-01 6.447215e+00  
## 619 257 0.858053739 0.141946261 1.654282e-01 6.044920e+00  
## 917 257 0.859357888 0.140642112 1.636595e-01 6.110246e+00  
## 864 258 0.855893880 0.144106120 1.683691e-01 5.939331e+00  
## 878 262 0.849130856 0.150869144 1.776748e-01 5.628261e+00  
## 934 265 0.844776927 0.155223073 1.837445e-01 5.442341e+00  
## 912 268 0.840190980 0.159809020 1.902056e-01 5.257469e+00  
## 931 269 0.838728062 0.161271938 1.922816e-01 5.200707e+00  
## 37 272 0.832788544 0.167211456 2.007850e-01 4.980451e+00  
## 106 273 0.829316342 0.170683658 2.058125e-01 4.858792e+00  
## 120 273 0.829632054 0.170367946 2.053536e-01 4.869649e+00  
## 1046 273 0.832205914 0.167794086 2.016257e-01 4.959686e+00  
## 1142 273 0.828564023 0.171435977 2.069073e-01 4.833081e+00  
## 1119 274 0.828023645 0.171976355 2.076950e-01 4.814753e+00  
## 963 275 0.823463065 0.176536935 2.143836e-01 4.664537e+00  
## 1190 275 0.828802109 0.171197891 2.065606e-01 4.841193e+00  
## 102 277 0.822014707 0.177985293 2.165232e-01 4.618442e+00  
## 52 278 0.821210989 0.178789011 2.177139e-01 4.593185e+00  
## 1008 278 0.820424842 0.179575158 2.188807e-01 4.568699e+00  
## 16 279 0.818751770 0.181248230 2.213714e-01 4.517295e+00  
## 114 279 0.817655097 0.182344903 2.230096e-01 4.484113e+00  
## 862 279 0.818747841 0.181252159 2.213773e-01 4.517176e+00  
## 935 279 0.818863164 0.181136836 2.212053e-01 4.520688e+00  
## 1163 281 0.813935058 0.186064942 2.285992e-01 4.374468e+00  
## 868 286 0.803538440 0.196461560 2.444955e-01 4.090054e+00  
## 17 289 0.796716389 0.203283611 2.551518e-01 3.919236e+00  
## 1055 290 0.795620121 0.204379879 2.568812e-01 3.892850e+00  
## 61 293 0.789878780 0.210121220 2.660170e-01 3.759158e+00  
## 639 294 0.781642072 0.218357928 2.793580e-01 3.579637e+00  
## 1047 298 0.774839131 0.225160869 2.905905e-01 3.441269e+00  
## 906 300 0.773391594 0.226608406 2.930060e-01 3.412899e+00  
## 67 301 0.768768168 0.231231832 3.007823e-01 3.324664e+00  
## 891 301 0.769556446 0.230443554 2.994498e-01 3.339457e+00  
## 178 302 0.764815181 0.235184819 3.075054e-01 3.251975e+00  
## 257 302 0.764815181 0.235184819 3.075054e-01 3.251975e+00  
## 809 302 0.764578705 0.235421295 3.079098e-01 3.247704e+00  
## 21 305 0.758882399 0.241117601 3.177272e-01 3.147354e+00  
## 658 305 0.756250190 0.243749810 3.223137e-01 3.102567e+00  
## 978 307 0.751404719 0.248595281 3.308407e-01 3.022603e+00  
## 5 309 0.747791147 0.252208853 3.372718e-01 2.964968e+00  
## 29 309 0.749012353 0.250987647 3.350915e-01 2.984260e+00  
## 688 309 0.745234224 0.254765776 3.418600e-01 2.925174e+00  
## 1091 309 0.750350068 0.249649932 3.327113e-01 3.005609e+00  
## 130 312 0.739394154 0.260605846 3.524586e-01 2.837212e+00  
## 174 312 0.738565011 0.261434989 3.539769e-01 2.825043e+00  
## 701 312 0.740649701 0.259350299 3.501659e-01 2.855789e+00  
## 78 313 0.736166635 0.263833365 3.583881e-01 2.790271e+00  
## 540 314 0.735337533 0.264662467 3.599197e-01 2.778397e+00  
## 110 316 0.725521809 0.274478191 3.783183e-01 2.643277e+00  
## 351 316 0.728733574 0.271266426 3.722436e-01 2.686413e+00  
## 34 317 0.727572249 0.272427751 3.744340e-01 2.670698e+00  
## 215 317 0.724159719 0.275840281 3.809108e-01 2.625286e+00  
## 495 317 0.724660633 0.275339367 3.799563e-01 2.631882e+00  
## 121 319 0.717580774 0.282419226 3.935713e-01 2.540835e+00  
## 25 321 0.714497229 0.285502771 3.995856e-01 2.502593e+00  
## 42 321 0.718212085 0.281787915 3.923464e-01 2.548768e+00  
## 630 321 0.712613331 0.287386669 4.032856e-01 2.479633e+00  
## 645 322 0.713043547 0.286956453 4.024389e-01 2.484849e+00  
## 1177 322 0.708608398 0.291391602 4.112167e-01 2.431808e+00  
## 26 325 0.708109638 0.291890362 4.122107e-01 2.425944e+00  
## 53 325 0.707037282 0.292962718 4.143526e-01 2.413404e+00  
## 267 325 0.704112538 0.295887462 4.202275e-01 2.379663e+00  
## 643 325 0.701669397 0.298330603 4.251726e-01 2.351986e+00  
## 1120 325 0.706416913 0.293583087 4.155946e-01 2.406191e+00  
## 36 326 0.704411342 0.295588658 4.196251e-01 2.383080e+00  
## 51 327 0.696659392 0.303340608 4.354217e-01 2.296624e+00  
## 1022 328 0.696850412 0.303149588 4.350282e-01 2.298701e+00  
## 905 330 0.691790733 0.308209267 4.455238e-01 2.244549e+00  
## 790 331 0.683612197 0.316387803 4.628177e-01 2.160678e+00  
## 633 333 0.677512350 0.322487650 4.759879e-01 2.100894e+00  
## 1167 333 0.680937536 0.319062464 4.685635e-01 2.134183e+00  
## 91 335 0.674117252 0.325882748 4.834215e-01 2.068588e+00  
## 506 338 0.664115491 0.335884509 5.057622e-01 1.977214e+00  
## 1037 338 0.662728765 0.337271235 5.089129e-01 1.964973e+00  
## 1005 339 0.660750138 0.339249862 5.134314e-01 1.947680e+00  
## 9 340 0.660159786 0.339840214 5.147848e-01 1.942559e+00  
## 235 340 0.661443266 0.338556734 5.118455e-01 1.953715e+00  
## 948 340 0.657141639 0.342858361 5.217420e-01 1.916656e+00  
## 837 342 0.653340132 0.346659868 5.305963e-01 1.884672e+00  
## 871 342 0.648664733 0.351335267 5.416284e-01 1.846284e+00  
## 902 343 0.649451695 0.350548305 5.397604e-01 1.852674e+00  
## 1112 343 0.652590956 0.347409044 5.323534e-01 1.878451e+00  
## 1133 344 0.650025980 0.349974020 5.384001e-01 1.857355e+00  
## 497 345 0.639290038 0.360709962 5.642352e-01 1.772310e+00  
## 243 349 0.630502134 0.369497866 5.860375e-01 1.706376e+00  
## 118 351 0.625962367 0.374037633 5.975401e-01 1.673528e+00  
## 897 352 0.623598758 0.376401242 6.035952e-01 1.656739e+00  
## 126 353 0.617555003 0.382444997 6.192890e-01 1.614755e+00  
## 546 353 0.617334687 0.382665313 6.198669e-01 1.613250e+00  
## 138 356 0.607968108 0.392031892 6.448231e-01 1.550813e+00  
## 65 359 0.599899542 0.400100458 6.669458e-01 1.499372e+00  
## 296 359 0.594789322 0.405210678 6.812676e-01 1.467852e+00  
## 988 359 0.597709536 0.402290464 6.730534e-01 1.485766e+00  
## 253 360 0.588920900 0.411079100 6.980209e-01 1.432622e+00  
## 232 362 0.584974002 0.415025998 7.094777e-01 1.409488e+00  
## 211 363 0.581045125 0.418954875 7.210367e-01 1.386892e+00  
## 343 363 0.582760422 0.417239578 7.159710e-01 1.396705e+00  
## 350 363 0.578788925 0.421211075 7.277456e-01 1.374107e+00  
## 673 363 0.584708337 0.415291663 7.102544e-01 1.407946e+00  
## 755 364 0.577071870 0.422928130 7.328864e-01 1.364468e+00  
## 899 365 0.578123647 0.421876353 7.297338e-01 1.370363e+00  
## 974 365 0.576047910 0.423952090 7.359667e-01 1.358757e+00  
## 982 365 0.578062700 0.421937300 7.299161e-01 1.370020e+00  
## 1030 365 0.579824809 0.420175191 7.246589e-01 1.379960e+00  
## 105 366 0.575467198 0.424532802 7.377185e-01 1.355531e+00  
## 548 366 0.574808836 0.425191164 7.397088e-01 1.351883e+00  
## 980 366 0.574636675 0.425363325 7.402300e-01 1.350931e+00  
## 112 367 0.570057770 0.429942230 7.542082e-01 1.325894e+00  
## 663 367 0.571658746 0.428341254 7.492954e-01 1.334587e+00  
## 1155 369 0.563869956 0.436130044 7.734586e-01 1.292894e+00  
## 735 370 0.560276241 0.439723759 7.848338e-01 1.274155e+00  
## 1107 370 0.559213337 0.440786663 7.882263e-01 1.268671e+00  
## 533 371 0.558224638 0.441775362 7.913935e-01 1.263594e+00  
## 970 371 0.558537162 0.441462838 7.903912e-01 1.265196e+00  
## 43 372 0.551265956 0.448734044 8.140065e-01 1.228491e+00  
## 240 372 0.550604775 0.449395225 8.161848e-01 1.225213e+00  
## 89 374 0.545710353 0.454289647 8.324739e-01 1.201239e+00  
## 938 375 0.544101714 0.455898286 8.378917e-01 1.193472e+00  
## 1027 377 0.534389698 0.465610302 8.712936e-01 1.147719e+00  
## 507 378 0.533775498 0.466224502 8.734468e-01 1.144889e+00  
## 780 379 0.529256502 0.470743498 8.894430e-01 1.124299e+00  
## 324 380 0.523151100 0.476848900 9.114936e-01 1.097100e+00  
## 1115 380 0.524416107 0.475583893 9.068827e-01 1.102678e+00  
## 738 381 0.523000791 0.476999209 9.120430e-01 1.096440e+00  
## 990 381 0.523630385 0.476369615 9.097440e-01 1.099210e+00  
## 608 382 0.519069982 0.480930018 9.265225e-01 1.079305e+00  
## 340 383 0.514114984 0.485885016 9.450902e-01 1.058100e+00  
## 731 384 0.509903591 0.490096409 9.611550e-01 1.040415e+00  
## 749 384 0.512339839 0.487660161 9.518295e-01 1.050608e+00  
## 807 386 0.505254072 0.494745928 9.792023e-01 1.021239e+00  
## 960 386 0.505224895 0.494775105 9.793166e-01 1.021120e+00  
## 867 387 0.499463245 0.500536755 1.002149e+00 9.978553e-01  
## 75 389 0.497588705 0.502411295 1.009692e+00 9.904011e-01  
## 1134 389 0.497241001 0.502758999 1.011097e+00 9.890246e-01  
## 216 392 0.483064668 0.516935332 1.070116e+00 9.344779e-01  
## 573 392 0.482874870 0.517125130 1.070930e+00 9.337679e-01  
## 1017 392 0.487878940 0.512121060 1.049689e+00 9.526633e-01  
## 638 393 0.478448394 0.521551606 1.090090e+00 9.173558e-01  
## 875 393 0.477580240 0.522419760 1.093889e+00 9.141696e-01  
## 1053 393 0.487758193 0.512241807 1.050196e+00 9.522030e-01  
## 664 395 0.478653033 0.521346967 1.089196e+00 9.181084e-01  
## 192 396 0.466353032 0.533646968 1.144298e+00 8.738980e-01  
## 660 396 0.473139408 0.526860592 1.113542e+00 8.980353e-01  
## 1073 397 0.462906235 0.537093765 1.160265e+00 8.618723e-01  
## 3 398 0.466193359 0.533806641 1.145033e+00 8.733375e-01  
## 568 398 0.462023727 0.537976273 1.164391e+00 8.588180e-01  
## 1186 399 0.461419435 0.538580565 1.167226e+00 8.567324e-01  
## 381 400 0.454559491 0.545440509 1.199932e+00 8.333805e-01  
## 411 401 0.452668091 0.547331909 1.209124e+00 8.270449e-01  
## 453 401 0.451877544 0.548122456 1.212989e+00 8.244098e-01  
## 585 403 0.444840521 0.555159479 1.247997e+00 8.012842e-01  
## 759 403 0.445005502 0.554994498 1.247163e+00 8.018197e-01  
## 1014 403 0.448262191 0.551737809 1.230837e+00 8.124551e-01  
## 223 406 0.433931007 0.566068993 1.304514e+00 7.665691e-01  
## 527 408 0.426218375 0.573781625 1.346215e+00 7.428233e-01  
## 716 408 0.429101142 0.570898858 1.330453e+00 7.516238e-01  
## 1166 408 0.429551013 0.570448987 1.328012e+00 7.530051e-01  
## 143 409 0.428505517 0.571494483 1.333692e+00 7.497982e-01  
## 956 409 0.429552108 0.570447892 1.328006e+00 7.530085e-01  
## 33 410 0.427714946 0.572285054 1.338006e+00 7.473809e-01  
## 1009 411 0.421067227 0.578932773 1.374918e+00 7.273163e-01  
## 333 413 0.415874332 0.584125668 1.404573e+00 7.119604e-01  
## 916 414 0.407758237 0.592241763 1.452434e+00 6.884996e-01  
## 108 417 0.397205640 0.602794360 1.517588e+00 6.589405e-01  
## 551 418 0.392539138 0.607460862 1.547517e+00 6.461966e-01  
## 625 418 0.395774834 0.604225166 1.526689e+00 6.550122e-01  
## 950 419 0.393689320 0.606310680 1.540074e+00 6.493195e-01  
## 458 421 0.385618811 0.614381189 1.593234e+00 6.276540e-01  
## 505 421 0.385123311 0.614876689 1.596571e+00 6.263424e-01  
## 1100 421 0.388470866 0.611529134 1.574196e+00 6.352451e-01  
## 852 422 0.378675996 0.621324004 1.640780e+00 6.094662e-01  
## 159 424 0.374844832 0.625155168 1.667771e+00 5.996029e-01  
## 667 424 0.374544942 0.625455058 1.669907e+00 5.988359e-01  
## 817 424 0.374466077 0.625533923 1.670469e+00 5.986343e-01  
## 929 424 0.374346766 0.625653234 1.671320e+00 5.983295e-01  
## 863 425 0.376320643 0.623679357 1.657308e+00 6.033880e-01  
## 6 426 0.375083623 0.624916377 1.666072e+00 6.002141e-01  
## 419 426 0.367830991 0.632169009 1.718640e+00 5.818555e-01  
## 682 426 0.367399877 0.632600123 1.721830e+00 5.807774e-01  
## 140 428 0.369183739 0.630816261 1.708678e+00 5.852477e-01  
## 908 429 0.360634375 0.639365625 1.772892e+00 5.640503e-01  
## 964 430 0.353719768 0.646280232 1.827097e+00 5.473164e-01  
## 1066 430 0.357023135 0.642976865 1.800939e+00 5.552659e-01  
## 774 432 0.350701691 0.649298309 1.851426e+00 5.401241e-01  
## 1189 434 0.343510314 0.656489686 1.911121e+00 5.232532e-01  
## 181 436 0.335798055 0.664201945 1.977980e+00 5.055662e-01  
## 301 436 0.335798055 0.664201945 1.977980e+00 5.055662e-01  
## 869 436 0.335610321 0.664389679 1.979646e+00 5.051408e-01  
## 1011 436 0.336197857 0.663802143 1.974439e+00 5.064730e-01  
## 924 437 0.334732616 0.665267384 1.987459e+00 5.031550e-01  
## 