shah\_vaishakhi

2024-02-23

## Answer the following questions using an RMarkdown document.

## Download the BankChurners.csv dataset in CSV format from:

# https://www.kaggle.com/sakshigoyal7/credit-card-customers

## Organize your code using the RMarkdown headers for each question and add descriptions to explain your code for each question.

## Use the library(formattable) and the currency() function to present the currency values with dollar signs, thousand separators,and rounded to two decimal places.

## All questions should be answered as a nicely formatted sentence using print statements, rather than a single numeric value.

# Loading Required Libraries

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.3 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(formattable)

# 1. Import the downloaded CSV file into a tibble data frame. Print the number of rows and columns using the appropriate functions.

## Loading and Organizing Data  
  
library(readr)  
bank\_data <- read\_csv("BankChurners.csv")

## Rows: 10127 Columns: 23  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (6): Attrition\_Flag, Gender, Education\_Level, Marital\_Status, Income\_Ca...  
## dbl (17): CLIENTNUM, Customer\_Age, Dependent\_count, Months\_on\_book, Total\_Re...  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

# View(bank\_data)

# Load the CSV file into a tibble data frame  
  
bank\_data <- tibble::as\_tibble(bank\_data)   
  
bank\_data

## # A tibble: 10,127 × 23  
## CLIENTNUM Attrition\_Flag Customer\_Age Gender Dependent\_count Education\_Level  
## <dbl> <chr> <dbl> <chr> <dbl> <chr>   
## 1 768805383 Existing Custo… 45 M 3 High School   
## 2 818770008 Existing Custo… 49 F 5 Graduate   
## 3 713982108 Existing Custo… 51 M 3 Graduate   
## 4 769911858 Existing Custo… 40 F 4 High School   
## 5 709106358 Existing Custo… 40 M 3 Uneducated   
## 6 713061558 Existing Custo… 44 M 2 Graduate   
## 7 810347208 Existing Custo… 51 M 4 Unknown   
## 8 818906208 Existing Custo… 32 M 0 High School   
## 9 710930508 Existing Custo… 37 M 3 Uneducated   
## 10 719661558 Existing Custo… 48 M 2 Graduate   
## # ℹ 10,117 more rows  
## # ℹ 17 more variables: Marital\_Status <chr>, Income\_Category <chr>,  
## # Card\_Category <chr>, Months\_on\_book <dbl>, Total\_Relationship\_Count <dbl>,  
## # Months\_Inactive\_12\_mon <dbl>, Contacts\_Count\_12\_mon <dbl>,  
## # Credit\_Limit <dbl>, Total\_Revolving\_Bal <dbl>, Avg\_Open\_To\_Buy <dbl>,  
## # Total\_Amt\_Chng\_Q4\_Q1 <dbl>, Total\_Trans\_Amt <dbl>, Total\_Trans\_Ct <dbl>,  
## # Total\_Ct\_Chng\_Q4\_Q1 <dbl>, Avg\_Utilization\_Ratio <dbl>, …

# Print the number of rows and columns  
  
print(paste("Number of rows:", nrow(bank\_data)))

## [1] "Number of rows: 10127"

print(paste("Number of columns:", ncol(bank\_data)))

## [1] "Number of columns: 23"

# Verify by using dim() - the dimensions r and c for dataset  
  
dim(bank\_data)

## [1] 10127 23

# 2. Clean the data by checking for missing values and removing any rows with missing values. Print the number of rows were omitted.

## Cleaning the data   
  
# Check for missing values   
  
missing\_values <- anyNA(bank\_data)  
  
# Print the result as a statement  
if (missing\_values) {  
 print("There are missing values in the dataset.")  
} else {  
 print("There are no missing values in the dataset.")  
}

## [1] "There are no missing values in the dataset."

## OR

# Alternative way to check missing values  
  
missing\_values\_check <- sum(is.na(bank\_data))  
  
print(paste("Number of missing values in the dataset:", missing\_values\_check))

## [1] "Number of missing values in the dataset: 0"

# Remove any rows with missing values   
  
clean\_bank\_data <-na.omit(bank\_data)   
  
print(clean\_bank\_data, n=100)

