Generating and Preprocessing synthetic data for Woolworths

Deepthi Suresh

17 September 2023

Setup

```r
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
```

```r
library(magrittr)
library(randomNames)
library(Hmisc)
```

```
##
## Attaching package: ’Hmisc’

## The following objects are masked from ’package:dplyr’:
##
##     src, summarize

## The following objects are masked from ’package:base’:
##
##     format.pval, units
```

Video presentation

Link : https://rmit-arc.instructuremedia.com/embed/a7bcd5c8-c9c9-4468-8746-c9b5a08c120b

Data generation

```r
#Creating data set 1 called Inventory Data set
#Variable 1 - Product ID
Product_id <- c(1:100)
head(Product_id)
```

```
## [1] 1 2 3 4 5 6
```

```r
#Variable 2 - Product Category
Category <- c(  "Fruits and vegetables",
                "bakery",
                "poultry",
                "dairy eggs & fridge",
                "Pantry",
                "Freezer",
                "Beauty",
                "Home & Lifestyle",
                "Cleaning and Maintenance",
                "Pet")
set.seed(3455)
Product_Category <- sample(Category,100,replace = TRUE)
head(Product_Category)
```

```
## [1] "Home & Lifestyle"    "Pantry"              "Beauty"
## [4] "Home & Lifestyle"    "bakery"              "dairy eggs & fridge"
```

```r
#creating a draft of the Inventory dataset
inventory_dataset <- data.frame(Product_id,Category)

# variable 3 - Product Names
#creating vectors with different product names for each Category
FV<- c("onions","tomato", "apples","banana","garlic" )
B <- c("cookies","cupcakes","croissant","bread","cakes")
P <- c("poultry","meat","seafood","BBQ meat")
D <- c("cheese","milk","eggs","dips","yoghurt")
PT <- c( "tea & coffee","baking","Herbs","oil","pasta")
Fr <- c("Ice cream", "frozen fruits","frozen meals","frozen meat","frozen pizza")
Be <- c("sun protection","dental care", "hair care","skin care")
HL <- c("Dining","party Supplies","Storage","Home Decor","Toys")
CM<- c("laundry","kitchen","pest control","hardware")
pet <- c("cat & kitten","Dog & puppy","Bird, Fish & Small & pets")

set.seed(799)
inventory_dataset <- inventory_dataset %>%
  mutate(Product_name = ifelse(Category=="Fruits and vegetables",sample(FV,nrow(inventory_dataset),repla
            ifelse(Category == "bakery",sample(B,nrow(inventory_dataset),replace= TRUE),
            ifelse(Category == "poultry",sample(P,nrow(inventory_dataset),replace= TRUE),
            ifelse(Category == "Pantry",sample(PT,nrow(inventory_dataset),replace= TRUE),
          ifelse(Category == "Freezer",sample(Fr,nrow(inventory_dataset),replace= TRUE),
          ifelse(Category == "Beauty",sample(Be,nrow(inventory_dataset),replace= TRUE),
          ifelse(Category == "dairy eggs & fridge",sample(D,nrow(inventory_dataset),replace= TRUE),
          ifelse(Category == "Home & Lifestyle",sample(HL,nrow(inventory_dataset),replace= TRUE),
                                                                             ifelse(Category == "Cleanin
                                                                                     ifelse(Category ==

head(inventory_dataset$Product_name)
```

```
## [1] "apples"       "bread"        "BBQ meat"     "cheese"       "oil"
## [6] "frozen pizza"
```

```
#Variable no 4 -Quantity in stock
set.seed(98786)
