course.R

Denis

2024-05-25

# Установка необходимых библиотек ------------------------------------------  
  
if (!require("caret")) install.packages("caret", dependencies=TRUE)

## Загрузка требуемого пакета: caret

## Загрузка требуемого пакета: ggplot2

## Загрузка требуемого пакета: lattice

if (!require("cluster")) install.packages("cluster", dependencies=TRUE)

## Загрузка требуемого пакета: cluster

if (!require("clusterSim")) install.packages("clusterSim", dependencies=TRUE)

## Загрузка требуемого пакета: clusterSim

## Загрузка требуемого пакета: MASS

if (!require("corrplot")) install.packages("corrplot", dependencies=TRUE)

## Загрузка требуемого пакета: corrplot

## corrplot 0.92 loaded

if (!require("data.table")) install.packages("data.table", dependencies=TRUE)

## Загрузка требуемого пакета: data.table

if (!require("dbscan")) install.packages("dbscan", dependencies=TRUE)

## Загрузка требуемого пакета: dbscan

##   
## Присоединяю пакет: 'dbscan'

## Следующий объект скрыт от 'package:stats':  
##   
## as.dendrogram

if (!require("dplyr")) install.packages("dplyr", dependencies=TRUE)

## Загрузка требуемого пакета: dplyr

##   
## Присоединяю пакет: 'dplyr'

## Следующие объекты скрыты от 'package:data.table':  
##   
## between, first, last

## Следующий объект скрыт от 'package:MASS':  
##   
## select

## Следующие объекты скрыты от 'package:stats':  
##   
## filter, lag

## Следующие объекты скрыты от 'package:base':  
##   
## intersect, setdiff, setequal, union

if (!require("factoextra")) install.packages("factoextra", dependencies=TRUE)

## Загрузка требуемого пакета: factoextra

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

if (!require("fpc")) install.packages("fpc", dependencies=TRUE)

## Загрузка требуемого пакета: fpc

##   
## Присоединяю пакет: 'fpc'

## Следующий объект скрыт от 'package:dbscan':  
##   
## dbscan

if (!require("ggplot2")) install.packages("ggplot2", dependencies=TRUE)  
if (!require("ggpubr")) install.packages("ggpubr", dependencies=TRUE)

## Загрузка требуемого пакета: ggpubr

if (!require("mclust")) install.packages("mclust", dependencies=TRUE)

## Загрузка требуемого пакета: mclust

## Package 'mclust' version 6.1.1  
## Type 'citation("mclust")' for citing this R package in publications.

if (!require("plotly")) install.packages("plotly", dependencies=TRUE)

## Загрузка требуемого пакета: plotly

##   
## Присоединяю пакет: 'plotly'

## Следующий объект скрыт от 'package:MASS':  
##   
## select

## Следующий объект скрыт от 'package:ggplot2':  
##   
## last\_plot

## Следующий объект скрыт от 'package:stats':  
##   
## filter

## Следующий объект скрыт от 'package:graphics':  
##   
## layout

if (!require("Rtsne")) install.packages("Rtsne", dependencies=TRUE)

## Загрузка требуемого пакета: Rtsne

if (!require("tidyr")) install.packages("tidyr", dependencies=TRUE)

## Загрузка требуемого пакета: tidyr

if (!require("umap")) install.packages("umap", dependencies=TRUE)

## Загрузка требуемого пакета: umap

library(caret)  
library(cluster)  
library(clusterSim)  
library(corrplot)  
library(data.table)  
library(dbscan)  
library(dplyr)  
library(factoextra)  
library(fpc)  
library(ggplot2)  
library(ggpubr)  
library(mclust)  
library(plotly)  
library(Rtsne)  
library(tidyr)  
library(umap)  
  
# Загрузка и предобработка данных -----------------------------------------  
  
data <- fread("bank\_transactions.csv")  
head(data)