## # A tibble: 10,127 × 23  
## CLIENTNUM Attrition\_Flag Customer\_Age Gender Dependent\_count Education\_Level  
## <dbl> <chr> <dbl> <chr> <dbl> <chr>   
## 1 768805383 Existing Cust… 45 M 3 High School   
## 2 818770008 Existing Cust… 49 F 5 Graduate   
## 3 713982108 Existing Cust… 51 M 3 Graduate   
## 4 769911858 Existing Cust… 40 F 4 High School   
## 5 709106358 Existing Cust… 40 M 3 Uneducated   
## 6 713061558 Existing Cust… 44 M 2 Graduate   
## 7 810347208 Existing Cust… 51 M 4 Unknown   
## 8 818906208 Existing Cust… 32 M 0 High School   
## 9 710930508 Existing Cust… 37 M 3 Uneducated   
## 10 719661558 Existing Cust… 48 M 2 Graduate   
## 11 708790833 Existing Cust… 42 M 5 Uneducated   
## 12 710821833 Existing Cust… 65 M 1 Unknown   
## 13 710599683 Existing Cust… 56 M 1 College   
## 14 816082233 Existing Cust… 35 M 3 Graduate   
## 15 712396908 Existing Cust… 57 F 2 Graduate   
## 16 714885258 Existing Cust… 44 M 4 Unknown   
## 17 709967358 Existing Cust… 48 M 4 Post-Graduate   
## 18 753327333 Existing Cust… 41 M 3 Unknown   
## 19 806160108 Existing Cust… 61 M 1 High School   
## 20 709327383 Existing Cust… 45 F 2 Graduate   
## 21 806165208 Existing Cust… 47 M 1 Doctorate   
## 22 708508758 Attrited Cust… 62 F 0 Graduate   
## 23 784725333 Existing Cust… 41 M 3 High School   
## 24 811604133 Existing Cust… 47 F 4 Unknown   
## 25 789124683 Existing Cust… 54 M 2 Unknown   
## 26 771071958 Existing Cust… 41 F 3 Graduate   
## 27 720466383 Existing Cust… 59 M 1 High School   
## 28 804424383 Existing Cust… 63 M 1 Unknown   
## 29 718813833 Existing Cust… 44 F 3 Uneducated   
## 30 806624208 Existing Cust… 47 M 4 High School   
## 31 778348233 Existing Cust… 53 M 3 Unknown   
## 32 712991808 Existing Cust… 53 M 2 Uneducated   
## 33 709029408 Existing Cust… 41 M 4 Graduate   
## 34 788658483 Existing Cust… 53 F 2 College   
## 35 787937058 Existing Cust… 58 M 0 Graduate   
## 36 715318008 Existing Cust… 55 F 1 College   
## 37 713962233 Existing Cust… 55 F 3 Graduate   
## 38 785432733 Existing Cust… 42 F 4 High School   
## 39 715190283 Existing Cust… 57 F 1 Graduate   
## 40 708300483 Attrited Cust… 66 F 0 Doctorate   
## 41 827111283 Existing Cust… 45 M 3 Graduate   
## 42 758551608 Existing Cust… 51 M 2 Unknown   
## 43 773146383 Existing Cust… 50 F 1 College   
## 44 778493808 Existing Cust… 49 M 3 High School   
## 45 720572508 Existing Cust… 38 F 4 Graduate   
## 46 712661433 Existing Cust… 49 M 4 Uneducated   
## 47 789172683 Existing Cust… 56 M 2 Doctorate   
## 48 738406533 Existing Cust… 59 M 1 Doctorate   
## 49 799723908 Existing Cust… 46 M 3 High School   
## 50 771490833 Existing Cust… 52 M 1 College   
## 51 720756708 Existing Cust… 52 F 3 Unknown   
## 52 779471883 Attrited Cust… 54 F 1 Graduate   
## 53 711525033 Existing Cust… 66 F 0 High School   
## 54 712813458 Existing