83 438 0.331987587 0.668012413 2.012161e+00 4.969782e-01  
## 451 438 0.331551114 0.668448886 2.016126e+00 4.960007e-01  
## 1056 438 0.329752771 0.670247229 2.032575e+00 4.919868e-01  
## 249 440 0.326428602 0.673571398 2.063457e+00 4.846236e-01  
## 23 441 0.323471558 0.676528442 2.091462e+00 4.781345e-01  
## 189 441 0.322804919 0.677195081 2.097846e+00 4.766794e-01  
## 50 442 0.320568108 0.679431892 2.119462e+00 4.718179e-01  
## 965 442 0.321720687 0.678279313 2.108286e+00 4.743189e-01  
## 214 443 0.319387157 0.680612843 2.130996e+00 4.692641e-01  
## 1019 443 0.317100271 0.682899729 2.153577e+00 4.643438e-01  
## 1025 444 0.313116460 0.686883540 2.193700e+00 4.558509e-01  
## 275 445 0.306882709 0.693117291 2.258574e+00 4.427573e-01  
## 1089 445 0.310949780 0.689050220 2.215953e+00 4.512730e-01  
## 680 446 0.305806162 0.694193838 2.270045e+00 4.405198e-01  
## 910 447 0.306113021 0.693886979 2.266767e+00 4.411569e-01  
## 149 448 0.301122512 0.698877488 2.320907e+00 4.308659e-01  
## 836 448 0.305124448 0.694875552 2.277351e+00 4.391066e-01  
## 265 450 0.294369076 0.705630924 2.397096e+00 4.171714e-01  
## 1185 450 0.293553490 0.706446510 2.406534e+00 4.155353e-01  
## 39 451 0.297541459 0.702458541 2.360876e+00 4.235716e-01  
## 850 451 0.290637745 0.709362255 2.440709e+00 4.097170e-01  
## 28 452 0.289520731 0.710479269 2.453984e+00 4.075006e-01  
## 893 453 0.285352383 0.714647617 2.504439e+00 3.992910e-01  
## 1136 453 0.289128229 0.710871771 2.458673e+00 4.067235e-01  
## 373 454 0.284888637 0.715111363 2.510144e+00 3.983836e-01  
## 580 454 0.286642222 0.713357778 2.488670e+00 4.018211e-01  
## 269 457 0.274968263 0.725031737 2.636783e+00 3.792500e-01  
## 845 457 0.278169444 0.721830556 2.594931e+00 3.853667e-01  
## 286 459 0.269993014 0.730006986 2.703800e+00 3.698499e-01  
## 1176 459 0.272949288 0.727050712 2.663684e+00 3.754199e-01  
## 744 460 0.267803633 0.732196367 2.734079e+00 3.657538e-01  
## 821 460 0.272207575 0.727792425 2.673667e+00 3.740181e-01  
## 378 461 0.267232770 0.732767230 2.742056e+00 3.646898e-01  
## 416 461 0.262472136 0.737527864 2.809928e+00 3.558810e-01  
## 648 464 0.259213142 0.740786858 2.857829e+00 3.499160e-01  
## 231 465 0.254687046 0.745312954 2.926387e+00 3.417183e-01  
## 760 465 0.254477229 0.745522771 2.929625e+00 3.413407e-01  
## 1180 466 0.250817843 0.749182157 2.986957e+00 3.347889e-01  
## 187 467 0.248069821 0.751930179 3.031123e+00 3.299107e-01  
## 128 468 0.246690808 0.753309192 3.053657e+00 3.274762e-01  
## 513 468 0.246743811 0.753256189 3.052787e+00 3.275696e-01  
## 1127 468 0.249362677 0.750637323 3.010223e+00 3.322013e-01  
## 1145 468 0.249362677 0.750637323 3.010223e+00 3.322013e-01  
## 1178 468 0.249362677 0.750637323 3.010223e+00 3.322013e-01  
## 424 469 0.242039289 0.757960711 3.131561e+00 3.193296e-01  
## 1003 470 0.241249776 0.758750224 3.145082e+00 3.179568e-01  
## 282 471 0.241000070 0.758999930 3.149376e+00 3.175232e-01  
## 865 471 0.240185449 0.759814551 3.163450e+00 3.161106e-01  
## 113 472 0.234612251 0.765387749 3.262352e+00 3.065273e-01  
## 263 472 0.236455664 0.763544336 3.229123e+00 3.096816e-01  
## 345 472 0.236455664 0.763544336 3.229123e+00 3.096816e-01  
## 399 472 0.239022613 0.760977387 3.183705e+00 3.140995e-01  
## 1102 472 0.240174819 0.759825181 3.163634e+00 3.160922e-01  
## 330 473 0.232459946 0.767540054 3.301816e+00 3.028636e-01  
## 763 476 0.228048421 0.771951579 3.385034e+00 2.954180e-01  
## 986 476 0.228619541 0.771380459 3.374079e+00 2.963771e-01  
## 785 478 0.224803492 0.775196508 3.448329e+00 2.899955e-01  
## 81 481 0.213735580 0.786264420 3.678678e+00 2.718368e-01  
## 154 481 0.214336171 0.785663829 3.665568e+00 2.728090e-01  
## 210 481 0.213100145 0.786899855 3.692629e+00 2.708097e-01  
## 307 481 0.214336171 0.785663829 3.665568e+00 2.728090e-01  
## 501 481 0.214336171 0.785663829 3.665568e+00 2.728090e-01  
## 671 482 0.213024634 0.786975366 3.694293e+00 2.706878e-01  
## 48 483 0.210411593 0.789588407 3.752590e+00 2.664826e-01  
## 308 483 0.211066130 0.788933870 3.737852e+00 2.675334e-01  
## 713 483 0.209485762 0.790514238 3.773594e+00 2.649994e-01  
## 1096 483 0.210337535 0.789662465 3.754263e+00 2.663639e-01  
## 1114 483 0.208182130 0.791817870 3.803486e+00 2.629167e-01  
## 718 484 0.206355517 0.793644483 3.846006e+00 2.600100e-01  
## 811 484 0.206355517 0.793644483 3.846006e+00 2.600100e-01  
## 1080 484 0.210734090 0.789265910 3.745317e+00 2.670001e-01  
## 706 485 0.206632116 0.793367884 3.839519e+00 2.604493e-01  
## 879 485 0.208124870 0.791875130 3.804808e+00 2.628254e-01  
## 452 487 0.200747082 0.799252918 3.981392e+00 2.511684e-01  
## 1058 487 0.201302638 0.798697362 3.967645e+00 2.520387e-01  
## 318 488 0.198203926 0.801796074 4.045309e+00 2.471999e-01  
## 829 488 0.198801334 0.801198666 4.030147e+00 2.481299e-01  
## 979 488 0.198755995 0.801244005 4.031295e+00 2.480593e-01  
## 391 491 0.190252589 0.809747411 4.256170e+00 2.349530e-01  
## 711 492 0.191844325 0.808155675 4.212560e+00 2.373854e-01  
## 1182 492 0.190866080 0.809133920 4.239276e+00 2.358894e-01  
## 422 493 0.188872051 0.811127949 4.294590e+00 2.328511e-01  
## 1071 493 0.185742954 0.814257046 4.383784e+00 2.281134e-01  
## 14 495 0.184701065 0.815298935 4.414154e+00 2.265440e-01  
## 651 495 0.184411306 0.815588694 4.422661e+00 2.261082e-01  
## 677 495 0.184467762 0.815532238 4.421001e+00 2.261931e-01  
## 968 495 0.184620468 0.815379532 4.416518e+00 2.264227e-01  
## 123 496 0.180871034 0.819128966 4.528801e+00 2.208090e-01  
## 284 496 0.181777902 0.818222098 4.501219e+00 2.221621e-01  
## 100 497 0.180482053 0.819517947 4.540717e+00 2.202295e-01  
## 582 497 0.179784927 0.820215073 4.562202e+00 2.191924e-01  
## 601 497 0.179596049 0.820403951 4.568051e+00 2.189117e-01  
## 535 498 0.178304974 0.821695026 4.608369e+00 2.169965e-01  
## 656 498 0.176309664 0.823690336 4.671839e+00 2.140485e-01  
## 757 499 0.175003408 0.824996592 4.714174e+00 2.121262e-01  
## 855 499 0.174978218 0.825021782 4.714997e+00 2.120892e-01  
## 1029 499 0.174978218 0.825021782 4.714997e+00 2.120892e-01  
## 882 500 0.173942201 0.826057799 4.749036e+00 2.105690e-01  
## 558 501 0.171821777 0.828178223 4.819984e+00 2.074696e-01  
## 613 501 0.174179773 0.825820227 4.741195e+00 2.109173e-01  
## 653 501 0.170792517 0.829207483 4.855057e+00 2.059708e-01  
## 860 501 0.172483078 0.827516922 4.797670e+00 2.084345e-01  
## 876 501 0.172483078 0.827516922 4.797670e+00 2.084345e-01  
## 1012 501 0.173142475 0.826857525 4.775590e+00 2.093982e-01  
## 973 502 0.170279865 0.829720135 4.872685e+00 2.052257e-01  
## 612 504 0.167476346 0.832523654 4.970992e+00 2.011671e-01  
## 634 504 0.166894592 0.833105408 4.991806e+00 2.003283e-01  
## 694 504 0.167476346 0.832523654 4.970992e+00 2.011671e-01  
## 12 505 0.165218209 0.834781791 5.052602e+00 1.979178e-01  
## 155 505 0.162782080 0.837217920 5.143182e+00 1.944321e-01  
## 274 505 0.162775537 0.837224463 5.143429e+00 1.944228e-01  
## 405 505 0.162782080 0.837217920 5.143182e+00 1.944321e-01  
## 414 505 0.162782080 0.837217920 5.143182e+00 1.944321e-01  
## 542 505 0.162782080 0.837217920 5.143182e+00 1.944321e-01  
## 808 506 0.163001897 0.836998103 5.134898e+00 1.947458e-01  
## 920 506 0.163453035 0.836546965 5.117965e+00 1.953901e-01  
## 923 506 0.163203355 0.836796645 5.127325e+00 1.950335e-01  
## 1074 506 0.163345226 0.836654774 5.122003e+00 1.952361e-01  
## 217 507 0.160996963 0.839003037 5.211297e+00 1.918908e-01  
## 230 507 0.160699268 0.839300732 5.222804e+00 1.914680e-01  
## 132 508 0.158329939 0.841670061 5.315925e+00 1.881140e-01  
## 914 508 0.158586213 0.841413787 5.305718e+00 1.884759e-01  
## 727 510 0.155629659 0.844370341 5.425510e+00 1.843145e-01  
## 901 511 0.153536799 0.846463201 5.513097e+00 1.813863e-01  
## 940 511 0.153503445 0.846496555 5.514512e+00 1.813397e-01  
## 137 512 0.151883725 0.848116275 5.583984e+00 1.790836e-01  
## 278 512 0.148729463 0.851270537 5.723617e+00 1.747147e-01  
## 446 513 0.147265301 0.852734699 5.790466e+00 1.726977e-01  
## 1171 513 0.151091453 0.848908547 5.618508e+00 1.779832e-01  
## 1033 514 0.148479451 0.851520549 5.734939e+00 1.743698e-01  
## 288 515 0.146033887 0.853966113 5.847726e+00 1.710067e-01  
## 7 516 0.144545905 0.855454095 5.918217e+00 1.689698e-01  
## 781 516 0.143678731 0.856321269 5.959972e+00 1.677860e-01  
## 1164 516 0.147496993 0.852503007 5.779799e+00 1.730164e-01  
## 702 517 0.142257733 0.857742267 6.029495e+00 1.658514e-01  
## 1026 517 0.142918934 0.857081066 5.996974e+00 1.667508e-01  
## 386 518 0.139348464 0.860651536 6.176254e+00 1.619104e-01  
## 1079 518 0.140401944 0.859598056 6.122409e+00 1.633344e-01  
## 752 519 0.139820445 0.860179555 6.152030e+00 1.625480e-01  
## 816 519 0.139237953 0.860762047 6.181950e+00 1.617613e-01  
## 915 520 0.137686651 0.862313349 6.262868e+00 1.596713e-01  
## 1062 520 0.137686651 0.862313349 6.262868e+00 1.596713e-01  
## 310 521 0.137276722 0.862723278 6.284556e+00 1.591202e-01  
## 460 521 0.137534007 0.862465993 6.270929e+00 1.594660e-01  
## 620 521 0.137276722 0.862723278 6.284556e+00 1.591202e-01  
## 621 522 0.135940828 0.864059172 6.356142e+00 1.573281e-01  
## 1002 522 0.133509129 0.866490871 6.490125e+00 1.540802e-01  
## 186 524 0.132180417 0.867819583 6.565417e+00 1.523132e-01  
## 342 524 0.131317799 0.868682201 6.615114e+00 1.511690e-01  
## 818 524 0.132375700 0.867624300 6.554257e+00 1.525726e-01  
## 1016 524 0.131806251 0.868193749 6.586894e+00 1.518166e-01  
## 85 525 0.131000930 0.868999070 6.633534e+00 1.507492e-01  
## 93 525 0.131000930 0.868999070 6.633534e+00 1.507492e-01  
## 172 525 0.131000930 0.868999070 6.633534e+00 1.507492e-01  
## 59 526 0.129723630 0.870276370 6.708696e+00 1.490603e-01  
## 898 526 0.127506278 0.872493722 6.842751e+00 1.461401e-01  
## 911 527 0.126849937 0.873150063 6.883331e+00 1.452785e-01  
## 298 529 0.123054148 0.876945852 7.126504e+00 1.403213e-01  
## 457 530 0.121640636 0.878359364 7.220937e+00 1.384862e-01  
## 972 530 0.121966180 0.878033820 7.198994e+00 1.389083e-01  
## 302 532 0.117934256 0.882065744 7.479301e+00 1.337023e-01  
## 363 532 0.119407745 0.880592255 7.374666e+00 1.355994e-01  
## 770 532 0.118950070 0.881049930 7.406889e+00 1.350095e-01  
## 793 532 0.119375944 0.880624056 7.376897e+00 1.355583e-01  
## 376 533 0.116678823 0.883321177 7.570536e+00 1.320911e-01  
## 517 533 0.118776217 0.881223783 7.419194e+00 1.347855e-01  
## 854 533 0.117853190 0.882146810 7.485133e+00 1.335982e-01  
## 941 533 0.117853190 0.882146810 7.485133e+00 1.335982e-01  
## 314 534 0.115548471 0.884451529 7.654377e+00 1.306442e-01  
## 998 535 0.114363171 0.885636829 7.744074e+00 1.291310e-01  
## 200 536 0.112674623 0.887325377 7.875113e+00 1.269823e-01  
## 338 536 0.111935358 0.888064642 7.933728e+00 1.260442e-01  
## 1023 536 0.112886797 0.887113203 7.858432e+00 1.272519e-01  
## 1188 536 0.115394740 0.884605260 7.665906e+00 1.304477e-01  
## 19 537 0.110849106 0.889150894 8.021273e+00 1.246685e-01  
## 44 537 0.112891855 0.887108145 7.858035e+00 1.272583e-01  
## 122 537 0.111258277 0.888741723 7.988095e+00 1.251863e-01  
## 283 537 0.111348356 0.888651644 7.980824e+00 1.253003e-01  
## 748 537 0.112955519 0.887044481 7.853042e+00 1.273392e-01  
## 802 537 0.110412353 0.889587647 8.056958e+00 1.241163e-01  
## 636 538 0.109390367 0.890609633 8.141573e+00 1.228264e-01  
## 672 538 0.110517970 0.889482030 8.048302e+00 1.242498e-01  
## 709 538 0.110517970 0.889482030 8.048302e+00 1.242498e-01  
## 872 538 0.111423545 0.888576455 7.974764e+00 1.253956e-01  
## 320 539 0.108733349 0.891266651 8.196810e+00 1.219987e-01  
## 641 539 0.108542099 0.891457901 8.213015e+00 1.217580e-01  
## 1000 539 0.110153804 0.889846196 8.078216e+00 1.237897e-01  
## 153 540 0.108470561 0.891529439 8.219091e+00 1.216680e-01  
## 198 541 0.106348123 0.893651877 8.403081e+00 1.190040e-01  
## 921 541 