## TransactionID CustomerID CustomerDOB CustGender CustLocation  
## <char> <char> <char> <char> <char>  
## 1: T1 C5841053 10/1/94 F JAMSHEDPUR  
## 2: T2 C2142763 4/4/57 M JHAJJAR  
## 3: T3 C4417068 26/11/96 F MUMBAI  
## 4: T4 C5342380 14/9/73 F MUMBAI  
## 5: T5 C9031234 24/3/88 F NAVI MUMBAI  
## 6: T6 C1536588 8/10/72 F ITANAGAR  
## CustAccountBalance TransactionDate TransactionTime TransactionAmount (INR)  
## <num> <char> <int> <num>  
## 1: 17819.05 2/8/16 143207 25.0  
## 2: 2270.69 2/8/16 141858 27999.0  
## 3: 17874.44 2/8/16 142712 459.0  
## 4: 866503.21 2/8/16 142714 2060.0  
## 5: 6714.43 2/8/16 181156 1762.5  
## 6: 53609.20 2/8/16 173940 676.0

# Очистка пустых значений  
colSums(is.na(data))

## TransactionID CustomerID CustomerDOB   
## 0 0 0   
## CustGender CustLocation CustAccountBalance   
## 0 0 2369   
## TransactionDate TransactionTime TransactionAmount (INR)   
## 0 0 0

data <- na.omit(data)  
colSums(is.na(data))

## TransactionID CustomerID CustomerDOB   
## 0 0 0   
## CustGender CustLocation CustAccountBalance   
## 0 0 0   
## TransactionDate TransactionTime TransactionAmount (INR)   
## 0 0 0

summary(data)

## TransactionID CustomerID CustomerDOB CustGender   
## Length:1046198 Length:1046198 Length:1046198 Length:1046198   
## Class :character Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character Mode :character   
##   
##   
##   
## CustLocation CustAccountBalance TransactionDate TransactionTime   
## Length:1046198 Min. : 0 Length:1046198 Min. : 0   
## Class :character 1st Qu.: 4722 Class :character 1st Qu.:124031   
## Mode :character Median : 16792 Mode :character Median :164231   
## Mean : 115404 Mean :157094   
## 3rd Qu.: 57657 3rd Qu.:200013   
## Max. :115035495 Max. :235959   
## TransactionAmount (INR)  
## Min. : 0   
## 1st Qu.: 161   
## Median : 459   
## Mean : 1573   
## 3rd Qu.: 1200   
## Max. :1560035

head(data)

## TransactionID CustomerID CustomerDOB CustGender CustLocation  
## <char> <char> <char> <char> <char>  
## 1: T1 C5841053 10/1/94 F JAMSHEDPUR  
## 2: T2 C2142763 4/4/57 M JHAJJAR  
## 3: T3 C4417068 26/11/96 F MUMBAI  
## 4: T4 C5342380 14/9/73 F MUMBAI  
## 5: T5 C9031234 24/3/88 F NAVI MUMBAI  
## 6: T6 C1536588 8/10/72 F ITANAGAR  
## CustAccountBalance TransactionDate TransactionTime TransactionAmount (INR)  
## <num> <char> <int> <num>  
## 1: 17819.05 2/8/16 143207 25.0  
## 2: 2270.69 2/8/16 141858 27999.0  
## 3: 17874.44 2/8/16 142712 459.0  
## 4: 866503.21 2/8/16 142714 2060.0  
## 5: 6714.43 2/8/16 181156 1762.5  
## 6: 53609.20 2/8/16 173940 676.0

# Просмотр типов данных  
str(data)

## Classes 'data.table' and 'data.frame': 1046198 obs. of 9 variables:  
## $ TransactionID : chr "T1" "T2" "T3" "T4" ...  
## $ CustomerID : chr "C5841053" "C2142763" "C4417068" "C5342380" ...  
## $ CustomerDOB : chr "10/1/94" "4/4/57" "26/11/96" "14/9/73" ...  
## $ CustGender : chr "F" "M" "F" "F" ...  
## $ CustLocation : chr "JAMSHEDPUR" "JHAJJAR" "MUMBAI" "MUMBAI" ...  
## $ CustAccountBalance : num 17819 2271 17874 866503 6714 ...  
## $ TransactionDate : chr "2/8/16" "2/8/16" "2/8/16" "2/8/16" ...  
## $ TransactionTime : int 143207 141858 142712 142714 181156 173940 173806 170537 192825 192446 ...  
## $ TransactionAmount (INR): num 25 27999 459 2060 1762 ...  
## - attr(\*, ".internal.selfref")=<externalptr>

