Store Records of an Electronic Store

Faleke Micheal Kayode

2026-02-06

##Importing the dataset into R markdown and installing the neccesary R libaries

```
messy_retail_sales_1000_rows <- read.csv("C:/Users/HP/Downloads/messy_retail_sales_1000_rows.csv")

##installing the neccessary R libaries
library(tidyverse)

## Warning: package 'lubridate' was built under R version 4.5.2

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.4     ✔ readr     2.1.5
## ✔ forcats   1.0.0     ✔ stringr   1.5.1
## ✔ ggplot2   4.0.0     ✔ tibble    3.3.0
## ✔ lubridate 1.9.4     ✔ tidyr     1.3.1
## ✔ purrr     1.1.0
## ── 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
```

##Renaming the dataset for easy recall

```
Retail_sales <- messy_retail_sales_1000_rows
```

##Checking the structure of the dataset

```
glimpse(Retail_sales)

## Rows: 1,050
## Columns: 7
## $ Order.ID      <int> 1102, 1435, 1270, 1106, 1071, 1020, 1121, 1466, 1214, 13…
## $ orderDate     <chr> "2023-03-15", "2023/02/10", "2023/02/10", "2023-01-05", …
## $ Customer.NAME <chr> "", "", "David", "Bob", "", "Eve", "", "", "", "Alice", …
## $ Product.Name  <chr> "Headphones", "Laptop", "Monitor", "Headphones", "Tablet…
## $ Quantity      <chr> "3", "four", "2", "3", "five", "1", "3", "3", "five", "f…
## $ Unit_Price... <int> 350, 200, 350, 1200, 350, 200, NA, 1200, 200, 350, 450, …
## $ Total.Sales   <int> 800, 450, 200, NA, 350, 450, 800, NA, 200, 1200, 350, 80…
```

##Inspecting the first 10 populations of the dataset

```
head(Retail_sales)
```

```
##   Order.ID  orderDate Customer.NAME Product.Name Quantity Unit_Price...
## 1     1102 2023-03-15                 Headphones        3           350
## 2     1435 2023/02/10                     Laptop     four           200
## 3     1270 2023/02/10         David      Monitor        2           350
## 4     1106 2023-01-05           Bob   Headphones        3          1200
## 5     1071 15-04-2023                     Tablet     five           350
## 6     1020 05/01/2023           Eve      Monitor        1           200
##   Total.Sales
## 1         800
## 2         450
## 3         200
## 4          NA
## 5         350
## 6         450
```

```
##Inspecting the last 10 populations
tail(Retail_sales)
```

```
##      Order.ID  orderDate Customer.NAME Product.Name Quantity Unit_Price...
## 1045     1418 2023-03-15                 Headphones     four           350
## 1046     1250 05/01/2023           Bob        Phone        3           200
## 1047     1531 2023-01-05         Frank       Tablet     four          1200
## 1048     1273 2023-01-05         Alice   Headphones                     NA
## 1049     1143 15-04-2023         Alice   Headphones        3          1200
## 1050     1312 15-04-2023       Charlie      Monitor     four            NA
##      Total.Sales
## 1045         450
## 1046        1200
## 1047        1200
## 1048         800
## 1049          NA
## 1050        1200
```

##Inspecting the dataset for any missing value

```
any(is.na(Retail_sales))
```

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

```
##checking for the count of data missing from the dataset
sum(is.na(Retail_sales))
```

```
## [1] 363
```

##checking if there missing data in every row

```
rowSums(is.na(Retail_sales))
```

```
##    [1] 0 0 0 1 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0 1 1 0 1 0 1 0 1 1 0 0 0 1 0 1 1
##   [38] 0 0 0 1 1 0 2 0 0 1 1 1 0 0 0 0 0 1 1 1 0 1 0 0 0 0 1 0 1 0 1 1 1 1 1 0 1
##   [75] 0 0 0 0 0 0 0 1 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 1 1 1 0 0 0 0 1 1
##  [112] 0 0 0 0 0 0 0 0 0 1 2 0 1 1 0 0 1 1 2 0 0 0 0 1 0 1 1 0 0 1 0 0 0 0 1 1 0
##  [149] 0 0 1 0 1 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0 1 0 0 1 1 0 2 0 0
##  [186] 0 1 0 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0 0 1 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0
##  [223] 0 1 0 1 0 0 1 1 0 0 1 0 0 0 0 1 1 0 0 0 1 0 0 0 0 1 2 1 0 0 0 1 0 0 0 0 0
##  [260] 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 1 1 0 0 0 1 0 2 0 0 0 1 0 0 0 0
##  [297] 1 0 1 0 0 0 0 0 2 1 1 0 0 0 0 0 1 2 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0
##  [334] 0 0 1 0 1 0 2 1 0 0 0 0 1 0 1 0 1 0 0 1 0 1 0 1 0 1 0 0 1 0 0 0 0 0 1 0 0
##  [371] 0 1 1 0 0 2 0 0 1 0 0 0 1 0 1 0 1 0 2 1 0 1 0 0 0 0 0 0 0 2 0 1 0 1 1 0 0
##  [408] 1 0 0 1 1 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1
##  [445] 0 1 0 1 0 0 1 1 0 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0
##  [482] 1 0 0 1 1 0 0 1 0 1 0 2 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 2 1 0 1 0
##  [519] 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0 0 1 1 0 0 1 1 0 0 1 0 0 2 0 1 1 0
##  [556] 2 0 0 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 0 2 0 1 0 1 0 0 1 0 0 1 1 0 0 0 0 1 0
##  [593] 1 1 0 0 1 1 0 1 0 1 0 1 0 0 0 1 0 0 0 0 1 0 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0
##  [630] 1 0 0 0 0 0 1 0 0 1 0 0 1 0 0 1 0 1 2 0 2 0 0 0 0 1 1 1 0 0 2 0 0 0 0 0 1
##  [667] 1 2 0 0 0 0 1 0 0 0 1 2 0 1 1 0 0 1 0 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0
##  [704] 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0
##  [741] 0 0 0 1 0 0 0 0 2 2 0 1 0 0 0 0 1 0 0 0 0 1 0 0 1 1 1 1 0 0 0 0 0 0 0 1 1
##  [778] 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1 0 0 1 1 0 1 0 1 0 0 0 1 0 1 1 1 1 1 1 0 0
##  [815] 2 1 0 0 0 1 1 1 0 0 0 1 1 0 1 2 0 0 0 0 0 0 0 1 0 1 1 1 0 1 0 1 0 1 0 0 0 1
##  [852] 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 1 2 1 0 1 1 0 0 0 0 1 0 1 0 1 1 0 0 0 1 0 0
##  [889] 1 1 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0
##  [926] 1 0 1 0 1 0 0 0 2 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 2 0
##  [963] 0 0 1 1 0 1 0 1 0 0 1 0 0 1 0 0 0 0 1 0 1 1 0 0 0 1 0 1 0 0 0 0 0 1 1 0 0
## [1000] 1 0 0 0 0 0 1 1 1 0 0 0 0 0 1 0 0 1 1 0 0 0 1 0 0 0 0 0 2 0 0 1 0 0 0 1 0
## [1037] 0 0 1 1 0 1 0 0 0 0 0 1 1 1
```

```
##Checking if there are any missing data in every column
colSums(is.na(Retail_sales))
```

```
##      Order.ID     orderDate Customer.NAME  Product.Name      Quantity
##             3             0             0             0             0
## Unit_Price...   Total.Sales
##           179           181
```