0.105611912 0.894388088 8.468629e+00 1.180829e-01  
## 1088 541 0.105611912 0.894388088 8.468629e+00 1.180829e-01  
## 164 542 0.107015407 0.892984593 8.344449e+00 1.198401e-01  
## 1109 542 0.105895949 0.894104051 8.443232e+00 1.184381e-01  
## 594 543 0.102893113 0.897106887 8.718823e+00 1.146944e-01  
## 810 543 0.103777399 0.896222601 8.636009e+00 1.157942e-01  
## 1124 543 0.103650377 0.896349623 8.647818e+00 1.156361e-01  
## 1173 543 0.105000859 0.894999141 8.523732e+00 1.173195e-01  
## 844 544 0.103440596 0.896559404 8.667384e+00 1.153751e-01  
## 440 545 0.101187066 0.898812934 8.882686e+00 1.125786e-01  
## 861 545 0.102162169 0.897837831 8.788359e+00 1.137869e-01  
## 946 545 0.102069094 0.897930906 8.797285e+00 1.136714e-01  
## 1010 545 0.101543390 0.898456610 8.848007e+00 1.130198e-01  
## 1152 545 0.102217752 0.897782248 8.783036e+00 1.138558e-01  
## 367 548 0.096415448 0.903584552 9.371782e+00 1.067033e-01  
## 827 549 0.097662865 0.902337135 9.239306e+00 1.082332e-01  
## 1148 549 0.097942078 0.902057922 9.210116e+00 1.085763e-01  
## 27 550 0.095573488 0.904426512 9.463153e+00 1.056730e-01  
## 616 550 0.095113845 0.904886155 9.513716e+00 1.051114e-01  
## 150 551 0.094229895 0.905770105 9.612343e+00 1.040329e-01  
## 182 551 0.094229895 0.905770105 9.612343e+00 1.040329e-01  
## 520 551 0.094229895 0.905770105 9.612343e+00 1.040329e-01  
## 631 551 0.094229895 0.905770105 9.612343e+00 1.040329e-01  
## 841 551 0.093063643 0.906936357 9.745335e+00 1.026132e-01  
## 1051 551 0.095201397 0.904798603 9.504047e+00 1.052183e-01  
## 652 553 0.091225494 0.908774506 9.961848e+00 1.003830e-01  
## 152 554 0.089580724 0.910419276 1.016312e+01 9.839502e-02  
## 466 554 0.090399806 0.909600194 1.006197e+01 9.938411e-02  
## 567 554 0.090399806 0.909600194 1.006197e+01 9.938411e-02  
## 448 555 0.090175883 0.909824117 1.008944e+01 9.911353e-02  
## 674 555 0.089711175 0.910288825 1.014688e+01 9.855243e-02  
## 698 555 0.090101399 0.909898601 1.009861e+01 9.902356e-02  
## 945 555 0.089653235 0.910346765 1.015409e+01 9.848251e-02  
## 953 555 0.089653235 0.910346765 1.015409e+01 9.848251e-02  
## 995 555 0.089392318 0.910607682 1.018664e+01 9.816776e-02  
## 1154 555 0.090390112 0.909609888 1.006316e+01 9.937239e-02  
## 11 556 0.088636528 0.911363472 1.028203e+01 9.725706e-02  
## 179 556 0.087684878 0.912315122 1.040447e+01 9.611249e-02  
## 388 556 0.089089639 0.910910361 1.022465e+01 9.780286e-02  
## 737 556 0.088548434 0.911451566 1.029325e+01 9.715100e-02  
## 740 556 0.088461581 0.911538419 1.030434e+01 9.704646e-02  
## 1125 556 0.090472338 0.909527662 1.005310e+01 9.947178e-02  
## 481 557 0.086533593 0.913466407 1.055621e+01 9.473101e-02  
## 131 558 0.084628808 0.915371192 1.081631e+01 9.245300e-02  
## 611 558 0.086246455 0.913753545 1.059468e+01 9.438700e-02  
## 678 558 0.086927371 0.913072629 1.050386e+01 9.520313e-02  
## 890 558 0.086246455 0.913753545 1.059468e+01 9.438700e-02  
## 280 559 0.085168136 0.914831864 1.074148e+01 9.309704e-02  
## 824 559 0.085013724 0.914986276 1.076281e+01 9.291257e-02  
## 427 560 0.083953355 0.916046645 1.091138e+01 9.164747e-02  
## 762 560 0.083906245 0.916093755 1.091806e+01 9.159133e-02  
## 873 560 0.083741784 0.916258216 1.094147e+01 9.139540e-02  
## 942 560 0.083812620 0.916187380 1.093138e+01 9.147978e-02  
## 1111 560 0.083812620 0.916187380 1.093138e+01 9.147978e-02  
## 778 561 0.083004916 0.916995084 1.104748e+01 9.051839e-02  
## 996 561 0.083036646 0.916963354 1.104288e+01 9.055612e-02  
## 1118 561 0.084191751 0.915808249 1.087765e+01 9.193164e-02  
## 261 562 0.081989038 0.918010962 1.119675e+01 8.931161e-02  
## 306 562 0.082132661 0.917867339 1.117542e+01 8.948206e-02  
## 927 562 0.082657451 0.917342549 1.109812e+01 9.010533e-02  
## 997 562 0.083952271 0.916047729 1.091153e+01 9.164618e-02  
## 626 563 0.080340483 0.919659517 1.144703e+01 8.735894e-02  
## 825 563 0.081360187 0.918639813 1.129102e+01 8.856593e-02  
## 1018 563 0.080573924 0.919426076 1.141096e+01 8.763502e-02  
## 775 565 0.079060968 0.920939032 1.164847e+01 8.584821e-02  
## 947 565 0.079249274 0.920750726 1.161841e+01 8.607028e-02  
## 436 566 0.076665360 0.923334640 1.204370e+01 8.303096e-02  
## 522 566 0.076665360 0.923334640 1.204370e+01 8.303096e-02  
## 62 567 0.077230136 0.922769864 1.194831e+01 8.369382e-02  
## 139 567 0.077565476 0.922434524 1.189233e+01 8.408779e-02  
## 162 567 0.077565476 0.922434524 1.189233e+01 8.408779e-02  
## 168 567 0.076438330 0.923561670 1.208244e+01 8.276473e-02  
## 197 567 0.077033703 0.922966297 1.198133e+01 8.346318e-02  
## 432 567 0.077033703 0.922966297 1.198133e+01 8.346318e-02  
## 509 567 0.077565476 0.922434524 1.189233e+01 8.408779e-02  
## 443 568 0.075587969 0.924412031 1.222962e+01 8.176870e-02  
## 589 568 0.074984994 0.925015006 1.233600e+01 8.106354e-02  
## 635 568 0.074984994 0.925015006 1.233600e+01 8.106354e-02  
## 471 569 0.075535078 0.924464922 1.223888e+01 8.170681e-02  
## 493 569 0.074654452 0.925345548 1.239505e+01 8.067738e-02  
## 804 569 0.074486021 0.925513979 1.242534e+01 8.048071e-02  
## 903 570 0.075053295 0.924946705 1.232387e+01 8.114337e-02  
## 76 571 0.073904169 0.926095831 1.253104e+01 7.980186e-02  
## 309 571 0.073700103 0.926299897 1.256850e+01 7.956398e-02  
## 24 572 0.072298189 0.927701811 1.283161e+01 7.793257e-02  
## 874 572 0.071519495 0.928480505 1.298220e+01 7.702854e-02  
## 316 574 0.069841093 0.930158907 1.331822e+01 7.508512e-02  
## 508 574 0.070630732 0.929369268 1.315814e+01 7.599857e-02  
## 669 574 0.070964145 0.929035855 1.309162e+01 7.638472e-02  
## 746 574 0.070964145 0.929035855 1.309162e+01 7.638472e-02  
## 877 574 0.069937195 0.930062805 1.329854e+01 7.519621e-02  
## 1 575 0.070496368 0.929503632 1.318513e+01 7.584303e-02  
## 596 575 0.070614680 0.929385320 1.316136e+01 7.597998e-02  
## 1162 576 0.068957124 0.931042876 1.350176e+01 7.406439e-02  
## 104 577 0.067863270 0.932136730 1.373551e+01 7.280399e-02  
## 291 577 0.067403261 0.932596739 1.383608e+01 7.227482e-02  
## 1168 577 0.067601162 0.932398838 1.379265e+01 7.250241e-02  
## 185 578 0.067674633 0.932325367 1.377659e+01 7.258693e-02  
## 77 579 0.066726522 0.933273478 1.398654e+01 7.149729e-02  
## 133 579 0.065415539 0.934584461 1.428689e+01 6.999425e-02  
## 233 579 0.065415539 0.934584461 1.428689e+01 6.999425e-02  
## 273 579 0.065097494 0.934902506 1.436157e+01 6.963025e-02  
## 292 579 0.066239576 0.933760424 1.409672e+01 7.093851e-02  
## 299 579 0.066726522 0.933273478 1.398654e+01 7.149729e-02  
## 380 579 0.065415539 0.934584461 1.428689e+01 6.999425e-02  
## 571 579 0.065339459 0.934660541 1.430469e+01 6.990715e-02  
## 839 579 0.066104697 0.933895303 1.412752e+01 7.078384e-02  
## 1156 579 0.065486248 0.934513752 1.427038e+01 7.007521e-02  
## 171 580 0.065470840 0.934529160 1.427398e+01 7.005757e-02  
## 679 580 0.065145253 0.934854747 1.435031e+01 6.968489e-02  
## 1092 580 0.064815129 0.935184871 1.442850e+01 6.930729e-02  
## 1104 580 0.064815129 0.935184871 1.442850e+01 6.930729e-02  
## 151 581 0.064433849 0.935566151 1.451979e+01 6.887150e-02  
## 238 581 0.064566931 0.935433069 1.448780e+01 6.902357e-02  
## 632 581 0.064440669 0.935559331 1.451815e+01 6.887930e-02  
## 783 582 0.063289295 0.936710705 1.480046e+01 6.756547e-02  
## 1191 582 0.063938963 0.936061037 1.463992e+01 6.830640e-02  
## 222 583 0.062106887 0.937893113 1.510127e+01 6.621958e-02  
## 447 583 0.061494641 0.938505359 1.526158e+01 6.552402e-02  
## 604 583 0.061494641 0.938505359 1.526158e+01 6.552402e-02  
## 767 583 0.063737786 0.936262214 1.468928e+01 6.807685e-02  
## 976 585 0.061337372 0.938662628 1.530327e+01 6.534549e-02  
## 1020 585 0.061740407 0.938259593 1.519685e+01 6.580312e-02  
## 47 586 0.060713791 0.939286209 1.547072e+01 6.463822e-02  
## 142 586 0.060469530 0.939530470 1.553725e+01 6.436144e-02  
## 734 586 0.060227532 0.939772468 1.560370e+01 6.408736e-02  
## 830 586 0.060048395 0.939951605 1.565323e+01 6.388456e-02  
## 46 587 0.060243308 0.939756692 1.559935e+01 6.410522e-02  
## 304 587 0.058705166 0.941294834 1.603428e+01 6.236640e-02  
## 326 587 0.058958436 0.941041564 1.596110e+01 6.265232e-02  
## 328 587 0.059852447 0.940147553 1.570775e+01 6.366282e-02  
## 627 587 0.059852447 0.940147553 1.570775e+01 6.366282e-02  
## 1116 587 0.059795901 0.940204099 1.572355e+01 6.359885e-02  
## 1117 587 0.059795901 0.940204099 1.572355e+01 6.359885e-02  
## 1158 587 0.059795901 0.940204099 1.572355e+01 6.359885e-02  
## 1179 587 0.059795901 0.940204099 1.572355e+01 6.359885e-02  
## 173 588 0.059160294 0.940839706 1.590323e+01 6.288031e-02  
## 858 588 0.058892068 0.941107932 1.598022e+01 6.257738e-02  
## 859 588 0.058892068 0.941107932 1.598022e+01 6.257738e-02  
## 650 589 0.057578055 0.942421945 1.636773e+01 6.109583e-02  
## 657 589 0.057500418 0.942499582 1.639118e+01 6.100843e-02  
## 1161 589 0.057891026 0.942108974 1.627383e+01 6.144833e-02  
## 8 590 0.057613338 0.942386662 1.635709e+01 6.113556e-02  
## 175 590 0.056700206 0.943299794 1.663662e+01 6.010836e-02  
## 226 590 0.056700206 0.943299794 1.663662e+01 6.010836e-02  
## 334 590 0.056700206 0.943299794 1.663662e+01 6.010836e-02  
## 103 591 0.057383130 0.942616870 1.642672e+01 6.087641e-02  
## 1050 591 0.056747762 0.943252238 1.662184e+01 6.016181e-02  
## 1122 591 0.056885930 0.943114070 1.657904e+01 6.031713e-02  
## 218 592 0.055102249 0.944897751 1.714808e+01 5.831557e-02  
## 640 593 0.056389031 0.943610969 1.673395e+01 5.975877e-02  
## 733 593 0.055936127 0.944063873 1.687753e+01 5.925036e-02  
## 822 593 0.056102379 0.943897621 1.682456e+01 5.943693e-02  
## 866 594 0.054936406 0.945063594 1.720287e+01 5.812985e-02  
## 918 595 0.053786428 0.946213572 1.759205e+01 5.684386e-02  
## 971 595 0.053786428 0.946213572 1.759205e+01 5.684386e-02  
## 1075 595 0.053334299 0.946665701 1.774966e+01 5.633911e-02  
## 1151 596 0.053441263 0.946558737 1.771213e+01 5.645847e-02  
## 129 597 0.051891927 0.948108073 1.827082e+01 5.473208e-02  
## 461 597 0.051774580 0.948225420 1.831450e+01 5.460155e-02  
## 614 597 0.052382316 0.947617684 1.809041e+01 5.527790e-02  
## 1044 597 0.052542193 0.947457807 1.803232e+01 5.545597e-02  
## 347 598 0.051052061 0.948947939 1.858785e+01 5.379859e-02  
## 794 598 0.051772640 0.948227360 1.831522e+01 5.459940e-02  
## 1169 598 0.051148577 0.948851423 1.855089e+01 5.390578e-02  
## 751 599 0.051074023 0.948925977 1.857942e+01 5.382298e-02  
## 337 600 0.049505236 0.950494764 1.919988e+01 5.208365e-02  
## 1083 600 0.050624733 0.949375267 1.875319e+01 5.332426e-02  
## 1143 600 0.050130860 0.949869140 1.894779e+01 5.277660e-02  
## 957 601 0.049906822 0.950093178 1.903734e+01 5.252834e-02  
## 1028 601 0.049906822 0.950093178 1.903734e+01 5.252834e-02  
## 80 602 0.049730360 0.950269640 1.910844e+01 5.233289e-02  
## 322 602 0.049283585 0.950716415 1.929073e+01 5.183837e-02  
## 430 602 0.048250658 0.951749342 1.972511e+01 5.069681e-02  
## 95 603 0.048348574 0.951651426 1.968313e+01 5.080492e-02  
## 161 603 0.048348574 0.951651426 1.968313e+01 5.080492e-02  
## 224 603 0.047579845 0.952420155 2.001730e+01 4.995678e-02  
## 541 603 0.048031844 0.951968156 1.981952e+01 5.045531e-02  
## 624 603 0.047928425 0.952071575 1.986445e+01 5.034120e-02  
## 792 603 0.047998505 0.952001495 1.983398e+01 5.041852e-02  
## 250 604 0.047124077 0.952875923 2.022057e+01 4.945458e-02  
## 293 604 0.047467795 0.952532205 2.006691e+01 4.983327e-02  
## 372 604 0.047574422 0.952425578 2.001970e+01 4.995080e-02  
## 591 605 0.046804324 0.953195676 2.036555e+01 4.910254e-02  
## 742 605 0.046713645 0.953286355 2.040702e+01 4.900274e-02  
## 370 606 0.046170789 0.953829211 2.065872e+01 4.840572e-02  
## 478 606 0.046169002 0.953830998 2.065955e+01 4.840376e-02  
## 518 606 0.046167569 0.953832431 2.066023e+01 4.840218e-02  
## 615 606 0.046436278 0.953563722 2.053489e+01 4.869761e-02  
## 708 606 0.046577314 