# Преобразуем столбец CustGender в строковый формат  
data$CustGender <- as.character(data$CustGender)  
  
# Удаляем строки, в которых значение в столбце CustGender не равно F или M  
data <- data[data$CustGender %in% c("F", "M"), ]  
  
# Заменяем F на 0, а M на 1 в столбце CustGender  
data$CustGender <- ifelse(data$CustGender == "F", 0, 1)  
  
# Преобразуем столбец CustGender в числовой формат  
data$CustGender <- as.numeric(data$CustGender)  
  
# Переименовываем столбец TransactionAmount (INR) в TransactionAmount  
colnames(data)[colnames(data) == "TransactionAmount (INR)"] <- "TransactionAmount"  
  
# Удалим ненужные столбцы  
data <- subset(data, select = -c(CustomerDOB, CustLocation, TransactionDate))  
head(data)

## TransactionID CustomerID CustGender CustAccountBalance TransactionTime  
## <char> <char> <num> <num> <int>  
## 1: T1 C5841053 0 17819.05 143207  
## 2: T2 C2142763 1 2270.69 141858  
## 3: T3 C4417068 0 17874.44 142712  
## 4: T4 C5342380 0 866503.21 142714  
## 5: T5 C9031234 0 6714.43 181156  
## 6: T6 C1536588 0 53609.20 173940  
## TransactionAmount  
## <num>  
## 1: 25.0  
## 2: 27999.0  
## 3: 459.0  
## 4: 2060.0  
## 5: 1762.5  
## 6: 676.0

# Убираем первую букву в ID и переводим столбец в числовой тип  
data$TransactionID <- as.numeric(gsub("^T", "", data$TransactionID))  
data$CustomerID <- as.numeric(gsub("^C", "", data$CustomerID))  
  
# Убираем операции на 0 рупий  
data <- subset(data, data$TransactionAmount != 0.0)  
  
# Считаем статистические данные  
data\_stats <- data %>%  
 group\_by(CustomerID) %>%  
 summarise(  
 totalTransactionAmount = sum(TransactionAmount),  
 avgTransactionAmount = mean(TransactionAmount),  
 numOfTransactions = n(),  
 stdTransactionAmount = sd(TransactionAmount),  
 quantile25 = quantile(TransactionAmount, 0.25),  
 dataMediana = median(TransactionAmount),  
 quantile75 = quantile(TransactionAmount, 0.75),  
 minTransactionAmount = min(TransactionAmount),  
 maxTransactionAmount = max(TransactionAmount),  
 dispersionTransactionAmount = var(TransactionAmount)  
 ) %>%  
 mutate(stdTransactionAmount = ifelse(is.na(stdTransactionAmount), 0, stdTransactionAmount),  
 dispersionTransactionAmount = ifelse(is.na(dispersionTransactionAmount), 0, dispersionTransactionAmount))  
  
# Проверка результата  
head(data)

## TransactionID CustomerID CustGender CustAccountBalance TransactionTime  
## <num> <num> <num> <num> <int>  
## 1: 1 5841053 0 17819.05 143207  
## 2: 2 2142763 1 2270.69 141858  
## 3: 3 4417068 0 17874.44 142712  
## 4: 4 5342380 0 866503.21 142714  
## 5: 5 9031234 0 6714.43 181156  
## 6: 6 1536588 0 53609.20 173940  
## TransactionAmount  
## <num>  
## 1: 25.0  
## 2: 27999.0  
## 3: 459.0  
## 4: 2060.0  
## 5: 1762.5  
## 6: 676.0

head(data\_stats)