Quantity_stock <- round(runif(100,min = 30, max = 150),0)

head(Quantity_stock)
```

```
## [1]  41 143  69  86  88  82
```

```
# variable no 5 - supplier
set.seed(67588)
Supplier <- randomNames(100)
head(Supplier)
```

```
## [1] "Chacon, Jonathan"   "el-Basher, Shaahir" "Lewis, Raycell"
## [4] "Kingery, Robert"    "Ho, David"          "Olsen, Li"
```

```
# Combining the variables to form the inventory data set
inventory_dataset <- data.frame(inventory_dataset,Quantity_stock,Supplier)
head(inventory_dataset)
```

```
##   Product_id              Category Product_name Quantity_stock
## 1          1 Fruits and vegetables       apples             41
## 2          2                bakery        bread            143
## 3          3               poultry     BBQ meat             69
## 4          4   dairy eggs & fridge       cheese             86
## 5          5                Pantry          oil             88
## 6          6               Freezer frozen pizza             82
##              Supplier
## 1   Chacon, Jonathan
## 2 el-Basher, Shaahir
## 3     Lewis, Raycell
## 4    Kingery, Robert
## 5          Ho, David
## 6          Olsen, Li
```

```
#Adding missing values to the Supplier Variable by introducing a rand variable
set.seed(5647)
inventory_dataset %<>%
  mutate(rand = runif(100, min = 0, max = 1))
inventory_dataset %<>%
  mutate(Supplier = case_when(rand >= 0.05 ~ Supplier))

# removing rand column
inventory_dataset<- inventory_dataset[,-6]

sum(is.na(inventory_dataset$Supplier))
```

```
## [1] 4
```

```r
head(inventory_dataset)
```

```
##   Product_id              Category Product_name Quantity_stock
## 1          1 Fruits and vegetables       apples             41
## 2          2                bakery        bread            143
## 3          3               poultry     BBQ meat             69
## 4          4   dairy eggs & fridge       cheese             86
## 5          5                Pantry          oil             88
## 6          6               Freezer frozen pizza             82
##             Supplier
## 1   Chacon, Jonathan
## 2 el-Basher, Shaahir
## 3     Lewis, Raycell
## 4    Kingery, Robert
## 5          Ho, David
## 6          Olsen, Li
```

```r
#Generating Dataset 2 called Sales Dataset
#Variable 1: Order No
set.seed(321434)
Order_No <- sample(round(runif(50, min = 40000, max = 50000),0),100,replace = TRUE)
head(Order_No)
```

```
## [1] 41246 41508 41837 45538 47904 46029
```

```r
#Variable 2 : Sale date
set.seed(4456)
date <- Sys.Date() - sort(sample(1:1000, 100))
head(date)
```

```
## [1] "2024-02-24" "2024-01-27" "2024-01-17" "2024-01-06" "2023-12-10"
## [6] "2023-11-17"
```

```r
#Variable 3: Quantity
set.seed(234656)
Sales_Quantity <- round(runif(100,min = 1, max = 25),0)
head(Sales_Quantity)
```

```
## [1]  7  7  6  5 20 10
```

```r
# Variable 4: Price per unit, after discount
set.seed(79789)
Price <- round(runif(100,min = 1, max = 25),2)
head(Price)
```

```
## [1] 22.29  8.15  7.20  7.57  3.67 11.74
```

```r
#Variable 5: Payment Methods
set.seed(475684)
Payment_method <- sample(1:4,100,replace = TRUE)
head(Payment_method)
```

```
## [1] 4 3 4 2 2 2
```

```r
#Variable 6: Discount Amount
set.seed(73245)
Discount <- sample(round(runif(50, min = 0, max = 15),2),100,replace = TRUE)
head(Discount)
```