0.953422686 2.046968e+01 4.885274e-02  
## 922 606 0.046436278 0.953563722 2.053489e+01 4.869761e-02  
## 928 606 0.046712538 0.953287462 2.040753e+01 4.900152e-02  
## 944 606 0.046436278 0.953563722 2.053489e+01 4.869761e-02  
## 962 606 0.046712538 0.953287462 2.040753e+01 4.900152e-02  
## 1121 606 0.046266292 0.953733708 2.061401e+01 4.851070e-02  
## 468 607 0.045368913 0.954631087 2.104152e+01 4.752507e-02  
## 496 607 0.045368913 0.954631087 2.104152e+01 4.752507e-02  
## 992 607 0.046198139 0.953801861 2.064589e+01 4.843578e-02  
## 1129 607 0.046033073 0.953966927 2.072351e+01 4.825437e-02  
## 203 608 0.044487816 0.955512184 2.147806e+01 4.655913e-02  
## 581 609 0.044439898 0.955560102 2.150230e+01 4.650665e-02  
## 697 609 0.044453334 0.955546666 2.149550e+01 4.652136e-02  
## 779 609 0.044453334 0.955546666 2.149550e+01 4.652136e-02  
## 208 610 0.043423262 0.956576738 2.202913e+01 4.539444e-02  
## 220 610 0.043423262 0.956576738 2.202913e+01 4.539444e-02  
## 681 610 0.043630274 0.956369726 2.191987e+01 4.562072e-02  
## 880 611 0.042854987 0.957145013 2.233451e+01 4.477377e-02  
## 1006 611 0.042854987 0.957145013 2.233451e+01 4.477377e-02  
## 1054 611 0.043782742 0.956217258 2.184005e+01 4.578744e-02  
## 1131 611 0.043375931 0.956624069 2.205426e+01 4.534271e-02  
## 20 612 0.043068399 0.956931601 2.221888e+01 4.500677e-02  
## 57 612 0.043068399 0.956931601 2.221888e+01 4.500677e-02  
## 371 612 0.042659482 0.957340518 2.244145e+01 4.456041e-02  
## 788 612 0.042761550 0.957238450 2.238549e+01 4.467178e-02  
## 339 613 0.042557589 0.957442411 2.249757e+01 4.444924e-02  
## 356 613 0.042149863 0.957850137 2.272487e+01 4.400465e-02  
## 428 613 0.042567772 0.957432228 2.249195e+01 4.446035e-02  
## 695 613 0.042125132 0.957874868 2.273880e+01 4.397770e-02  
## 397 614 0.041316344 0.958683656 2.320350e+01 4.309695e-02  
## 410 614 0.041846016 0.958153984 2.289714e+01 4.367358e-02  
## 415 614 0.041792553 0.958207447 2.292771e+01 4.361535e-02  
## 84 615 0.041356477 0.958643523 2.318001e+01 4.314062e-02  
## 124 615 0.041356477 0.958643523 2.318001e+01 4.314062e-02  
## 134 615 0.041309370 0.958690630 2.320758e+01 4.308936e-02  
## 270 615 0.041309370 0.958690630 2.320758e+01 4.308936e-02  
## 305 615 0.041356477 0.958643523 2.318001e+01 4.314062e-02  
## 771 615 0.040770844 0.959229156 2.352733e+01 4.250376e-02  
## 144 616 0.040004606 0.959995394 2.399712e+01 4.167166e-02  
## 193 616 0.040738362 0.959261638 2.354689e+01 4.246846e-02  
## 227 616 0.039999923 0.960000077 2.400005e+01 4.166658e-02  
## 268 616 0.040342216 0.959657784 2.378793e+01 4.203813e-02  
## 395 616 0.040004606 0.959995394 2.399712e+01 4.167166e-02  
## 536 616 0.040202443 0.959797557 2.387411e+01 4.188638e-02  
## 593 616 0.039999923 0.960000077 2.400005e+01 4.166658e-02  
## 595 616 0.040342216 0.959657784 2.378793e+01 4.203813e-02  
## 1170 617 0.039770222 0.960229778 2.414444e+01 4.141740e-02  
## 45 618 0.039996277 0.960003723 2.400233e+01 4.166263e-02  
## 714 618 0.039146071 0.960853929 2.454535e+01 4.074092e-02  
## 768 618 0.039586823 0.960413177 2.426093e+01 4.121853e-02  
## 820 618 0.039146071 0.960853929 2.454535e+01 4.074092e-02  
## 319 619 0.038980491 0.961019509 2.465386e+01 4.056160e-02  
## 565 619 0.039007378 0.960992622 2.463618e+01 4.059072e-02  
## 579 619 0.038514428 0.961485572 2.496430e+01 4.005721e-02  
## 597 619 0.039211410 0.960788590 2.450278e+01 4.081169e-02  
## 704 619 0.039354095 0.960645905 2.441032e+01 4.096629e-02  
## 715 619 0.038990388 0.961009612 2.464735e+01 4.057232e-02  
## 796 619 0.038514428 0.961485572 2.496430e+01 4.005721e-02  
## 426 620 0.038382026 0.961617974 2.505386e+01 3.991401e-02  
## 696 620 0.038859622 0.961140378 2.473365e+01 4.043075e-02  
## 848 620 0.038459674 0.961540326 2.500126e+01 3.999798e-02  
## 932 620 0.038056973 0.961943027 2.527639e+01 3.956261e-02  
## 987 620 0.038406532 0.961593468 2.503724e+01 3.994051e-02  
## 1004 620 0.038056973 0.961943027 2.527639e+01 3.956261e-02  
## 1147 621 0.037590133 0.962409867 2.560273e+01 3.905834e-02  
## 1160 621 0.038590739 0.961409261 2.491295e+01 4.013976e-02  
## 248 622 0.036915210 0.963084790 2.608910e+01 3.833018e-02  
## 856 622 0.037736373 0.962263627 2.549963e+01 3.921625e-02  
## 239 623 0.037254267 0.962745733 2.584256e+01 3.869585e-02  
## 473 623 0.036802883 0.963197117 2.617178e+01 3.820909e-02  
## 692 623 0.037281007 0.962718993 2.582331e+01 3.872470e-02  
## 753 623 0.037281007 0.962718993 2.582331e+01 3.872470e-02  
## 834 623 0.037025171 0.962974829 2.600865e+01 3.844874e-02  
## 1090 623 0.037390300 0.962609700 2.574490e+01 3.884264e-02  
## 228 624 0.036682056 0.963317944 2.626129e+01 3.807887e-02  
## 538 624 0.036975571 0.963024429 2.604488e+01 3.839526e-02  
## 575 624 0.036413056 0.963586944 2.646268e+01 3.778907e-02  
## 665 624 0.036520447 0.963479553 2.638192e+01 3.790475e-02  
## 1048 624 0.036511087 0.963488913 2.638894e+01 3.789466e-02  
## 277 625 0.035908855 0.964091145 2.684828e+01 3.724633e-02  
## 739 625 0.035488493 0.964511507 2.717815e+01 3.679427e-02  
## 784 625 0.035488493 0.964511507 2.717815e+01 3.679427e-02  
## 576 626 0.035696946 0.964303054 2.701360e+01 3.701839e-02  
## 728 626 0.035579115 0.964420885 2.710638e+01 3.689169e-02  
## 797 626 0.035579115 0.964420885 2.710638e+01 3.689169e-02  
## 985 626 0.035441822 0.964558178 2.721525e+01 3.674410e-02  
## 994 626 0.035643952 0.964356048 2.705525e+01 3.696140e-02  
## 1041 626 0.035508198 0.964491802 2.716251e+01 3.681545e-02  
## 116 627 0.034906497 0.965093503 2.764796e+01 3.616903e-02  
## 394 627 0.035246738 0.964753262 2.737142e+01 3.653446e-02  
## 401 627 0.035246738 0.964753262 2.737142e+01 3.653446e-02  
## 724 627 0.034830228 0.965169772 2.771069e+01 3.608715e-02  
## 71 628 0.034976795 0.965023205 2.759038e+01 3.624451e-02  
## 209 628 0.034502707 0.965497293 2.798323e+01 3.573569e-02  
## 425 628 0.034502707 0.965497293 2.798323e+01 3.573569e-02  
## 803 628 0.034777792 0.965222208 2.775398e+01 3.603087e-02  
## 1126 628 0.034682892 0.965317108 2.783266e+01 3.592901e-02  
## 1132 628 0.034897653 0.965102347 2.765522e+01 3.615954e-02  
## 191 629 0.034089094 0.965910906 2.833490e+01 3.529217e-02  
## 212 629 0.033877216 0.966122784 2.851836e+01 3.506513e-02  
## 219 629 0.033877700 0.966122300 2.851794e+01 3.506564e-02  
## 279 629 0.033838811 0.966161189 2.855187e+01 3.502398e-02  
## 290 629 0.034089094 0.965910906 2.833490e+01 3.529217e-02  
## 335 629 0.034117149 0.965882851 2.831077e+01 3.532224e-02  
## 417 629 0.034089094 0.965910906 2.833490e+01 3.529217e-02  
## 488 629 0.034089094 0.965910906 2.833490e+01 3.529217e-02  
## 563 629 0.034604028 0.965395972 2.789837e+01 3.584439e-02  
## 577 629 0.034117149 0.965882851 2.831077e+01 3.532224e-02  
## 603 629 0.034117149 0.965882851 2.831077e+01 3.532224e-02  
## 881 629 0.034049076 0.965950924 2.836937e+01 3.524928e-02  
## 892 629 0.034329807 0.965670193 2.812921e+01 3.555024e-02  
## 1034 629 0.034049076 0.965950924 2.836937e+01 3.524928e-02  
## 125 630 0.033840805 0.966159195 2.855012e+01 3.502612e-02  
## 237 630 0.033840805 0.966159195 2.855012e+01 3.502612e-02  
## 459 630 0.033840805 0.966159195 2.855012e+01 3.502612e-02  
## 465 630 0.033840805 0.966159195 2.855012e+01 3.502612e-02  
## 477 630 0.033840805 0.966159195 2.855012e+01 3.502612e-02  
## 528 630 0.033396137 0.966603863 2.894358e+01 3.454997e-02  
## 966 630 0.034136151 0.965863849 2.829446e+01 3.534261e-02  
## 967 630 0.033906802 0.966093198 2.849261e+01 3.509682e-02  
## 99 631 0.032896070 0.967103930 2.939877e+01 3.401503e-02  
## 190 631 0.033125434 0.966874566 2.918828e+01 3.426032e-02  
## 207 631 0.033117549 0.966882451 2.919547e+01 3.425189e-02  
## 260 631 0.033096359 0.966903641 2.921480e+01 3.422922e-02  
## 491 631 0.033246669 0.966753331 2.907820e+01 3.439002e-02  
## 549 631 0.033096359 0.966903641 2.921480e+01 3.422922e-02  
## 618 631 0.033225594 0.966774406 2.909728e+01 3.436747e-02  
## 884 631 0.033237322 0.966762678 2.908666e+01 3.438002e-02  
## 1063 631 0.033237322 0.966762678 2.908666e+01 3.438002e-02  
## 1072 631 0.033424840 0.966575160 2.891787e+01 3.458069e-02  
## 225 632 0.032475926 0.967524074 2.979204e+01 3.356601e-02  
## 364 632 0.032475926 0.967524074 2.979204e+01 3.356601e-02  
## 455 632 0.032475926 0.967524074 2.979204e+01 3.356601e-02  
## 511 632 0.032475926 0.967524074 2.979204e+01 3.356601e-02  
## 838 632 0.032761037 0.967238963 2.952406e+01 3.387068e-02  
## 851 633 0.032444574 0.967555426 2.982179e+01 3.353252e-02  
## 993 633 0.032206229 0.967793771 3.004989e+01 3.327799e-02  
## 1130 633 0.032959232 0.967040768 2.934051e+01 3.408257e-02  
## 234 634 0.032078618 0.967921382 3.017341e+01 3.314176e-02  
## 445 634 0.032078618 0.967921382 3.017341e+01 3.314176e-02  
## 537 634 0.031733155 0.968266845 3.051278e+01 3.277315e-02  
## 543 634 0.032216602 0.967783398 3.003990e+01 3.328906e-02  
## 726 634 0.031936617 0.968063383 3.031202e+01 3.299021e-02  
## 789 634 0.032292935 0.967707065 2.996653e+01 3.337057e-02  
## 843 634 0.031844332 0.968155668 3.040276e+01 3.289175e-02  
## 853 634 0.031742165 0.968257835 3.050384e+01 3.278276e-02  
## 97 635 0.031605134 0.968394866 3.064043e+01 3.263662e-02  
## 246 635 0.031605134 0.968394866 3.064043e+01 3.263662e-02  
## 687 635 0.031609214 0.968390786 3.063634e+01 3.264097e-02  
## 717 635 0.032358608 0.967641392 2.990368e+01 3.344070e-02  
## 255 636 0.031088687 0.968911313 3.116604e+01 3.208621e-02  
## 666 636 0.031329245 0.968670755 3.091906e+01 3.234251e-02  
## 951 636 0.031265141 0.968734859 3.098450e+01 3.227420e-02  
## 984 636 0.031323021 0.968676979 3.092540e+01 3.233588e-02  
## 1175 636 0.031007406 0.968992594 3.125036e+01 3.199963e-02  
## 63 637 0.031199483 0.968800517 3.105181e+01 3.220424e-02  
## 194 637 0.030765188 0.969234812 3.150427e+01 3.174173e-02  
## 287 637 0.030765188 0.969234812 3.150427e+01 3.174173e-02  
## 617 637 0.030670735 0.969329265 3.160437e+01 3.164119e-02  
## 285 638 0.030120105 0.969879895 3.220041e+01 3.105550e-02  
## 390 638 0.030242014 0.969757986 3.206658e+01 3.118511e-02  
## 564 638 0.030399067 0.969600933 3.189575e+01 3.135214e-02  
## 926 638 0.029912463 0.970087537 3.243088e+01 3.083481e-02  
## 1049 638 0.029912463 0.970087537 3.243088e+01 3.083481e-02  
## 1015 639 0.030032782 0.969967218 3.229695e+01 3.096268e-02  
## 609 640 0.029398859 0.970601141 3.301493e+01 3.028933e-02  
## 889 640 0.029482926 0.970517074 3.291794e+01 3.037858e-02  
## 202 641 0.029163084 0.970836916 3.328993e+01 3.003912e-02  
## 262 641 0.029271980 0.970728020 3.316236e+01 3.015467e-02  
## 281 641 0.029271980 0.970728020 3.316236e+01 3.015467e-02  
## 297 641 0.029059816 0.970940184 3.341178e+01 2.992956e-02  
## 429 641 0.029059816 0.970940184 3.341178e+01 2.992956e-02  
## 526 641 0.029059816 0.970940184 3.341178e+01 2.992956e-02  
## 556 641 0.029006507 0.970993493 3.347502e+01 2.987302e-02  
## 831 641 0.028983600 0.971016400 3.350227e+01 2.984872e-02  
## 907 641 0.029400942 0.970599058 3.301252e+01 3.029154e-02  
## 1039 641 0.029400942 0.970599058 3.301252e+01 3.029154e-02  
## 1108 641 0.028853724 0.971146276 3.365757e+01 2.971100e-02  
## 1157 641 0.029348366 0.970651634 3.307345e+01 3.023574e-02  
## 321 642 0.029331695 0.970668305 3.309281e+01 3.021804e-02  
## 346 642 0.029331695 0.970668305 3.309281e+01 3.021804e-02  
## 379 642 0.028775869 0.971224131 3.375134e+01 2.962845e-02  
## 795 642 0.028241520 0.971758480 3.440886e+01 2.906228e-02  
## 895 643 0.028509417 0.971490583 3.407613e+01 2.934606e-02  
## 1032 643 0.028577978 0.971422022 3.399198e+01 2.941870e-02  
## 135 644 0.027956989 0.972043011 3.476923e+01 2.876106e-02  
## 241 644 0.027956989 0.972043011 3.476923e+01 2.876106e-02  
## 289 644 0.027956989 0.972043011 3.476923e+01 2.876106e-02  
## 368 644 0.027797175 0.972202825 3.497488e+01 2.859195e-02  
## 913 644 0.027920009 0.972079991 