##Standardizing the name of every colum in the dataset

```
Retail_sales<- Retail_sales %>% rename(Order_ID = Order.ID, Order_Date = orderDate, Customer_Name = Customer.NAME,  Product_Label = Product.Name, Unit_Price = Unit_Price...)
```

##Cleaning and arranging the order id in chronological order to remove all inconsistent formatting

```
Retail_sales$Order.ID<- seq_len(nrow(Retail_sales))+ 1000

##checking if there are any missing values in the order id column after the cleaning
colSums(is.na(Retail_sales))
```

```
##      Order_ID    Order_Date Customer_Name Product_Label      Quantity
##             3             0             0             0             0
##    Unit_Price   Total.Sales      Order.ID
##           179           181             0
```

##cleaning the order date column to remove all inconsistent formatting and arranging the dates to follow chronological order starting from 14th febuary 2023 to 20th september 2025

```
Retail_sales <- Retail_sales %>% mutate(Order_Date = as.Date(Order_Date, tryFormats = c("%Y-%m-%d", "%d-%m-%Y", "%d/%m/%Y"))) %>% filter(Order_Date>= as.Date("2023-02-14"), Order_Date<= as.Date("2025-09-20")) %>% arrange(Order_Date)
```

##Removing the duplicate dataset

```
Retail_sales <- Retail_sales%>% select(-Order.ID)
```

##A brief summary of the column

```
summary(Retail_sales$Customer_Name)

##    Length     Class      Mode 
##       186 character character

##Identifying the names of the customers in the column available
unique(Retail_sales$Customer_Name)

## [1] ""        "Bob"     "David"   "Alice"   "Eve"     "Charlie" "Frank"
```

##creating a pool of missing names with in the column

```
additional_names<- c("Grace", "Henry", "Ivy", "Jack", "Kemi","Liam", "Maya", "Noah", "Olivia", "Paul", "Akoma", "Nath", "Prince", "Bob", "David", "Alice", "Eve", "Charlie","Frank", "Deborah", "Shola","lois","Louis", "Peter", "Dan","Mike", "Funke", "Aramide", "Iremide", "Jon", "Declan")
new_names<- c(Retail_sales$Customer_Name, additional_names)
```

##Assigning the names to the customer name coulmn in the dataset

```
set.seed(123)
Retail_sales$Customer_Name <- sample(new_names, size = nrow(Retail_sales), replace = TRUE)

##some rows within the column are still not populated so we identify the if there are any empty rows
sum(is.na(Retail_sales$Customer_Name) | Retail_sales$Customer_Name == "")

## [1] 19

##so therefore 16 rows/ cells within the customer name column are not populated with a name

##Fixing the missing cells within the dataset
Retail_sales <- Retail_sales %>% mutate(Customer_Name = ifelse(is.na(Customer_Name) | Customer_Name == "", sample(new_names, n(), replace = TRUE),Customer_Name))

##Rechecking if there are any empty or unfiled cells in the column
sum(is.na(Retail_sales$Customer_Name) | Retail_sales$Customer_Name == "")
```

```
## [1] 3
```

##Converting the quantity column to all numeric

```
Retail_sales$Quantity<- as.numeric(Retail_sales$Quantity)

## Warning: NAs introduced by coercion
```

##clearing the nulls from each cell of the unit price column

```
set.seed(123)

Retail_sales <-Retail_sales %>% mutate(Quantity = ifelse(is.na(Quantity), sample(c(1,2,3), n(),
 replace = TRUE), Quantity))


##cleaning the unit price column to remove the nulls
set.seed(123)

valid_values <- c(150, 200, 350, 400, 460, 500)

idx <- !(Retail_sales$Unit_Price %in% valid_values)

Retail_sales$Unit_Price[idx] <- sample(
  valid_values,
  size = sum(idx),
  replace = TRUE)
```

##mulitpying the quantity and unit oprice columns together to get an accurate view on total sales

```
Retail_sales <- Retail_sales %>% mutate(Total.Sales = Quantity * Unit_Price)
```