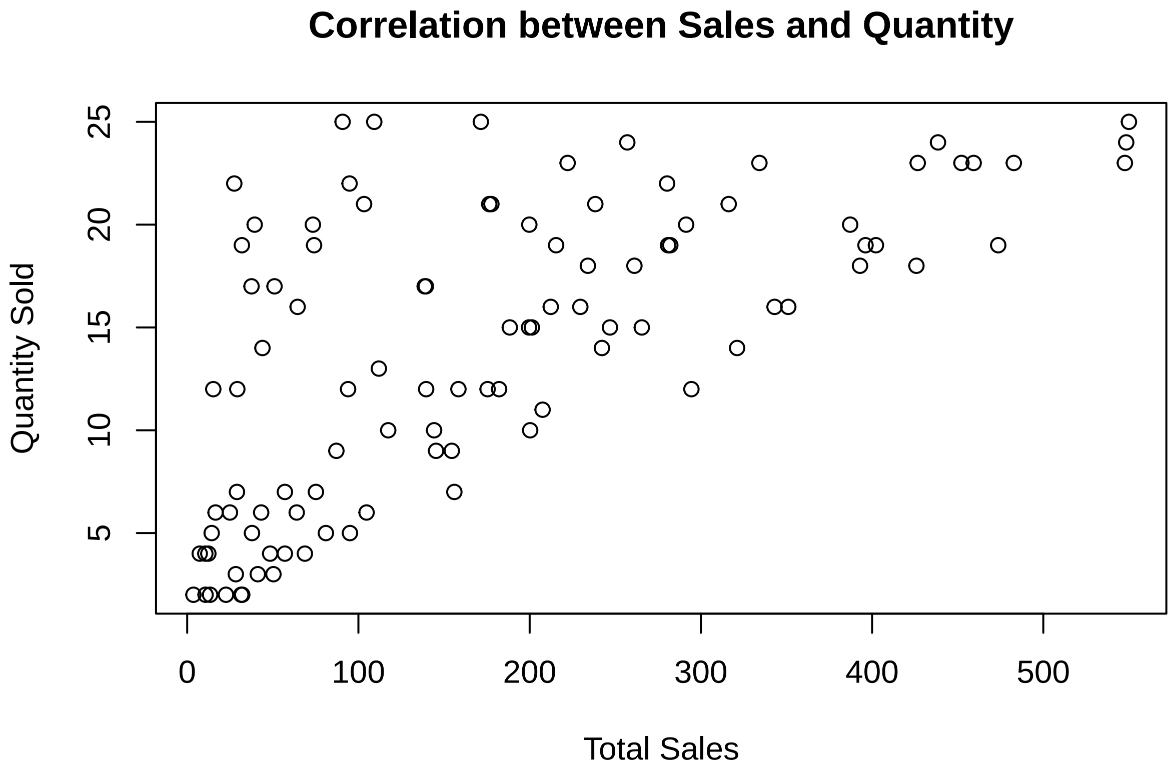
```
## [1]  3.90 11.37 10.37 14.38  6.92  4.61
```

```r
#Variable 7: Productid
productid <- sample(round(runif(100, min = 0, max = 100),0),100,replace = FALSE)
head(productid)
```

```
## [1] 84 23 70 99 36 55
```

```r
#generating a temporary dataset
Sales_Dataset <- data.frame(Order_No,date,productid, Sales_Quantity,Price,Discount,Payment_method)

#variable 8: Total sales
# assuming pricing errors/importing errors and estimating 5 as sd
Sales_Dataset <- Sales_Dataset %>%
  mutate(Total_Sales = (Price * Sales_Quantity))
# rounding the sales data to 2 decimals
Sales_Dataset$Total_Sales <- round(Sales_Dataset$Total_Sales,2)
#Generating a plot to check the correlation
plot(x = Sales_Dataset$Total_Sales,
     y = Sales_Dataset$Sales_Quantity,
     main = "Correlation between Sales and Quantity",
     xlab = "Total Sales",
     ylab = "Quantity Sold")
```



```
#Adding Missing values in discount by using a rand variable
set.seed(9865)
Sales_Dataset %<>%
  mutate(rand = runif(100, min = 0, max = 1))
Sales_Dataset %<>%
  mutate(Discount = case_when(rand >= 0.05 ~ Discount))
# removing rand column
Sales_Dataset<- Sales_Dataset[,-9]
sum(is.na(Sales_Dataset$Discount)) # no of missing values
```

```
## [1] 6
```

```
head(Sales_Dataset)
```

```
##   Order_No       date productid Sales_Quantity Price Discount Payment_method
## 1    41246 2024-02-24        84              7 22.29     3.90              4
## 2    41508 2024-01-27        23              7  8.15    11.37              3
## 3    41837 2024-01-17        70              6  7.20       NA              4
## 4    45538 2024-01-06        99              5  7.57    14.38              2
## 5    47904 2023-12-10        36             20  3.67     6.92              2
## 6    46029 2023-11-17        55             10 11.74     4.61              2
##   Total_Sales
## 1      156.03
## 2       57.05
## 3       43.20
## 4       37.85
## 5       73.40
## 6      117.40
```