3.481661e+01 2.872193e-02  
## 1165 644 0.027953845 0.972046155 3.477325e+01 2.875773e-02  
## 163 645 0.027626550 0.972373450 3.519706e+01 2.841146e-02  
## 180 645 0.027408581 0.972591419 3.548492e+01 2.818098e-02  
## 276 645 0.027408581 0.972591419 3.548492e+01 2.818098e-02  
## 325 645 0.027408581 0.972591419 3.548492e+01 2.818098e-02  
## 406 645 0.027408581 0.972591419 3.548492e+01 2.818098e-02  
## 646 645 0.027408581 0.972591419 3.548492e+01 2.818098e-02  
## 887 645 0.027581999 0.972418001 3.525553e+01 2.836434e-02  
## 1082 645 0.027565497 0.972434503 3.527723e+01 2.834689e-02  
## 1087 645 0.027263645 0.972736355 3.567888e+01 2.802778e-02  
## 1098 645 0.027581999 0.972418001 3.525553e+01 2.836434e-02  
## 141 646 0.027176782 0.972823218 3.579612e+01 2.793599e-02  
## 519 646 0.027176782 0.972823218 3.579612e+01 2.793599e-02  
## 719 646 0.026889012 0.973110988 3.618991e+01 2.763201e-02  
## 787 646 0.026889012 0.973110988 3.618991e+01 2.763201e-02  
## 498 647 0.026870054 0.973129946 3.621615e+01 2.761199e-02  
## 685 647 0.027123816 0.972876184 3.586797e+01 2.788003e-02  
## 691 647 0.027251264 0.972748736 3.569554e+01 2.801470e-02  
## 939 647 0.026297412 0.973702588 3.702655e+01 2.700764e-02  
## 66 648 0.027149746 0.972850254 3.583276e+01 2.790743e-02  
## 92 648 0.027149746 0.972850254 3.583276e+01 2.790743e-02  
## 147 648 0.027149746 0.972850254 3.583276e+01 2.790743e-02  
## 444 648 0.026263826 0.973736174 3.707518e+01 2.697222e-02  
## 454 648 0.026456132 0.973543868 3.679842e+01 2.717508e-02  
## 489 648 0.026245280 0.973754720 3.710209e+01 2.695266e-02  
## 503 648 0.026432967 0.973567033 3.683155e+01 2.715064e-02  
## 516 648 0.026432967 0.973567033 3.683155e+01 2.715064e-02  
## 722 648 0.027105856 0.972894144 3.589240e+01 2.786105e-02  
## 1045 648 0.026294076 0.973705924 3.703138e+01 2.700412e-02  
## 146 649 0.026241012 0.973758988 3.710829e+01 2.694816e-02  
## 199 649 0.025789596 0.974210404 3.777533e+01 2.647231e-02  
## 242 649 0.025789596 0.974210404 3.777533e+01 2.647231e-02  
## 323 649 0.026436637 0.973563363 3.682629e+01 2.715451e-02  
## 369 649 0.025960669 0.974039331 3.751981e+01 2.665259e-02  
## 439 649 0.025893389 0.974106611 3.761990e+01 2.658168e-02  
## 572 649 0.025789596 0.974210404 3.777533e+01 2.647231e-02  
## 600 649 0.025893389 0.974106611 3.761990e+01 2.658168e-02  
## 374 650 0.025927564 0.974072436 3.756899e+01 2.661770e-02  
## 418 650 0.026148746 0.973851254 3.724275e+01 2.685086e-02  
## 470 650 0.025927564 0.974072436 3.756899e+01 2.661770e-02  
## 553 650 0.026148746 0.973851254 3.724275e+01 2.685086e-02  
## 684 650 0.025872225 0.974127775 3.765149e+01 2.655938e-02  
## 721 650 0.025872225 0.974127775 3.765149e+01 2.655938e-02  
## 981 651 0.025616026 0.974383974 3.803806e+01 2.628946e-02  
## 1076 651 0.025473407 0.974526593 3.825663e+01 2.613926e-02  
## 229 652 0.024995567 0.975004433 3.900709e+01 2.563636e-02  
## 396 652 0.024961481 0.975038519 3.906172e+01 2.560051e-02  
## 888 652 0.025275111 0.974724889 3.856461e+01 2.593051e-02  
## 991 652 0.024892173 0.975107827 3.917327e+01 2.552761e-02  
## 1042 652 0.025275111 0.974724889 3.856461e+01 2.593051e-02  
## 204 653 0.024918763 0.975081237 3.913040e+01 2.555558e-02  
## 431 653 0.024623600 0.975376400 3.961145e+01 2.524523e-02  
## 435 653 0.024414404 0.975585596 3.995943e+01 2.502538e-02  
## 550 653 0.024539584 0.975460416 3.975049e+01 2.515692e-02  
## 644 653 0.025138102 0.974861898 3.878025e+01 2.578632e-02  
## 699 653 0.025138102 0.974861898 3.878025e+01 2.578632e-02  
## 720 653 0.024635795 0.975364205 3.959134e+01 2.525805e-02  
## 989 653 0.024854685 0.975145315 3.923386e+01 2.548819e-02  
## 358 654 0.024015947 0.975984053 4.063900e+01 2.460690e-02  
## 408 654 0.024015947 0.975984053 4.063900e+01 2.460690e-02  
## 510 654 0.024455804 0.975544196 3.989009e+01 2.506888e-02  
## 514 654 0.024746707 0.975253293 3.940942e+01 2.537465e-02  
## 622 654 0.024015947 0.975984053 4.063900e+01 2.460690e-02  
## 385 656 0.023410355 0.976589645 4.171614e+01 2.397154e-02  
## 766 656 0.023529054 0.976470946 4.150065e+01 2.409601e-02  
## 303 657 0.023548267 0.976451733 4.146597e+01 2.411616e-02  
## 456 657 0.023609483 0.976390517 4.135586e+01 2.418037e-02  
## 469 657 0.023671612 0.976328388 4.124469e+01 2.424554e-02  
## 532 657 0.023761956 0.976238044 4.108408e+01 2.434033e-02  
## 936 657 0.023580334 0.976419666 4.140822e+01 2.414979e-02  
## 1144 657 0.023777799 0.976222201 4.105604e+01 2.435695e-02  
## 74 658 0.023149702 0.976850298 4.219710e+01 2.369831e-02  
## 101 658 0.023149702 0.976850298 4.219710e+01 2.369831e-02  
## 117 658 0.023149702 0.976850298 4.219710e+01 2.369831e-02  
## 434 658 0.023149702 0.976850298 4.219710e+01 2.369831e-02  
## 490 658 0.023149702 0.976850298 4.219710e+01 2.369831e-02  
## 566 658 0.023105223 0.976894777 4.228026e+01 2.365170e-02  
## 705 658 0.023492757 0.976507243 4.156631e+01 2.405794e-02  
## 413 659 0.022819781 0.977180219 4.282163e+01 2.335268e-02  
## 500 659 0.022976431 0.977023569 4.252286e+01 2.351676e-02  
## 515 659 0.023126994 0.976873006 4.223951e+01 2.367451e-02  
## 776 659 0.022881579 0.977118421 4.270328e+01 2.341741e-02  
## 1040 659 0.022932507 0.977067493 4.260622e+01 2.347075e-02  
## 393 661 0.022479935 0.977520065 4.348411e+01 2.299690e-02  
## 1070 661 0.022225193 0.977774807 4.399398e+01 2.273038e-02  
## 329 662 0.021869519 0.978130481 4.472574e+01 2.235849e-02  
## 826 662 0.021984986 0.978015014 4.448559e+01 2.247919e-02  
## 883 662 0.022228963 0.977771037 4.398635e+01 2.273432e-02  
## 959 662 0.022117557 0.977882443 4.421295e+01 2.261781e-02  
## 96 663 0.021961248 0.978038752 4.453475e+01 2.245437e-02  
## 98 663 0.022129259 0.977870741 4.418904e+01 2.263005e-02  
## 244 663 0.021887009 0.978112991 4.468920e+01 2.237677e-02  
## 327 663 0.021865891 0.978134109 4.473333e+01 2.235470e-02  
## 331 663 0.021738457 0.978261543 4.500143e+01 2.222152e-02  
## 383 663 0.021887009 0.978112991 4.468920e+01 2.237677e-02  
## 402 663 0.021887009 0.978112991 4.468920e+01 2.237677e-02  
## 421 663 0.021887009 0.978112991 4.468920e+01 2.237677e-02  
## 442 663 0.021331259 0.978668741 4.587956e+01 2.179620e-02  
## 659 663 0.022105922 0.977894078 4.423675e+01 2.260564e-02  
## 68 664 0.021655912 0.978344088 4.517677e+01 2.213527e-02  
## 82 664 0.021655912 0.978344088 4.517677e+01 2.213527e-02  
## 87 664 0.021655912 0.978344088 4.517677e+01 2.213527e-02  
## 710 664 0.021530296 0.978469704 4.544618e+01 2.200405e-02  
## 756 664 0.021541989 0.978458011 4.542097e+01 2.201626e-02  
## 769 664 0.021541989 0.978458011 4.542097e+01 2.201626e-02  
## 70 665 0.021642941 0.978357059 4.520444e+01 2.212172e-02  
## 90 665 0.021642941 0.978357059 4.520444e+01 2.212172e-02  
## 341 665 0.020676263 0.979323737 4.736464e+01 2.111280e-02  
## 360 665 0.020676263 0.979323737 4.736464e+01 2.111280e-02  
## 1184 665 0.021178229 0.978821771 4.621830e+01 2.163645e-02  
## 449 666 0.020765982 0.979234018 4.715568e+01 2.120635e-02  
## 484 666 0.020821441 0.979178559 4.702742e+01 2.126419e-02  
## 590 666 0.020338292 0.979661708 4.816834e+01 2.076053e-02  
## 662 666 0.020573928 0.979426072 4.760520e+01 2.100611e-02  
## 754 666 0.020573928 0.979426072 4.760520e+01 2.100611e-02  
## 952 666 0.020762876 0.979237124 4.716289e+01 2.120311e-02  
## 487 667 0.020897782 0.979102218 4.685197e+01 2.134382e-02  
## 689 667 0.020601795 0.979398205 4.753946e+01 2.103516e-02  
## 700 667 0.020409865 0.979590135 4.799591e+01 2.083511e-02  
## 195 668 0.020132132 0.979867868 4.867184e+01 2.054576e-02  
## 486 669 0.019908486 0.980091514 4.922984e+01 2.031288e-02  
## 1135 669 0.020207623 0.979792377 4.848628e+01 2.062439e-02  
## 148 670 0.019388435 0.980611565 5.057714e+01 1.977178e-02  
## 294 670 0.019388435 0.980611565 5.057714e+01 1.977178e-02  
## 315 670 0.019388435 0.980611565 5.057714e+01 1.977178e-02  
## 377 670 0.019746508 0.980253492 4.964186e+01 2.014429e-02  
## 531 670 0.019504740 0.980495260 5.026959e+01 1.989274e-02  
## 561 670 0.019626320 0.980373680 4.995199e+01 2.001922e-02  
## 786 670 0.019771815 0.980228185 4.957705e+01 2.017062e-02  
## 900 670 0.019824518 0.980175482 4.944259e+01 2.022548e-02  
## 1078 670 0.019784814 0.980215186 4.954382e+01 2.018415e-02  
## 1084 670 0.019824518 0.980175482 4.944259e+01 2.022548e-02  
## 69 671 0.019544186 0.980455814 5.016611e+01 1.993378e-02  
## 158 671 0.019544186 0.980455814 5.016611e+01 1.993378e-02  
## 438 671 0.019343281 0.980656719 5.069754e+01 1.972482e-02  
## 741 671 0.019471897 0.980528103 5.035606e+01 1.985858e-02  
## 205 672 0.019195890 0.980804110 5.109448e+01 1.957158e-02  
## 772 672 0.019124861 0.980875139 5.128796e+01 1.949775e-02  
## 773 672 0.019189060 0.980810940 5.111303e+01 1.956448e-02  
## 777 672 0.019331722 0.980668278 5.072845e+01 1.971280e-02  
## 259 673 0.019043734 0.980956266 5.151071e+01 1.941344e-02  
## 499 673 0.018600081 0.981399919 5.276321e+01 1.895260e-02  
## 828 673 0.018900442 0.981099558 5.190881e+01 1.926455e-02  
## 332 674 0.018653446 0.981346554 5.260940e+01 1.900801e-02  
## 398 674 0.018342242 0.981657758 5.351896e+01 1.868497e-02  
## 524 674 0.018800532 0.981199468 5.218998e+01 1.916077e-02  
## 552 674 0.018342242 0.981657758 5.351896e+01 1.868497e-02  
## 933 674 0.018655657 0.981344343 5.260304e+01 1.901031e-02  
## 1057 674 0.018730877 0.981269123 5.238778e+01 1.908842e-02  
## 196 675 0.018397048 0.981602952 5.335655e+01 1.874184e-02  
## 213 675 0.018396161 0.981603839 5.335917e+01 1.874092e-02  
## 354 675 0.018367142 0.981632858 5.344505e+01 1.871081e-02  
## 355 675 0.018385937 0.981614063 5.338939e+01 1.873031e-02  
## 463 675 0.018396161 0.981603839 5.335917e+01 1.874092e-02  
## 977 675 0.018588961 0.981411039 5.279537e+01 1.894106e-02  
## 365 676 0.018074118 0.981925882 5.432773e+01 1.840680e-02  
## 930 676 0.018374315 0.981625685 5.342380e+01 1.871825e-02  
## 954 676 0.018247340 0.981752660 5.380251e+01 1.858649e-02  
## 403 677 0.017988720 0.982011280 5.459039e+01 1.831824e-02  
## 1065 677 0.018138748 0.981861252 5.413060e+01 1.847384e-02  
## 266 678 0.017554072 0.982445928 5.596684e+01 1.786772e-02  
## 529 678 0.017628482 0.982371518 5.572638e+01 1.794482e-02  
## 1095 678 0.017350763 0.982649237 5.663435e+01 1.765713e-02  
## 1097 678 0.017350763 0.982649237 5.663435e+01 1.765713e-02  
## 72 679 0.017535225 0.982464775 5.602807e+01 1.784820e-02  
## 375 679 0.017501340 0.982498660 5.613848e+01 1.781309e-02  
## 1103 679 0.017667915 0.982332085 5.559977e+01 1.798568e-02  
## 272 680 0.017171060 0.982828940 5.723752e+01 1.747106e-02  
## 352 680 0.017411634 0.982588366 5.643286e+01 1.772017e-02  
## 610 680 0.017131722 0.982868278 5.737125e+01 1.743033e-02  
## 119 681 0.016829577 0.983170423 5.841920e+01 1.711766e-02  
## 494 681 0.016821043 0.983178957 5.844935e+01 1.710883e-02  
## 975 681 0.016609922 0.983390078 5.920498e+01 1.689047e-02  
## 412 682 0.016444294 0.983555706 5.981137e+01 1.671923e-02  
## 1024 682 0.016701103 0.983298897 5.887629e+01 1.698477e-02  
## 1099 682 0.016802639 0.983197361 5.851446e+01 1.708979e-02  
## 245 683 0.016628227 0.983371773 5.913870e+01 1.690940e-02  
## 361 683 0.016379420 0.983620580 6.005222e+01 1.665217e-02  
## 1038 683 0.016571572 0.983428428 5.934431e+01 1.685082e-02  
## 4 684 0.016549374 0.983450626 5.942525e+01 1.682786e-02  
## 58 684 0.016549374 0.983450626 5.942525e+01 1.682786e-02  
## 799 684 0.016110836 0.983889164 6.107003e+01 1.637465e-02  
## 389 685 0.015962031 0.984037969 6.164867e+01 1.622095e-02  
## 480 685 0.015962031 0.984037969 6.164867e+01 1.622095e-02  
## 606 685 0.015962031 0.984037969 6.164867e+01 1.622095e-02  
## 668 685 0.016096538 0.983903462 6.112516e+01 1.635988e-02  
## 317 686 0.015808638 0.984191362 6.225656e+01 1.606257e-02  
## 336 687 0.015394918 0.984605082 6.395650e+01 1.563563e-02  
## 607 687 0.015703400 0.984296600 6.268048e+01 