Cust… 49 M 2 Unknown   
## 55 714374133 Attrited Cust… 56 M 2 Graduate   
## 56 717891558 Existing Cust… 49 F 4 Graduate   
## 57 716632758 Existing Cust… 49 F 3 Graduate   
## 58 768563658 Existing Cust… 56 M 2 Uneducated   
## 59 711427458 Existing Cust… 44 F 5 Graduate   
## 60 714091983 Existing Cust… 42 M 2 High School   
## 61 787584108 Existing Cust… 55 M 3 Unknown   
## 62 712030833 Attrited Cust… 48 M 2 Graduate   
## 63 711481533 Existing Cust… 39 M 1 High School   
## 64 710082708 Existing Cust… 44 M 4 Post-Graduate   
## 65 708155733 Existing Cust… 53 M 2 High School   
## 66 788979258 Existing Cust… 51 M 4 Uneducated   
## 67 807986133 Existing Cust… 57 M 2 College   
## 68 788730933 Existing Cust… 44 F 2 Uneducated   
## 69 711314058 Existing Cust… 49 M 2 Graduate   
## 70 717975333 Existing Cust… 50 M 2 Doctorate   
## 71 715971108 Existing Cust… 51 M 4 Graduate   
## 72 720096558 Existing Cust… 55 F 2 Graduate   
## 73 719580033 Existing Cust… 54 M 1 Graduate   
## 74 820582308 Existing Cust… 42 M 5 Uneducated   
## 75 789973308 Existing Cust… 44 M 1 College   
## 76 712876233 Existing Cust… 53 M 2 Graduate   
## 77 804595158 Existing Cust… 44 F 4 Graduate   
## 78 714826758 Existing Cust… 37 F 3 Uneducated   
## 79 779058108 Existing Cust… 49 M 3 Graduate   
## 80 710790258 Existing Cust… 47 M 2 Graduate   
## 81 715623483 Existing Cust… 47 M 3 Graduate   
## 82 715156383 Existing Cust… 44 M 1 Unknown   
## 83 711013983 Attrited Cust… 55 F 4 Unknown   
## 84 755420433 Existing Cust… 59 F 1 Graduate   
## 85 794543958 Existing Cust… 53 M 1 Graduate   
## 86 716396358 Existing Cust… 52 M 2 Graduate   
## 87 715398033 Existing Cust… 53 M 2 High School   
## 88 711743883 Existing Cust… 43 F 3 Uneducated   
## 89 719720058 Existing Cust… 44 M 3 High School   
## 90 778992108 Existing Cust… 57 M 2 Unknown   
## 91 717539808 Existing Cust… 51 F 2 High School   
## 92 714070758 Existing Cust… 49 M 4 High School   
## 93 714107958 Existing Cust… 45 M 1 Graduate   
## 94 789140283 Existing Cust… 53 M 0 Graduate   
## 95 715550508 Existing Cust… 45 F 3 Unknown   
## 96 719712633 Existing Cust… 64 M 1 Graduate   
## 97 772629333 Existing Cust… 45 M 3 Graduate   
## 98 720336708 Existing Cust… 53 M 3 Doctorate   
## 99 802013583 Existing Cust… 56 M 3 College   
## 100 711887583 Attrited Cust… 47 M 2 Unknown   
## # ℹ 10,027 more rows  
## # ℹ 17 more variables: Marital\_Status <chr>, Income\_Category <chr>,  
## # Card\_Category <chr>, Months\_on\_book <dbl>, Total\_Relationship\_Count <dbl>,  
## # Months\_Inactive\_12\_mon <dbl>, Contacts\_Count\_12\_mon <dbl>,  
## # Credit\_Limit <dbl>, Total\_Revolving\_Bal <dbl>, Avg\_Open\_To\_Buy <dbl>,  
## # Total\_Amt\_Chng\_Q4\_Q1 <dbl>, Total\_Trans\_Amt <dbl>, Total\_Trans\_Ct <dbl>,  
## # Total\_Ct\_Chng\_Q4\_Q1 <dbl>, Avg\_Utilization\_Ratio <dbl>, …