I have considered two datsets from Woolworths (the data is only a subset of the original dataset because of its sheer size.) The first dataset is the Inventory dataset containing information regarding the various product categories and product names that Woolworths carry, along with their supplier name and Stock in Hand.the datset consists of 100 observations and 5 variables as given below:

Product_id(NUM) - unique identifier of the products available

Category(CHR) - broad categorisation of the product

Product_Name(CHR) - individual product names

Quantity_stock(NUM) - the stock in hand for each product

Supplier(CHR) - the name of the supplier providing the products to Woolworths

In order to generate the variables, the following steps were taken:

Product_Id was generated using a vector of 1-100. A random sample from a vector containing a subset of Categories in Woolworths were introduced to generate the product category variable using the 'sample' function. In order to create the product names under each category, 10 vectors were created, from which a random sample would be assigned to each observation under each category using the 'Ifelse' and 'sample' function. Quantity _stock was generated using the 'runif' function keeping an assumption that stores have lead times and require a minimum of 30 units to order a new batch and maximum of 100 (scaling it down for the purpose of this assignment). 'Suppliers' variable was created using the 'randomnames' function. Once the variables were created it was combined to form a dataframe called Inventory_Dataset using the 'data.frame' function.

In order to induce missing values, the supplier variable was chosen. A random variable 'rand' was first created using the runif function.then a 'case_when' function was applied on supplier variable with a condition on the random variable of ">= 0.05". This then deletes 5% of the suppliers from the dataset,selected at random by converting to NAs where the 'rand' value is less than 0.05.

The second dataset is a Sales dataset of Woolworths. It consists of 100 observations with 8 variables as given below:

Order_No(NUM) - the transaction number for the purchase

Sale_Date(DATE) - the date on which the sale happened

Sales_Quantity(NUM) - the no of units purchased

Price(NUM) - the price of the products after discounts

Payment Methods(INT) - the method of payment used by the customers like Cash,card, online or giftcards

Discount(NUM) - the discount offered by Woolworths on individual items

Product_id(NUM) - id of the items purchased

Total_Sales(NUM) - correlated variable calculated using a formula indicating the total sales generated by the store

In order to generate the variables the following steps have been taken:

Order_No is generated using random sampling from a vector containing numbers between 40000 and 50000 (assuming the order numbers are a 5 digit code at Woolworths and we are creating only a sample of 100 observationa). Sales_Date is generated keeping the system date(using the 'sys.date' and random dates using the' runif' and 'sample' function. Quantity and Price are generated in a similar way using the 'runif' and 'sample' function. An assumption of maximum 25 quantities can be purchased, is maintained and the price range is assumed to be 1-25 AUD for the products in the dataset. The payment methods available to the customers are cash,credit card, online payments and gift cards- generated using the sample function. The discount that is offered is assumed to be capped at 15$ per product and a random sample is generated. Productid is a random sample of observations from 1-100.

In order to generate the correlated variable, the following formula is used

Total_Sales = (Price * Sales_Quantity).

An assumption of 5 SD is maintained for this variable. A scatter plot for the variables Quantity sold and Total Price are generated which shows a mildly positive correlation, which could mean that more the quantity sold more would be the total sales that are generated.