1.595393e-02  
## 690 687 0.015550639 0.984449361 6.330604e+01 1.579628e-02  
## 764 687 0.015568233 0.984431767 6.323336e+01 1.581444e-02  
## 157 688 0.015487196 0.984512804 6.356947e+01 1.573082e-02  
## 539 688 0.015487196 0.984512804 6.356947e+01 1.573082e-02  
## 1036 690 0.014705143 0.985294857 6.700342e+01 1.492461e-02  
## 1061 690 0.015158137 0.984841863 6.497117e+01 1.539144e-02  
## 1123 690 0.014782142 0.985217858 6.664920e+01 1.500393e-02  
## 18 691 0.014813545 0.985186455 6.650579e+01 1.503629e-02  
## 38 691 0.014813545 0.985186455 6.650579e+01 1.503629e-02  
## 109 691 0.014577945 0.985422055 6.759677e+01 1.479361e-02  
## 256 691 0.014915202 0.985084798 6.604569e+01 1.514103e-02  
## 450 691 0.014577945 0.985422055 6.759677e+01 1.479361e-02  
## 384 692 0.014532859 0.985467141 6.780959e+01 1.474718e-02  
## 464 692 0.014623910 0.985376090 6.738117e+01 1.484094e-02  
## 525 692 0.014303883 0.985696117 6.891109e+01 1.451145e-02  
## 857 692 0.014763509 0.985236491 6.673457e+01 1.498474e-02  
## 955 692 0.014763509 0.985236491 6.673457e+01 1.498474e-02  
## 476 693 0.014409429 0.985590571 6.839900e+01 1.462010e-02  
## 730 694 0.013991983 0.986008017 7.046950e+01 1.419054e-02  
## 847 694 0.013991983 0.986008017 7.046950e+01 1.419054e-02  
## 73 695 0.013884957 0.986115043 7.102039e+01 1.408046e-02  
## 251 695 0.013884957 0.986115043 7.102039e+01 1.408046e-02  
## 806 695 0.013865278 0.986134722 7.112261e+01 1.406023e-02  
## 569 696 0.013791933 0.986208067 7.150615e+01 1.398481e-02  
## 758 696 0.013863496 0.986136504 7.113188e+01 1.405839e-02  
## 782 696 0.013863496 0.986136504 7.113188e+01 1.405839e-02  
## 1059 696 0.014021353 0.985978647 7.031979e+01 1.422075e-02  
## 1067 696 0.013865784 0.986134216 7.111998e+01 1.406075e-02  
## 555 697 0.013422651 0.986577349 7.350093e+01 1.360527e-02  
## 252 698 0.013271482 0.986728518 7.434953e+01 1.344998e-02  
## 586 698 0.013432176 0.986567824 7.344810e+01 1.361506e-02  
## 623 698 0.013145132 0.986854868 7.507379e+01 1.332023e-02  
## 736 698 0.013235826 0.986764174 7.455252e+01 1.341336e-02  
## 833 698 0.013639765 0.986360235 7.231505e+01 1.382838e-02  
## 184 699 0.013238198 0.986761802 7.453898e+01 1.341580e-02  
## 999 699 0.013123803 0.986876197 7.519743e+01 1.329833e-02  
## 423 700 0.013184336 0.986815664 7.484758e+01 1.336049e-02  
## 578 700 0.013032723 0.986967277 7.572993e+01 1.320482e-02  
## 1101 700 0.013097277 0.986902723 7.535175e+01 1.327109e-02  
## 382 701 0.012889879 0.987110121 7.658024e+01 1.305820e-02  
## 534 701 0.012889879 0.987110121 7.658024e+01 1.305820e-02  
## 805 701 0.012908068 0.987091932 7.647093e+01 1.307687e-02  
## 1110 701 0.012903858 0.987096142 7.649620e+01 1.307254e-02  
## 562 702 0.012772946 0.987227054 7.729048e+01 1.293820e-02  
## 362 703 0.012560760 0.987439240 7.861302e+01 1.272054e-02  
## 387 703 0.012560760 0.987439240 7.861302e+01 1.272054e-02  
## 1064 703 0.012398156 0.987601844 7.965716e+01 1.255380e-02  
## 1085 703 0.012398156 0.987601844 7.965716e+01 1.255380e-02  
## 745 704 0.012285757 0.987714243 8.039507e+01 1.243857e-02  
## 812 704 0.012332263 0.987667737 8.008812e+01 1.248625e-02  
## 359 705 0.012329507 0.987670493 8.010624e+01 1.248342e-02  
## 647 705 0.012468113 0.987531887 7.920460e+01 1.262553e-02  
## 145 706 0.012105433 0.987894567 8.160754e+01 1.225377e-02  
## 712 706 0.011807375 0.988192625 8.369283e+01 1.194846e-02  
## 637 707 0.011828111 0.988171889 8.354435e+01 1.196969e-02  
## 1140 707 0.011992417 0.988007583 8.238603e+01 1.213798e-02  
## 312 708 0.011579407 0.988420593 8.536021e+01 1.171506e-02  
## 404 708 0.011678653 0.988321347 8.462631e+01 1.181666e-02  
## 886 709 0.011656225 0.988343775 8.479107e+01 1.179370e-02  
## 1013 709 0.011806871 0.988193129 8.369644e+01 1.194794e-02  
## 1043 709 0.011722035 0.988277965 8.430942e+01 1.186107e-02  
## 1159 709 0.011623471 0.988376529 8.503282e+01 1.176016e-02  
## 86 710 0.011364388 0.988635612 8.699418e+01 1.149502e-02  
## 177 710 0.011364388 0.988635612 8.699418e+01 1.149502e-02  
## 366 710 0.011380534 0.988619466 8.686934e+01 1.151154e-02  
## 441 710 0.011380534 0.988619466 8.686934e+01 1.151154e-02  
## 729 710 0.011521521 0.988478479 8.579409e+01 1.165581e-02  
## 969 710 0.011637584 0.988362416 8.492849e+01 1.177461e-02  
## 1077 710 0.011443161 0.988556839 8.638844e+01 1.157562e-02  
## 94 711 0.011358383 0.988641617 8.704070e+01 1.148888e-02  
## 160 711 0.011358383 0.988641617 8.704070e+01 1.148888e-02  
## 165 711 0.011358383 0.988641617 8.704070e+01 1.148888e-02  
## 88 713 0.010987354 0.989012646 9.001372e+01 1.110942e-02  
## 111 713 0.010987354 0.989012646 9.001372e+01 1.110942e-02  
## 504 713 0.010981529 0.989018471 9.006200e+01 1.110346e-02  
## 521 713 0.010986024 0.989013976 9.002474e+01 1.110806e-02  
## 557 713 0.010850234 0.989149766 9.116391e+01 1.096925e-02  
## 570 713 0.010850234 0.989149766 9.116391e+01 1.096925e-02  
## 743 713 0.010953273 0.989046727 9.029692e+01 1.107458e-02  
## 747 713 0.011109321 0.988890679 8.901451e+01 1.123412e-02  
## 798 714 0.010926276 0.989073724 9.052249e+01 1.104698e-02  
## 605 715 0.010598887 0.989401113 9.334953e+01 1.071243e-02  
## 703 715 0.010832080 0.989167920 9.131837e+01 1.095070e-02  
## 761 715 0.010650970 0.989349030 9.288817e+01 1.076563e-02  
## 791 715 0.010693642 0.989306358 9.251351e+01 1.080923e-02  
## 813 715 0.010574477 0.989425523 9.356732e+01 1.068749e-02  
## 353 717 0.010324672 0.989675328 9.585538e+01 1.043238e-02  
## 547 718 0.010092423 0.989907577 9.808424e+01 1.019532e-02  
## 732 718 0.010152015 0.989847985 9.750261e+01 1.025614e-02  
## 188 719 0.010167343 0.989832657 9.735411e+01 1.027178e-02  
## 583 719 0.010167343 0.989832657 9.735411e+01 1.027178e-02  
## 300 720 0.010021325 0.989978675 9.878721e+01 1.012277e-02  
## 530 720 0.010035798 0.989964202 9.864329e+01 1.013754e-02  
## 545 720 0.010021325 0.989978675 9.878721e+01 1.012277e-02  
## 628 720 0.010035798 0.989964202 9.864329e+01 1.013754e-02  
## 919 720 0.009941127 0.990058873 9.959221e+01 1.004095e-02  
## 925 720 0.009941127 0.990058873 9.959221e+01 1.004095e-02  
## 961 720 0.010028383 0.989971617 9.871697e+01 1.012997e-02  
## 1060 720 0.010028383 0.989971617 9.871697e+01 1.012997e-02  
## 1069 720 0.009941127 0.990058873 9.959221e+01 1.004095e-02  
## 254 721 0.009745379 0.990254621 1.016127e+02 9.841286e-03  
## 433 721 0.009729344 0.990270656 1.017819e+02 9.824934e-03  
## 602 721 0.009786441 0.990213559 1.011822e+02 9.883162e-03  
## 642 721 0.009845648 0.990154352 1.005677e+02 9.943549e-03  
## 958 721 0.009942840 0.990057160 9.957489e+01 1.004269e-02  
## 801 723 0.009617621 0.990382379 1.029758e+02 9.711018e-03  
## 1086 723 0.009610789 0.990389211 1.030497e+02 9.704053e-03  
## 1106 723 0.009610789 0.990389211 1.030497e+02 9.704053e-03  
## 943 724 0.009491440 0.990508560 1.043581e+02 9.582391e-03  
## 1031 724 0.009328269 0.990671731 1.062010e+02 9.416105e-03  
## 815 725 0.009251946 0.990748054 1.070854e+02 9.338344e-03  
## 392 726 0.009142405 0.990857595 1.083804e+02 9.226760e-03  
## 544 726 0.009082071 0.990917929 1.091070e+02 9.165311e-03  
## 584 726 0.009082071 0.990917929 1.091070e+02 9.165311e-03  
## 800 726 0.009279206 0.990720794 1.067678e+02 9.366116e-03  
## 835 726 0.009081300 0.990918700 1.091164e+02 9.164526e-03  
## 166 727 0.009057562 0.990942438 1.094050e+02 9.140351e-03  
## 832 727 0.009133906 0.990866094 1.084822e+02 9.218104e-03  
## 136 728 0.008939683 0.991060317 1.108608e+02 9.020322e-03  
## 649 728 0.009102766 0.990897234 1.088567e+02 9.186388e-03  
## 723 729 0.008835792 0.991164208 1.121761e+02 8.914559e-03  
## 258 730 0.008801434 0.991198566 1.126178e+02 8.879587e-03  
## 598 731 0.008480328 0.991519672 1.169200e+02 8.552859e-03  
## 169 732 0.008367802 0.991632198 1.185057e+02 8.438413e-03  
## 236 732 0.008354312 0.991645688 1.186987e+02 8.424695e-03  
## 349 732 0.008354312 0.991645688 1.186987e+02 8.424695e-03  
## 725 732 0.008367802 0.991632198 1.185057e+02 8.438413e-03  
## 840 732 0.008354383 0.991645617 1.186977e+02 8.424766e-03  
## 167 733 0.008167794 0.991832206 1.214321e+02 8.235056e-03  
## 462 733 0.008213329 0.991786671 1.207533e+02 8.281346e-03  
## 1113 733 0.008396019 0.991603981 1.181041e+02 8.467109e-03  
## 474 734 0.008271737 0.991728263 1.198936e+02 8.340730e-03  
## 475 734 0.008137982 0.991862018 1.218806e+02 8.204752e-03  
## 482 735 0.008080033 0.991919967 1.227619e+02 8.145852e-03  
## 156 737 0.007937308 0.992062692 1.249873e+02 8.000812e-03  
## 183 737 0.007937308 0.992062692 1.249873e+02 8.000812e-03  
## 295 738 0.007793044 0.992206956 1.273196e+02 7.854253e-03  
## 554 738 0.007751578 0.992248422 1.280060e+02 7.812134e-03  
## 896 738 0.007773593 0.992226407 1.276406e+02 7.834495e-03  
## 707 739 0.007577549 0.992422451 1.309688e+02 7.635406e-03  
## 765 739 0.007577549 0.992422451 1.309688e+02 7.635406e-03  
## 127 742 0.007284720 0.992715280 1.362736e+02 7.338176e-03  
## 655 742 0.007345189 0.992654811 1.351435e+02 7.399540e-03  
## 814 742 0.007269822 0.992730178 1.365549e+02 7.323060e-03  
## 1139 742 0.007342470 0.992657530 1.351940e+02 7.396780e-03  
## 221 744 0.007166882 0.992833118 1.385307e+02 7.218617e-03  
## 348 744 0.007166882 0.992833118 1.385307e+02 7.218617e-03  
## 560 744 0.007138685 0.992861315 1.390818e+02 7.190012e-03  
## 654 744 0.007138685 0.992861315 1.390818e+02 7.190012e-03  
## 842 744 0.007112349 0.992887651 1.396005e+02 7.163296e-03  
## 949 744 0.007187316 0.992812684 1.381340e+02 7.239348e-03  
## 483 745 0.007031931 0.992968069 1.412085e+02 7.081729e-03  
## 502 745 0.007031931 0.992968069 1.412085e+02 7.081729e-03  
## 904 745 0.007078031 0.992921969 1.402822e+02 7.128487e-03  
## 311 747 0.006901106 0.993098894 1.439043e+02 6.949062e-03  
## 1068 747 0.006953926 0.993046074 1.428036e+02 7.002622e-03  
## 661 749 0.006687757 0.993312243 1.485270e+02 6.732784e-03  
## 79 750 0.006543916 0.993456084 1.518137e+02 6.587021e-03  
## 115 750 0.006525075 0.993474925 1.522550e+02 6.567931e-03  
## 683 751 0.006578433 0.993421567 1.510119e+02 6.621995e-03  
## 750 751 0.006578433 0.993421567 1.510119e+02 6.621995e-03  
## 437 752 0.006367250 0.993632750 1.560537e+02 6.408052e-03  
## 485 755 0.006142921 0.993857079 1.617890e+02 6.180889e-03  
## 823 755 0.006214759 0.993785241 1.599073e+02 6.253624e-03  
## 344 756 0.006047159 0.993952841 1.643669e+02 6.083950e-03  
## 587 758 0.005948244 0.994051756 1.671168e+02 5.983837e-03  
## 492 759 0.005831648 0.994168352 1.704781e+02 5.865856e-03  
## 479 763 0.005487003 0.994512997 1.812489e+02 5.517276e-03  
## 512 763 0.005487003 0.994512997 1.812489e+02 5.517276e-03  
## 407 766 0.005271865 0.994728135 1.886862e+02 5.299805e-03  
## 420 766 0.005300028 0.994699972 1.876782e+02 5.328268e-03  
## 559 767 0.005189863 0.994810137 1.916833e+02 5.216938e-03  
## 1001 767 0.005150849 0.994849151 1.931427e+02 5.177518e-03  
## 409 768 0.005123013 0.994876987 1.941976e+02 5.149394e-03  
## 937 768 0.005149233 0.994850767 1.932037e+02 5.175885e-03  
## 1105 768 0.005149233 0.994850767 1.932037e+02 5.175885e-03  
## 574 771 0.004884620 0.995115380 2.037242e+02 4.908596e-03  
## 599 771 0.004884620 0.995115380 2.037242e+02 4.908596e-03  
## 357 772 0.004825009 0.995174991 2.062535e+02 4.848403e-03  
## 400 772 0.004825009 0.995174991 2.062535e+02 4.848403e-03  
## 472 772 0.004814729 0.995185271 2.066960e+02 4.838023e-03  
## 588 773 0.004738502 0.995261498 2.100372e+02 4.761062e-03  
## 693 773 0.004738502 0.995261498 2.100372e+02 4.761062e-03  
## 176 776 0.004679618 0.995320382 2.126927e+02 4.701619e-03  
## 206 776 0.004679618 0.995320382 2.126927e+02 4.701619e-03  
## 1094 776 0.004645564 0.995354436 2.142591e+02 4.667246e-03  
## 523 781 0.004204757 0.995795243 2.368259e+02 4.222511e-03  
## 107 782 0.004245480 0.995754520 2.345446e+02 4.263581e-03  
## 1093 796 0.003498198 0.996501802 2.848615e+02 3.510478e-03