# Print the number of rows omitted  
  
print(paste("Number of rows omitted due to missing values:", nrow(bank\_data) - nrow(clean\_bank\_data)))

## [1] "Number of rows omitted due to missing values: 0"

# 3. Calculate the mean and median credit limit of the customers.

# Mean credit limit of the customers   
  
mean\_credit\_limit <- mean(clean\_bank\_data$Credit\_Limit,na.rm = TRUE)  
  
print(paste("Mean credit limit:", currency(mean\_credit\_limit, symbol = "$", digits = 2, format = "f", big.mark = ",")))

## [1] "Mean credit limit: $8,631.95"

# Median credit limit of the customers   
  
median\_credit\_limit <- median(clean\_bank\_data$Credit\_Limit,na.rm = TRUE)  
  
print(paste("Median credit limit:", currency(median\_credit\_limit, symbol = "$", digits = 2, format = "f", big.mark = ",")))

## [1] "Median credit limit: $4,549.00"

# 4. Calculate the 25th and 75th percentile of the credit limit of the customers.

# Calculate 25th and 75th percentiles  
  
percentile\_25\_75 <- quantile(clean\_bank\_data$Credit\_Limit, probs = c(.25, .75), na.rm = TRUE)  
currency(percentile\_25\_75,symbol = "$", digits = 2, format = "f", big.mark = ",")

## 25% 75%   
## $2,555.00 $11,067.50

# Print the percentiles  
  
print(paste("25th percentile of credit limit:", currency(percentile\_25\_75[1], symbol = "$", digits = 2, format = "f", big.mark = ",")))

## [1] "25th percentile of credit limit: $2,555.00"

print(paste("75th percentile of credit limit:", currency(percentile\_25\_75[2], symbol = "$", digits = 2, format = "f", big.mark = ",")))

## [1] "75th percentile of credit limit: $11,067.50"

# 5. Calculate the range of the credit limit of the customers.

# range of the credit limit of the customers  
credit\_limit\_range <- range(clean\_bank\_data$Credit\_Limit,na.rm = TRUE)

# Print the range   
  
# Format minimum and maximum values of the range  
formatted\_min\_value <- currency(credit\_limit\_range[1], symbol = "$", digits = 2, format = "f", big.mark = ",")  
formatted\_max\_value <- currency(credit\_limit\_range[2], symbol = "$", digits = 2, format = "f", big.mark = ",")  
  
print(paste("Range of credit limit:", formatted\_min\_value, "to", formatted\_max\_value))

## [1] "Range of credit limit: $1,438.30 to $34,516.00"

# 6. Calculate the variance and standard deviation of the credit limit of the customers.

# variance of the credit limit of the customers  
  
credit\_limit\_variance <- var(clean\_bank\_data$Credit\_Limit,na.rm = TRUE)  
  
print(paste("Variance of credit limit:", currency(credit\_limit\_variance, symbol = "$", digits = 2, format = "f", big.mark = ",")))

## [1] "Variance of credit limit: $82,605,861.00"

# standard deviation of the credit limit of the customers  
  
credit\_limit\_sd <- sd(clean\_bank\_data$Credit\_Limit,na.rm = TRUE)  
  
print(paste("Standard deviation of credit limit:", currency(credit\_limit\_sd, symbol = "$", digits = 2, format = "f", big.mark = ",")))

## [1] "Standard deviation of credit limit: $9,088.78"

# 7. Calculate the correlation between the customer age and their credit limit. Print the result to the console.

# correlation between the customer age and their credit limit  
correlation\_age\_credit\_limit <- cor(clean\_bank\_data$Customer\_Age, clean\_bank\_data$Credit\_Limit)

# Print the result to the console  
  
print(paste("Correlation between customer age and credit limit:", round(correlation\_age\_credit\_limit,8)))

## [1] "Correlation between customer age and credit limit: 0.00247623"

# 8. Subset the data to only include customers with a credit limit greater than $10,000. Store the result in a new tibble data frame. Print the number of rows.

# Subset data for customers with credit limit > $10,000  
  
high\_credit\_customers <- subset(clean\_bank\_data, Credit\_Limit > 10000)

# Print the number of rows in the subset  
  
print(paste("Number of rows with credit limit greater than", currency(10000, digit = 0), ":", comma(nrow(high\_credit\_customers), digits = 0)))

## [1] "Number of rows with credit limit greater than $10,000 : 2,754"

# 9. Calculate the mean and standard deviation of the age of the customers that you identified in question 8.

# Calculate mean and standard deviation of age for high credit customers  
  
mean\_age\_high\_cr\_cus <- mean(high\_credit\_customers$Customer\_Age,na.rm = TRUE)  
sd\_age\_high\_cr\_cus <- sd(high\_credit\_customers$Customer\_Age,na.rm = TRUE)

# Print the mean and standard deviation of age for high credit customers  
  
print(paste("Mean age of customers with credit limit greater than", currency(10000, digit = 0), ":", round(mean\_age\_high\_cr\_cus, 2), "years." ))

## [1] "Mean age of customers with credit limit greater than $10,000 : 46.33 years."

print(paste("Standard deviation of age of customers with credit limit greater than", currency(10000, digit = 0), ":", round(sd\_age\_high\_cr\_cus, 2), "years."))

## [1] "Standard deviation of age of customers with credit limit greater than $10,000 : 7.32 years."