## # A tibble: 6 × 11  
## CustomerID totalTransactionAmount avgTransactionAmount numOfTransactions  
## <dbl> <dbl> <dbl> <int>  
## 1 1010011 5106 2553 2  
## 2 1010012 1499 1499 1  
## 3 1010014 1455 728. 2  
## 4 1010018 30 30 1  
## 5 1010024 5000 5000 1  
## 6 1010028 557 557 1  
## # ℹ 7 more variables: stdTransactionAmount <dbl>, quantile25 <dbl>,  
## # dataMediana <dbl>, quantile75 <dbl>, minTransactionAmount <dbl>,  
## # maxTransactionAmount <dbl>, dispersionTransactionAmount <dbl>

# Выбираем клиентов с более чем 1 транзакцией  
data\_stats <- subset(data\_stats, data\_stats$numOfTransactions > 1)  
head(data\_stats)

## # A tibble: 6 × 11  
## CustomerID totalTransactionAmount avgTransactionAmount numOfTransactions  
## <dbl> <dbl> <dbl> <int>  
## 1 1010011 5106 2553 2  
## 2 1010014 1455 728. 2  
## 3 1010031 1864 932 2  
## 4 1010035 750 375 2  
## 5 1010041 16917 5639 3  
## 6 1010046 157520. 78760. 2  
## # ℹ 7 more variables: stdTransactionAmount <dbl>, quantile25 <dbl>,  
## # dataMediana <dbl>, quantile75 <dbl>, minTransactionAmount <dbl>,  
## # maxTransactionAmount <dbl>, dispersionTransactionAmount <dbl>

# Корреляционная матрица для данных data\_stats  
correlation\_matrix <- cor(data\_stats[, -1])   
cat("Корреляционная матрица:\n")

## Корреляционная матрица:

print(correlation\_matrix)

## totalTransactionAmount avgTransactionAmount  
## totalTransactionAmount 1.00000000 0.9835482429  
## avgTransactionAmount 0.98354824 1.0000000000  
## numOfTransactions 0.07079428 0.0005387445  
## stdTransactionAmount 0.96960045 0.9729006663  
## quantile25 0.87758010 0.9385183130  
## dataMediana 0.89667800 0.9576611759  
## quantile75 0.97854240 0.9967284352  
## minTransactionAmount 0.29379973 0.3165740388  
## maxTransactionAmount 0.99229548 0.9802182346  
## dispersionTransactionAmount 0.61805107 0.6183619929  
## numOfTransactions stdTransactionAmount quantile25  
## totalTransactionAmount 0.0707942773 0.96960045 0.87758010  
## avgTransactionAmount 0.0005387445 0.97290067 0.93851831  
## numOfTransactions 1.0000000000 0.01878916 -0.07671854  
## stdTransactionAmount 0.0187891620 1.00000000 0.84719232  
## quantile25 -0.0767185397 0.84719232 1.00000000  
## dataMediana -0.0657789191 0.90632565 0.98061532  
## quantile75 0.0019072347 0.98315599 0.92299313  
## minTransactionAmount -0.0671076768 0.10136017 0.50969924  
## maxTransactionAmount 0.0452764533 0.98999551 0.85395020  
## dispersionTransactionAmount 0.0023037597 0.64847533 0.53029523  
## dataMediana quantile75 minTransactionAmount  
## totalTransactionAmount 0.89667800 0.978542404 0.29379973  
## avgTransactionAmount 0.95766118 0.996728435 0.31657404  
## numOfTransactions -0.06577892 0.001907235 -0.06710768  
## stdTransactionAmount 0.90632565 0.983155988 0.10136017  
## quantile25 0.98061532 0.922993126 0.50969924  
## dataMediana 1.00000000 0.955788292 0.33213966  
## quantile75 0.95578829 1.000000000 0.24995344  
## minTransactionAmount 0.33213966 0.249953444 1.00000000  
## maxTransactionAmount 0.89167713 0.981396009 0.19984089  
## dispersionTransactionAmount 0.57659907 0.624544610 0.02117053  
## maxTransactionAmount dispersionTransactionAmount  
## totalTransactionAmount 0.99229548 0.61805107  
## avgTransactionAmount 0.98021823 0.61836199  
## numOfTransactions 0.04527645 0.00230376  
## stdTransactionAmount 0.98999551 0.64847533  
## quantile25 0.85395020 0.53029523  
## dataMediana 0.89167713 0.57659907  
## quantile75 0.98139601 0.62454461  
## minTransactionAmount 0.19984089 0.02117053  
## maxTransactionAmount 1.00000000 0.63834145  
## dispersionTransactionAmount 0.63834145 1.00000000