# CHẠY LẠI MÔ HÌNH VỚI DANH SÁCH BIẾN MỚI  
  
# Bước 0: (Giả định bạn đã chạy các bước trước đó và có đối tượng 'bins' và 'train\_woe')  
  
# ====================================================================  
# Bước 1: Xem IV của các biến (SỬA ĐỔI CHÍNH XÁC CHO PHIÊN BẢN 0.4.4)  
# ====================================================================  
  
# Tạo một dataframe rỗng để chứa tổng hợp IV  
iv\_summary <- data.frame(  
 variable = character(),  
 iv = numeric(),  
 stringsAsFactors = FALSE  
)  
  
# Lặp qua từng biến trong đối tượng 'bins' để trích xuất IV  
for (var\_name in names(bins)) {  
 # Mỗi phần tử trong 'bins' là một data.table con  
 var\_bin\_info <- bins[[var\_name]]  
  
 # Đảm bảo dataframe không rỗng và có cột 'total\_iv'  
 if (nrow(var\_bin\_info) > 0 && "total\_iv" %in% names(var\_bin\_info)) {  
 # Lấy giá trị total\_iv từ HÀNG ĐẦU TIÊN (hoặc bất kỳ hàng nào, vì nó lặp lại)  
 current\_iv <- var\_bin\_info$total\_iv[1]  
  
 # Thêm vào dataframe tổng hợp  
 iv\_summary <- rbind(iv\_summary, data.frame(variable = var\_name, iv = current\_iv))  
 }  
}  
  
print(iv\_summary)

## variable iv  
## 1 REASON 0.006870878  
## 2 JOB 0.042882597  
## 3 LOAN 0.174467862  
## 4 VALUE 0.238382973  
## 5 YOJ 0.085394147  
## 6 DEROG 0.340486483  
## 7 DELINQ 0.592419972  
## 8 CLAGE 0.223957304  
## 9 NINQ 0.178724706  
## 10 CLNO 0.086255386  
## 11 DEBTINC 1.921393985

# Sắp xếp theo IV giảm dần để dễ nhìn  
iv\_summary\_sorted <- iv\_summary %>%  
 arrange(desc(iv))  
print(iv\_summary\_sorted)

## variable iv  
## 1 DEBTINC 1.921393985  
## 2 DELINQ 0.592419972  
## 3 DEROG 0.340486483  
## 4 VALUE 0.238382973  
## 5 CLAGE 0.223957304  
## 6 NINQ 0.178724706  
## 7 LOAN 0.174467862  
## 8 CLNO 0.086255386  
## 9 YOJ 0.085394147  
## 10 JOB 0.042882597  
## 11 REASON 0.006870878

# ====================================================================  
# Các bước tiếp theo (Lọc biến, xây dựng lại mô hình, v.v.)  
# giữ nguyên như hướng dẫn trước đó:  
# ====================================================================  
  
# Đặt ngưỡng IV  
iv\_threshold <- 0.1 # Ví dụ: bỏ tất cả các biến có IV < 0.02  
  
# Lọc ra các biến có IV lớn hơn hoặc bằng ngưỡng  
selected\_vars\_iv <- iv\_summary\_sorted %>%  
 filter(iv >= iv\_threshold) %>%  
 pull(variable) # Lấy tên các biến  
  
cat("\nCác biến được chọn dựa trên IV (ngưỡng >=", iv\_threshold, "):\n")

##   
## Các biến được chọn dựa trên IV (ngưỡng >= 0.1 ):

print(selected\_vars\_iv)

## [1] "DEBTINC" "DELINQ" "DEROG" "VALUE" "CLAGE" "NINQ" "LOAN"

# ====================================================================  
# Bước 3: Xây dựng lại công thức mô hình với các biến đã chọn  
# ====================================================================  
  
# Đảm bảo các biến đã chọn là các biến WoE  
selected\_woe\_vars <- paste0(selected\_vars\_iv, "\_woe")  
  
# Xây dựng công thức mô hình mới  
formula\_str\_new <- paste("BAD ~", paste(selected\_woe\_vars, collapse = " + "))  
model\_formula\_new <- as.formula(formula\_str\_new)  
  
cat("\nCông thức mô hình Logistic Regression MỚI:\n")

##   
## Công thức mô hình Logistic Regression MỚI:

print(model\_formula\_new)

## BAD ~ DEBTINC\_woe + DELINQ\_woe + DEROG\_woe + VALUE\_woe + CLAGE\_woe +   
## NINQ\_woe + LOAN\_woe

# ====================================================================  
# Bước 4: Huấn luyện lại mô hình Logistic Regression  
# ====================================================================  
  
logistic\_model\_new <- glm(model\_formula\_new, data = train\_woe, family = binomial(link = "logit"))  
  
# Xem tóm tắt mô hình mới  
summary(logistic\_model\_new)

##   
## Call:  
## glm(formula = model\_formula\_new, family = binomial(link = "logit"),   
## data = train\_woe)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.36987 0.05097 -26.876 < 2e-16 \*\*\*  
## DEBTINC\_woe 0.92088 0.03358 27.422 < 2e-16 \*\*\*  
## DELINQ\_woe 0.91222 0.06254 14.585 < 2e-16 \*\*\*  
## DEROG\_woe 0.67009 0.08242 8.130 4.29e-16 \*\*\*  
## VALUE\_woe 0.80084 0.10162 7.881 3.25e-15 \*\*\*  
## CLAGE\_woe 1.11565 0.10766 10.362 < 2e-16 \*\*\*  
## NINQ\_woe 0.44101 0.11498 3.836 0.000125 \*\*\*  
## LOAN\_woe 0.50115 0.11642 4.305 1.67e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 4767.8 on 4768 degrees of freedom  
## Residual deviance: 2769.0 on 4761 degrees of freedom  
## AIC: 2785  
##   
## Number of Fisher Scoring iterations: 6

# ====================================================================  
# Bước 5: Đánh giá mô hình mới (trên tập Test)  
# ====================================================================  
  
# Dự đoán xác suất "bad" trên tập kiểm tra với mô hình mới  
test\_pred\_prob\_new <- predict(logistic\_model\_new, newdata = test\_woe, type = "response")  
  
# Chuyển đổi xác suất thành dự đoán lớp (ngưỡng 0.5)  
test\_pred\_class\_new <- ifelse(test\_pred\_prob\_new > 0.5, "1", "0")  
test\_pred\_class\_new <- as.factor(test\_pred\_class\_new)  
  
# Tạo confusion matrix  
cat("\nConfusion Matrix cho mô hình MỚI:\n")

##   
## Confusion Matrix cho mô hình MỚI:

confusionMatrix(test\_pred\_class\_new, test\_woe$BAD, positive = "1")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 911 98  
## 1 43 139  
##   
## Accuracy : 0.8816   
## 95% CI : (0.8619, 0.8994)  
## No Information Rate : 0.801   
## P-Value [Acc > NIR] : 9.178e-14   
##   
## Kappa : 0.5932   
##   
## Mcnemar's Test P-Value : 5.426e-06   
##   
## Sensitivity : 0.5865   
## Specificity : 0.9549   
## Pos Pred Value : 0.7637   
## Neg Pred Value : 0.9029   
## Prevalence : 0.1990   
## Detection Rate : 0.1167   
## Detection Prevalence : 0.1528   
## Balanced Accuracy : 0.7707   
##   
## 'Positive' Class : 1   
##

# Tính toán ROC curve và AUC  
library(pROC)  
roc\_obj\_new <- roc(response = test\_woe$BAD, predictor = test\_pred\_prob\_new)

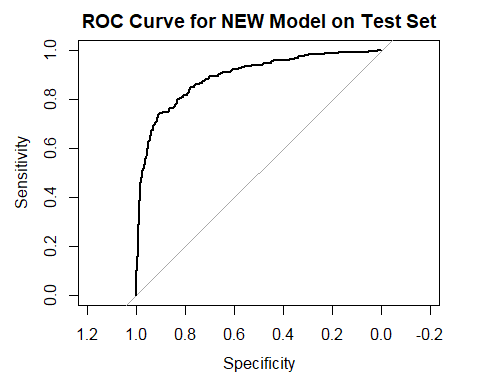
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

cat("\nAUC cho mô hình MỚI:", auc(roc\_obj\_new), "\n")

##   
## AUC cho mô hình MỚI: 0.8920446

plot(roc\_obj\_new, main = "ROC Curve for NEW Model on Test Set")



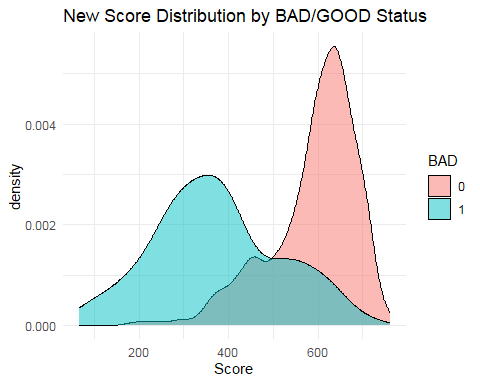
# ====================================================================  
# Bước 6: Tính điểm Scorecard MỚI (nếu cần)  
# ====================================================================  
  
# Lọc 'bins' chỉ bao gồm các biến được chọn  
bins\_selected <- bins[selected\_vars\_iv]  
  
# Tạo scorecard mới với mô hình đã huấn luyện lại và bins đã lọc  
card\_new <- scorecard(bins\_selected,   
 logistic\_model\_new,  
 points0 = 600,   
 pdo = 50,   
 odds0 = 1/19)  
print(card\_new)