# Визуализация корреляционной матрицы  
corrplot(correlation\_matrix, method = "circle", type = "upper", tl.cex = 0.8)  
  
# Анализ данных -----------------------------------------------------------  
  
# Количество уникальных клиентов  
num\_unique\_customers <- length(unique(data\_stats$CustomerID))  
cat("Количество уникальных клиентов:", num\_unique\_customers, "\n")

## Количество уникальных клиентов: 142614

# Описательная статистика суммы транзакций  
cat("Описательная статистика суммы транзакций:\n")

## Описательная статистика суммы транзакций:

summary(data$TransactionAmount)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0 163 460 1574 1200 1560035

# Описательная статистика для avgTransactionAmount  
cat("Описательная статистика для среднего размера транзакций:\n")

## Описательная статистика для среднего размера транзакций:

summary(data\_stats$avgTransactionAmount)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.0 313.4 665.3 1570.2 1414.5 362236.0

# Описательная статистика для numOfTransactions  
cat("Описательная статистика для количества транзакций:\n")

## Описательная статистика для количества транзакций:

summary(data\_stats$numOfTransactions)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 2.000 2.000 2.000 2.143 2.000 6.000

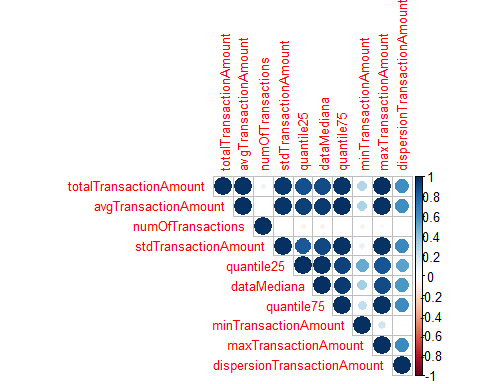
# Корреляционная матрица для данных data\_stats  
correlation\_matrix <- cor(data\_stats[, -1]) # Убираем CustomerID перед расчетом корреляции  
cat("Корреляционная матрица:\n")

## Корреляционная матрица:

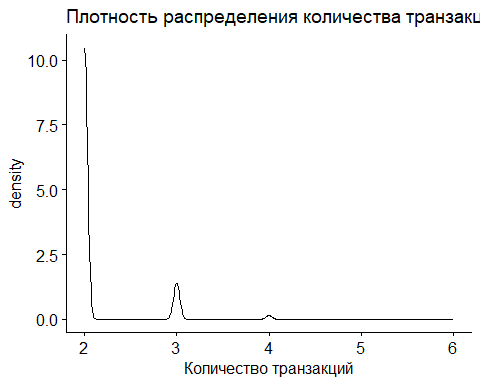
print(correlation\_matrix)

## totalTransactionAmount avgTransactionAmount  
## totalTransactionAmount 1.00000000 0.9835482429  
## avgTransactionAmount 0.98354824 1.0000000000  
## numOfTransactions 0.07079428 0.0005387445  
## stdTransactionAmount 0.96960045 0.9729006663  
## quantile25 0.87758010 0.9385183130  
## dataMediana 0.89667800 0.9576611759  
## quantile75 0.97854240 0.9967284352  
## minTransactionAmount 0.29379973 0.3165740388  
## maxTransactionAmount 0.99229548 0.9802182346  
## dispersionTransactionAmount 0.61805107 0.6183619929  
## numOfTransactions stdTransactionAmount quantile25  
## totalTransactionAmount 0.0707942773 0.96960045 0.87758010  
## avgTransactionAmount 0.0005387445 0.97290067 0.93851831  
## numOfTransactions 1.0000000000 0.01878916 -0.07671854  
## stdTransactionAmount 0.0187891620 1.00000000 0.84719232  
## quantile25 -0.0767185397 0.84719232 1.00000000  
## dataMediana -0.0657789191 0.90632565 0.98061532  
## quantile75 0.0019072347 0.98315599 0.92299313  
## minTransactionAmount -0.0671076768 0.10136017 0.50969924  
## maxTransactionAmount 0.0452764533 0.98999551 0.85395020  
## dispersionTransactionAmount 0.0023037597 0.64847533 0.53029523  
## dataMediana quantile75 minTransactionAmount  
## totalTransactionAmount 0.89667800 0.978542404 0.29379973  
## avgTransactionAmount 0.95766118 0.996728435 0.31657404  
## numOfTransactions -0.06577892 0.001907235 -0.06710768  
## stdTransactionAmount 0.90632565 0.983155988 0.10136017  
## quantile25 0.98061532 0.922993126 0.50969924  
## dataMediana 1.00000000 0.955788292 0.33213966  
## quantile75 0.95578829 1.000000000 0.24995344  
## minTransactionAmount 0.33213966 0.249953444 1.00000000  
## maxTransactionAmount 0.89167713 0.981396009 0.19984089  
## dispersionTransactionAmount 0.57659907 0.624544610 0.02117053  
## maxTransactionAmount dispersionTransactionAmount  
## totalTransactionAmount 0.99229548 0.61805107  
## avgTransactionAmount 0.98021823 0.61836199  
## numOfTransactions 0.04527645 0.00230376  
## stdTransactionAmount 0.98999551 0.64847533  
## quantile25 0.85395020 0.53029523  
## dataMediana 0.89167713 0.57659907  
## quantile75 0.98139601 0.62454461  
## minTransactionAmount 0.19984089 0.02117053  
## maxTransactionAmount 1.00000000 0.63834145  
## dispersionTransactionAmount 0.63834145 1.00000000

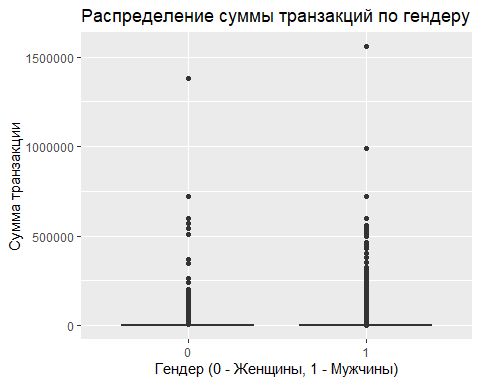
# Визуализация корреляционной матрицы  
corrplot(correlation\_matrix, method = "circle", type = "upper", tl.cex = 0.8)



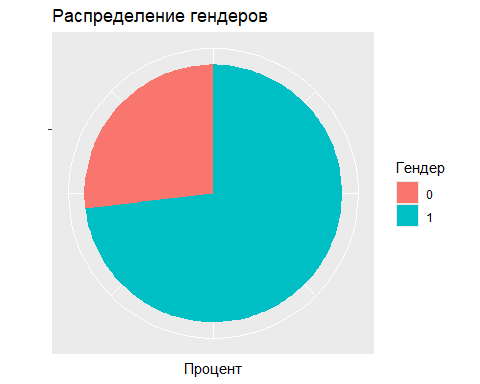
# Проверка распределения количества транзакций  
ggdensity(data\_stats$numOfTransactions, main = "Плотность распределения количества транзакций", xlab = "Количество транзакций")