## $basepoints  
## variable bin woe points  
## <char> <lgcl> <lgcl> <num>  
## 1: basepoints NA NA 486  
##   
## $DEBTINC  
## variable bin count count\_distr neg pos posprob woe  
## <char> <char> <int> <num> <int> <int> <num> <num>  
## 1: DEBTINC missing 1017 0.21325225 385 632 0.62143559 1.8843011  
## 2: DEBTINC [-Inf,40) 3032 0.63577270 2852 180 0.05936675 -1.3741639  
## 3: DEBTINC [40,42) 434 0.09100440 393 41 0.09447005 -0.8715825  
## 4: DEBTINC [42, Inf) 286 0.05997064 187 99 0.34615385 0.7526663  
## bin\_iv total\_iv breaks is\_special\_values points  
## <num> <num> <char> <lgcl> <num>  
## 1: 1.06086338 1.921394 missing TRUE -125  
## 2: 0.76693190 1.921394 40 FALSE 91  
## 3: 0.05220188 1.921394 42 FALSE 58  
## 4: 0.04139683 1.921394 Inf FALSE -50  
##   
## $DELINQ  
## variable bin count count\_distr neg pos posprob woe  
## <char> <char> <int> <num> <int> <int> <num> <num>  
## 1: DELINQ missing 453 0.09498847 398 55 0.1214128 -0.5904638  
## 2: DELINQ [-Inf,1) 3337 0.69972741 2877 460 0.1378484 -0.4446219  
## 3: DELINQ [1,2) 541 0.11344097 358 183 0.3382625 0.7176082  
## 4: DELINQ [2, Inf) 438 0.09184315 184 254 0.5799087 1.7110535  
## bin\_iv total\_iv breaks is\_special\_values points  
## <num> <num> <char> <lgcl> <num>  
## 1: 0.02745495 0.59242 missing TRUE 39  
## 2: 0.12028801 0.59242 1 FALSE 29  
## 3: 0.07063845 0.59242 2 FALSE -47  
## 4: 0.37403857 0.59242 Inf FALSE -113  
##   
## $DEROG  
## variable bin count count\_distr neg pos posprob woe  
## <char> <char> <int> <num> <int> <int> <num> <num>  
## 1: DEROG missing 552 0.1157475 482 70 0.1268116 -0.5407939  
## 2: DEROG [-Inf,1) 3622 0.7594884 3022 600 0.1656543 -0.2280895  
## 3: DEROG [1, Inf) 595 0.1247641 313 282 0.4739496 1.2843589  
## bin\_iv total\_iv breaks is\_special\_values points  
## <num> <num> <char> <lgcl> <num>  
## 1: 0.02852567 0.3404865 missing TRUE 26  
## 2: 0.03682941 0.3404865 1 FALSE 11  
## 3: 0.27513140 0.3404865 Inf FALSE -62  
##   
## $VALUE  
## variable bin count count\_distr neg pos posprob  
## <char> <char> <int> <num> <int> <int> <num>  
## 1: VALUE [-Inf,50000)%,%missing 593 0.12434473 357 236 0.39797639  
## 2: VALUE [50000,85000) 1602 0.33591948 1292 310 0.19350811  
## 3: VALUE [85000,125000) 1514 0.31746697 1300 214 0.14134742  
## 4: VALUE [125000,170000) 532 0.11155378 417 115 0.21616541  
## 5: VALUE [170000,200000) 258 0.05409939 236 22 0.08527132  
## 6: VALUE [200000, Inf) 270 0.05661564 215 55 0.20370370  
## woe bin\_iv total\_iv breaks is\_special\_values points  
## <num> <num> <num> <char> <lgcl> <num>  
## 1: 0.97475104 1.504725e-01 0.238383 50000%,%missing FALSE -56  
## 2: -0.03871937 4.977557e-04 0.238383 85000 FALSE 2  
## 3: -0.41548851 4.811012e-02 0.238383 125000 FALSE 24  
## 4: 0.10050092 1.160808e-03 0.238383 170000 FALSE -6  
## 5: -0.98413433 3.810511e-02 0.238383 200000 FALSE 57  
## 6: 0.02535018 3.666018e-05 0.238383 Inf FALSE -1  
##   
## $CLAGE  
## variable bin count count\_distr neg pos posprob woe  
## <char> <char> <int> <num> <int> <int> <num> <num>  
## 1: CLAGE missing 243 0.05095408 185 58 0.2386831 0.2287422  
## 2: CLAGE [-Inf,70) 248 0.05200252 154 94 0.3790323 0.8949972  
## 3: CLAGE [70,150) 1631 0.34200042 1205 426 0.2611895 0.3488595  
## 4: CLAGE [150,240) 1627 0.34116167 1355 272 0.1671789 -0.2170996  
## 5: CLAGE [240, Inf) 1020 0.21388132 918 102 0.1000000 -0.8085696  
## bin\_iv total\_iv breaks is\_special\_values points  
## <num> <num> <char> <lgcl> <num>  
## 1: 0.002849439 0.2239573 missing TRUE -18  
## 2: 0.052262175 0.2239573 70 FALSE -72  
## 3: 0.045974814 0.2239573 150 FALSE -28  
## 4: 0.015039914 0.2239573 240 FALSE 17  
## 5: 0.107830962 0.2239573 Inf FALSE 65  
##   
## $NINQ  
## variable bin count count\_distr neg pos posprob woe  
## <char> <char> <int> <num> <int> <int> <num> <num>  
## 1: NINQ missing 397 0.08324596 342 55 0.1385390 -0.43882253  
## 2: NINQ [-Inf,1) 2032 0.42608513 1715 317 0.1560039 -0.29961157  
## 3: NINQ [1,2) 1097 0.23002726 884 213 0.1941659 -0.03450988  
## 4: NINQ [2,3) 594 0.12455441 460 134 0.2255892 0.15526833  
## 5: NINQ [3,4) 317 0.06647096 226 91 0.2870662 0.47897953  
## 6: NINQ [4, Inf) 332 0.06961627 190 142 0.4277108 1.09745800  
## bin\_iv total\_iv breaks is\_special\_values points  
## <num> <num> <char> <lgcl> <num>  
## 1: 0.0139659890 0.1787247 missing TRUE 14  
## 2: 0.0348515798 0.1787247 1 FALSE 10  
## 3: 0.0002711095 0.1787247 2 FALSE 1  
## 4: 0.0031430673 0.1787247 3 FALSE -5  
## 5: 0.0174250033 0.1787247 4 FALSE -15  
## 6: 0.1090679574 0.1787247 Inf FALSE -35  
##   
## $LOAN  
## variable bin count count\_distr neg pos posprob woe  
## <char> <char> <int> <num> <int> <int> <num> <num>  
## 1: LOAN [-Inf,6000) 247 0.05179283 133 114 0.4615385 1.23450434  
## 2: LOAN [6000,15000) 1795 0.37638918 1430 365 0.2033426 0.02312265  
## 3: LOAN [15000,16000) 281 0.05892221 200 81 0.2882562 0.48478681  
## 4: LOAN [16000,38000) 2171 0.45523170 1846 325 0.1497006 -0.34829621  
## 5: LOAN [38000, Inf) 275 0.05766408 208 67 0.2436364 0.25580956  
## bin\_iv total\_iv breaks is\_special\_values points  
## <num> <num> <char> <lgcl> <num>  
## 1: 0.1048140856 0.1744679 6000 FALSE -45  
## 2: 0.0002026375 0.1744679 15000 FALSE -1  
## 3: 0.0158461599 0.1744679 16000 FALSE -18  
## 4: 0.0495414201 0.1744679 38000 FALSE 13  
## 5: 0.0040635588 0.1744679 Inf FALSE -9

# Áp dụng scorecard mới để tính điểm  
train\_score\_new <- scorecard\_ply(train\_dat, card\_new)  
test\_score\_new <- scorecard\_ply(test\_dat, card\_new)  
  
# BỔ SUNG: THÊM BIẾN BAD VÀO train\_score\_new và test\_score\_new THỦ CÔNG  
train\_score\_new$BAD <- train\_dat$BAD  
test\_score\_new$BAD <- test\_dat$BAD  
  
head(test\_score\_new)

## score BAD  
## <num> <fctr>  
## 1: 617 0  
## 2: 206 1  
## 3: 370 0  
## 4: 630 0  
## 5: 340 1  
## 6: 465 0

# Vẽ biểu đồ phân phối điểm mới  
ggplot(test\_score\_new, aes(x = score, fill = BAD)) +  
 geom\_density(alpha = 0.5) +  
 labs(title = "New Score Distribution by BAD/GOOD Status", x = "Score", fill = "BAD") +  
 theme\_minimal()

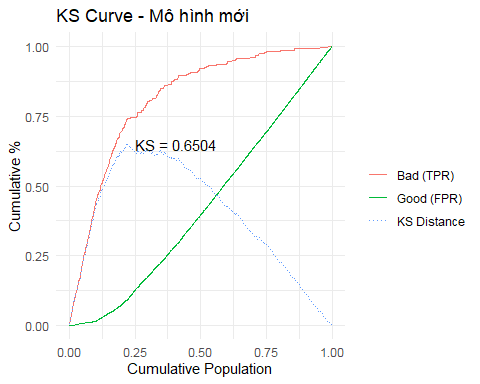


# ====================================================================  
# Bổ sung: Cách tùy chỉnh Base Point, PDO, Odds THỦ CÔNG (Nếu bạn muốn)  
# (Không thay đổi so với lần trước)  
# ====================================================================

# --- KS cho mô hình mới ---  
actual\_new <- as.numeric(as.character(test\_woe$BAD))  
predicted\_new <- test\_pred\_prob\_new  
  
ks\_df\_new <- data.frame(actual = actual\_new, predicted = predicted\_new)  
ks\_df\_new <- ks\_df\_new[order(-ks\_df\_new$predicted), ]  
  
ks\_df\_new$bad <- ifelse(ks\_df\_new$actual == 1, 1, 0)  
ks\_df\_new$good <- ifelse(ks\_df\_new$actual == 0, 1, 0)  
ks\_df\_new$bad\_cum\_pct <- cumsum(ks\_df\_new$bad) / sum(ks\_df\_new$bad)  
ks\_df\_new$good\_cum\_pct <- cumsum(ks\_df\_new$good) / sum(ks\_df\_new$good)  
ks\_df\_new$ks\_stat <- abs(ks\_df\_new$bad\_cum\_pct - ks\_df\_new$good\_cum\_pct)  
  
ks\_val\_new <- max(ks\_df\_new$ks\_stat)  
cat("KS Statistic (mô hình mới):", round(ks\_val\_new, 4), "\n")

## KS Statistic (mô hình mới): 0.6504

# Vẽ KS Curve cho mô hình mới  
ggplot(ks\_df\_new, aes(x = seq\_along(actual) / length(actual))) +  
 geom\_line(aes(y = bad\_cum\_pct, color = "Bad (TPR)")) +  
 geom\_line(aes(y = good\_cum\_pct, color = "Good (FPR)")) +  
 geom\_line(aes(y = ks\_stat, color = "KS Distance"), linetype = "dotted") +  
 labs(  
 title = "KS Curve - Mô hình mới",  
 x = "Cumulative Population",  
 y = "Cumulative %",  
 color = ""  
 ) +  
 annotate("text", x = which.max(ks\_df\_new$ks\_stat) / nrow(ks\_df\_new),  
 y = ks\_val\_new,  
 label = paste0("KS = ", round(ks\_val\_new, 4)),  
 hjust = -0.1, vjust = 0.5, color = "black") +  
 theme\_minimal()



# --- PSI cho mô hình mới ---  
  
cut\_bins\_new <- woebin\_ply(data.frame(score = c(train\_score\_new$score, test\_score\_new$score)),   
 woebin(data.frame(score = c(train\_score\_new$score, test\_score\_new$score),  
 BAD = c(train\_score\_new$BAD, test\_score\_new$BAD)),  
 y = "BAD", x = "score", bin\_num\_limit = 10))

## ℹ Converting into woe values ...  
## ℹ Creating woe binning ...

## ✔ Binning on 5960 rows and 2 columns in 00:00:00  
## ✔ Woe transformating on 5960 rows and 1 columns in 00:00:00

train\_score\_new$bin <- cut\_bins\_new$score\_woe[1:nrow(train\_score\_new)]  
test\_score\_new$bin <- cut\_bins\_new$score\_woe[(nrow(train\_score\_new)+1):(nrow(cut\_bins\_new))]  
  
psi\_table\_new <- data.frame(table(train\_score\_new$bin) / nrow(train\_score\_new))  
names(psi\_table\_new) <- c("bin", "train\_pct")  
psi\_table\_new$test\_pct <- table(test\_score\_new$bin) / nrow(test\_score\_new)  
psi\_table\_new$psi <- (psi\_table\_new$train\_pct - psi\_table\_new$test\_pct) \*   
 log(psi\_table\_new$train\_pct / psi\_table\_new$test\_pct)  
  
psi\_val\_new <- sum(psi\_table\_new$psi, na.rm = TRUE)  
cat("PSI (mô hình mới):", round(psi\_val\_new, 4), "\n")

## PSI (mô hình mới): 0.0011

print(psi\_table\_new)

## bin train\_pct test\_pct psi  
## 1 -2.54460800921573 0.38309918 0.39126784 1.723459e-04  
## 2 -0.987977425992535 0.30928916 0.31402183 7.186977e-05  
## 3 0.676940685312424 0.15726567 0.15365239 8.398604e-05  
## 4 2.03579832787851 0.08597190 0.08312343 9.597644e-05  
## 5 3.87147225878571 0.06437408 0.05793451 6.787175e-04

# --- Gini cho mô hình mới ---  
  
gini\_df\_new <- data.frame(  
 actual = as.numeric(as.character(test\_woe$BAD)),  
 predicted = test\_pred\_prob\_new  
)  
gini\_df\_new <- gini\_df\_new[order(-gini\_df\_new$predicted), ]  
gini\_df\_new$cum\_pop <- seq\_along(gini\_df\_new$actual) / nrow(gini\_df\_new)  
gini\_df\_new$cum\_bad <- cumsum(gini\_df\_new$actual) / sum(gini\_df\_new$actual)  
  
gini\_val\_new <- 2 \* auc(roc(response = test\_woe$BAD, predictor = test\_pred\_prob\_new)) - 1

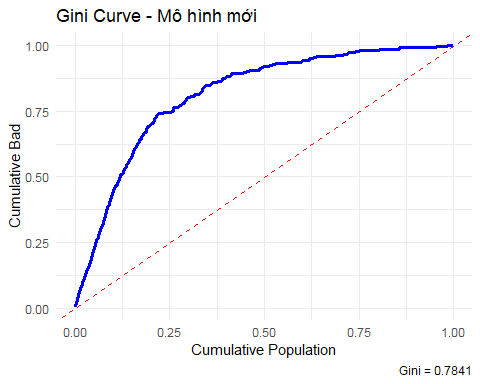
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

cat("Gini (mô hình mới):", round(gini\_val\_new, 4), "\n")

## Gini (mô hình mới): 0.7841

ggplot(gini\_df\_new, aes(x = cum\_pop, y = cum\_bad)) +  
 geom\_line(color = "blue", size = 1.2) +  
 geom\_abline(slope = 1, intercept = 0, color = "red", linetype = "dashed") +  
 labs(  
 title = "Gini Curve - Mô hình mới",  
 x = "Cumulative Population",  
 y = "Cumulative Bad",  
 caption = paste("Gini =", round(gini\_val\_new, 4))  
 ) +  
 theme\_minimal()



# Dự đoán xác suất từ mô hình mới  
test\_prob\_bad\_new <- predict(logistic\_model\_new, newdata = test\_woe, type = "response")  
test\_prob\_good\_new <- 1 - test\_prob\_bad\_new  
  
# Tính odds  
odds\_good\_bad\_new <- test\_prob\_good\_new / test\_prob\_bad\_new  
odds\_bad\_good\_new <- test\_prob\_bad\_new / test\_prob\_good\_new  
  
# Tạo bảng phân tích  
analysis\_df\_new <- data.frame(  
 score = test\_score\_new$score,  
 prob\_bad = test\_prob\_bad\_new,  
 prob\_good = test\_prob\_good\_new,  
 odds\_good\_bad = odds\_good\_bad\_new,  
 odds\_bad\_good = odds\_bad\_good\_new  
)  
  
# Sắp xếp theo điểm  
analysis\_df\_sorted\_new <- analysis\_df\_new %>% arrange(score)  
  
cat("\n--- Phân tích Odds theo Score (mô hình mới) ---\n")

##   
## --- Phân tích Odds theo Score (mô hình mới) ---

print(head(analysis\_df\_sorted\_new))

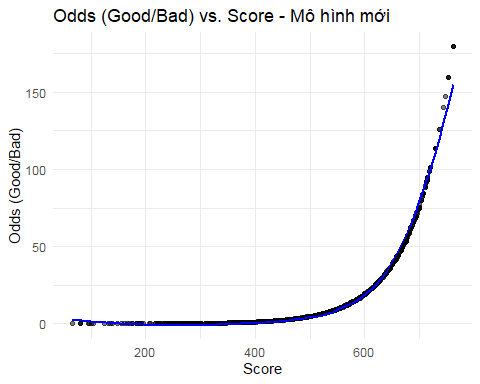
## score prob\_bad prob\_good odds\_good\_bad odds\_bad\_good  
## 1183 66 0.9884818 0.01151823 0.01165245 85.81887  
## 10 80 0.9861179 0.01388213 0.01407756 71.03505  
## 1137 80 0.9861579 0.01384207 0.01403636 71.24353  
## 1141 80 0.9861579 0.01384207 0.01403636 71.24353  
## 40 95 0.9828058 0.01719419 0.01749500 57.15918  
## 64 96 0.9826902 0.01730979 0.01761469 56.77079

print(tail(analysis\_df\_sorted\_new))

## score prob\_bad prob\_good odds\_good\_bad odds\_bad\_good  
## 707 753 0.006220155 0.9937798 159.7677 0.006259088  
## 765 753 0.006220155 0.9937798 159.7677 0.006259088  
## 835 753 0.006220155 0.9937798 159.7677 0.006259088  
## 462 762 0.005537646 0.9944624 179.5821 0.005568483  
## 479 762 0.005537646 0.9944624 179.5821 0.005568483  
## 512 762 0.005537646 0.9944624 179.5821 0.005568483

# Vẽ Odds vs Score  
ggplot(analysis\_df\_new, aes(x = score, y = odds\_good\_bad)) +  
 geom\_point(alpha = 0.5) +  
 geom\_smooth(method = "loess", se = FALSE, color = "blue") +  
 labs(title = "Odds (Good/Bad) vs. Score - Mô hình mới", x = "Score", y = "Odds (Good/Bad)") +  
 theme\_minimal()

## `geom\_smooth()` using formula = 'y ~ x'



ggplot(analysis\_df\_new, aes(x = score, y = odds\_bad\_good)) +  
 geom\_point(alpha = 0.5) +  
 geom\_smooth(method = "loess", se = FALSE, color = "red") +  
 labs(title = "Odds (Bad/Good) vs. Score - Mô hình mới", x = "Score", y = "Odds (Bad/Good)") +  
 theme\_minimal()

## `geom\_smooth()` using formula = 'y ~ x'