# Графики анализа данных data ---------------------------------------------  
  
# 1. Распределение суммы транзакций по гендеру  
ggplot(data, aes(x = as.factor(CustGender), y = TransactionAmount)) +  
 geom\_boxplot() +  
 labs(title = "Распределение суммы транзакций по гендеру", x = "Гендер (0 - Женщины, 1 - Мужчины)", y = "Сумма транзакции")



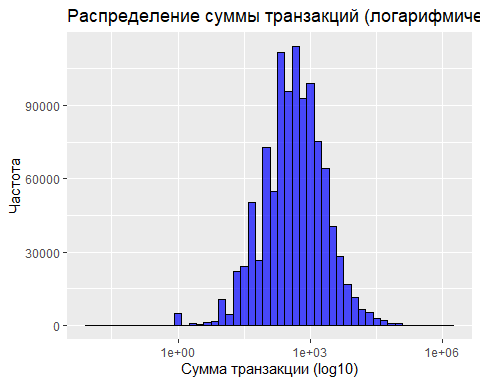
# 2. Круговая диаграмма распределения гендеров  
gender\_distribution <- data %>%  
 group\_by(CustGender) %>%  
 summarise(count = n()) %>%  
 mutate(percentage = round((count / sum(count)) \* 100, 1))  
  
ggplot(gender\_distribution, aes(x = "", y = percentage, fill = as.factor(CustGender))) +  
 geom\_bar(width = 1, stat = "identity") +  
 coord\_polar("y") +  
 labs(title = "Распределение гендеров", x = "", y = "Процент", fill = "Гендер") +  
 theme(axis.text.x = element\_blank())



print(gender\_distribution)

## # A tibble: 2 × 3  
## CustGender count percentage  
## <dbl> <int> <dbl>  
## 1 0 281120 26.9  
## 2 1 763147 73.1

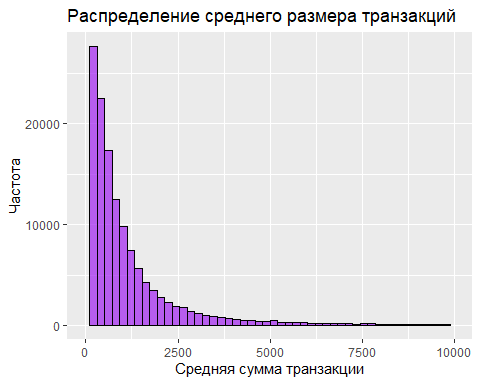
# 3. Распределение суммы транзакций с логарифмической шкалой  
ggplot(data, aes(x = TransactionAmount)) +  
 geom\_histogram(bins = 50, fill = "blue", color = "black", alpha = 0.7) +  
 scale\_x\_log10() +  
 labs(title = "Распределение суммы транзакций (логарифмическая шкала)", x = "Сумма транзакции (log10)", y = "Частота")



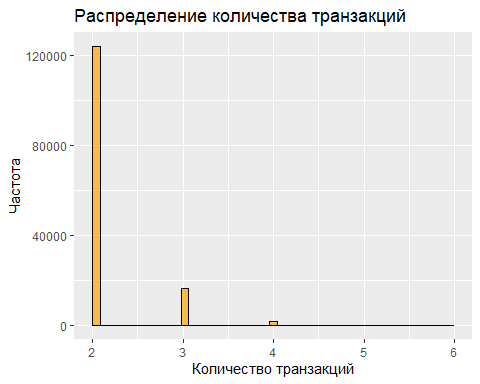
# Графики анализа данных data\_stats ---------------------------------------  
  
# 1. Распределение среднего размера транзакций (ограниченный диапазон)  
ggplot(data\_stats, aes(x = avgTransactionAmount)) +  
 geom\_histogram(bins = 50, fill = "purple", color = "black", alpha = 0.7) +  
 scale\_x\_continuous(limits = c(0, 10000)) + # Ограничиваем ось x до 10000  
 labs(title = "Распределение среднего размера транзакций", x = "Средняя сумма транзакции", y = "Частота")

## Warning: Removed 3131 rows containing non-finite outside the scale range  
## (`stat\_bin()`).

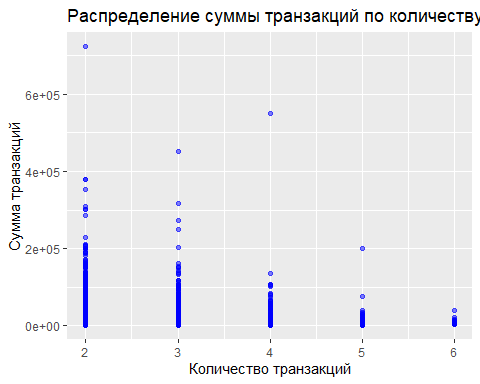
## Warning: Removed 2 rows containing missing values or values outside the scale range  
## (`geom\_bar()`).



# 2. Распределение количества транзакций  
ggplot(data\_stats, aes(x = numOfTransactions)) +  
 geom\_histogram(bins = 50, fill = "orange", color = "black", alpha = 0.7) +  
 labs(title = "Распределение количества транзакций", x = "Количество транзакций", y = "Частота")

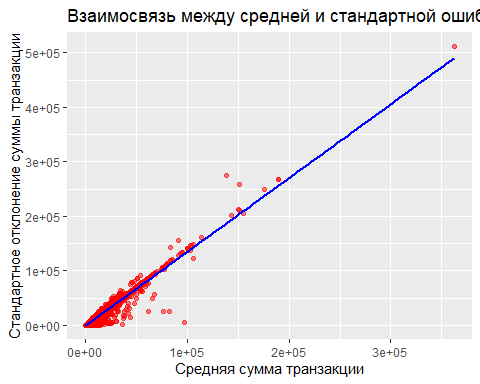


# 3. Распределение суммы транзакций по количеству транзакций  
ggplot(data\_stats, aes(x = numOfTransactions, y = totalTransactionAmount)) +  
 geom\_point(color = "blue", alpha = 0.5) +  
 labs(title = "Распределение суммы транзакций по количеству транзакций", x = "Количество транзакций", y = "Сумма транзакций")



# 4. Взаимосвязь между средней и стандартной ошибкой суммы транзакций (убираем нулевые std)  
filtered\_data <- data\_stats %>% filter(stdTransactionAmount != 0)  
ggplot(filtered\_data, aes(x = avgTransactionAmount, y = stdTransactionAmount)) +  
 geom\_point(color = "red", alpha = 0.5) +  
 geom\_smooth(method = "lm", se = FALSE, color = "blue") + # Линейная регрессия  
 labs(title = "Взаимосвязь между средней и стандартной ошибкой суммы транзакций", x = "Средняя сумма транзакции", y = "Стандартное отклонение суммы транзакции")

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



# Рассчитаем коэффициент линейной регрессии  
model <- lm(stdTransactionAmount ~ avgTransactionAmount, data = filtered\_data)  
k <- coef(model)[2]  
cat("Коэффициент k для линейной зависимости:", k, "\n")

## Коэффициент k для линейной зависимости: 1.354534

# Нормализация данных -----------------------------------------------------  
  
# Данные для визуализации  
data\_stats <- subset(data\_stats, select = -c(CustomerID, numOfTransactions))  
  
normalized\_data <- scale(data\_stats[, c("totalTransactionAmount",  
 "avgTransactionAmount", "quantile25", "dataMediana",  
 "quantile75", "minTransactionAmount", "maxTransactionAmount")])  
  
normalized\_data\_unique <- unique(normalized\_data)  
  
set.seed(2007)  
sample\_indices <- sample(1:nrow(normalized\_data), 5000)  
sample\_normalized\_data <- normalized\_data[sample\_indices, ]  
  
kNNdistplot(sample\_normalized\_data, k = 3) # k = minPts - 1  
abline(h = 1, col = "red") # Примерная линия для eps