All the variables are then combined to form the Sales_Dataset

In order to add missing values in the dataset, the discount variable has been selected as this is not a variable that is being used for calculations in this particular study. Missing value is introduced using a random variable called 'rand' and 'case_when' function with a criteria of 'rand >= 0.05'. This then deletes 5% of the Discount from the dataset,selected at random by converting to NAs where the 'rand' value is less than 0.05.'Sum(is.na()' function is used to calculate the no of NA values in the discount variable

The 'head' of sales and Inventory dataset provides a glimpse of the newly created dataset

Merging data sets

```
merged1 <- left_join(inventory_dataset,Sales_Dataset,by = join_by
                        (Product_id==productid))
head(merged1,10)
```

```
##    Product_id              Category Product_name Quantity_stock
## 1           1 Fruits and vegetables       apples             41
## 2           2                bakery        bread            143
## 3           2                bakery        bread            143
## 4           3               poultry     BBQ meat             69
## 5           4   dairy eggs & fridge       cheese             86
## 6           5                Pantry          oil             88
## 7           6               Freezer frozen pizza             82
## 8           7                Beauty    hair care             87
## 9           7                Beauty    hair care             87
## 10          8      Home & Lifestyle       Dining            135
##              Supplier Order_No       date Sales_Quantity Price Discount
## 1     Chacon, Jonathan       NA       <NA>             NA    NA       NA
## 2   el-Basher, Shaahir    41837 2022-04-11             25  6.86     1.14
## 3   el-Basher, Shaahir    44366 2021-12-12             21 11.35     9.38
## 4       Lewis, Raycell    44790 2023-09-05             20  1.97     0.23
## 5      Kingery, Robert    48976 2022-04-24             21  4.92     8.55
## 6            Ho, David       NA       <NA>             NA    NA       NA
## 7            Olsen, Li       NA       <NA>             NA    NA       NA
## 8      Valadez, Desira    48494 2023-08-26              2 16.08     5.97
## 9      Valadez, Desira    41227 2022-03-19              6 10.66     9.38
## 10 Wang, Miriam Aileen    46560 2023-11-14             23 14.53     4.01
##    Payment_method Total_Sales
## 1              NA          NA
## 2               3      171.50
## 3               1      238.35
## 4                3       39.40
## 5                1      103.32
## 6               NA          NA
## 7               NA          NA
## 8                3       32.16
## 9                4       63.96
## 10               1      334.19
```

The sales and inventory dataset is then joined using the left join function with the key variable of Product_id in Inventory Dataset and productid from the sales Dataset. Since we have used a left join all the variable from the Inventory dataset will be a part of the merged data set but only the common variables from the Sales data will be part of the new data set. The new merged1 datset has 137 observations with 12 variables.

Checking structure of combined data

```
# str of the dataset
head(merged1)
```

```
##   Product_id              Category Product_name Quantity_stock
## 1          1 Fruits and vegetables       apples             41
## 2          2                bakery        bread            143
## 3          2                bakery        bread            143
## 4          3               poultry     BBQ meat             69
## 5          4   dairy eggs & fridge       cheese             86
## 6          5                Pantry          oil             88
##             Supplier Order_No       date Sales_Quantity Price Discount
## 1   Chacon, Jonathan       NA       <NA>             NA    NA       NA
## 2 el-Basher, Shaahir    41837 2022-04-11             25  6.86     1.14
## 3 el-Basher, Shaahir    44366 2021-12-12             21 11.35     9.38
## 4     Lewis, Raycell    44790 2023-09-05             20  1.97     0.23
## 5    Kingery, Robert    48976 2022-04-24             21  4.92     8.55
## 6          Ho, David       NA       <NA>             NA    NA       NA
##   Payment_method Total_Sales
## 1             NA          NA
## 2              3      171.50
## 3              1      238.35
## 4              3       39.40
## 5              1      103.32
## 6             NA          NA
```

```
dim(merged1)
```

```
## [1] 137  12
```

```
str(merged1)
```

```
## ’data.frame’:    137 obs. of  12 variables:
##  $ Product_id    : num  1 2 2 3 4 5 6 7 7 8 ...
##  $ Category      : chr  "Fruits and vegetables" "bakery" "bakery" "poultry" ...
##  $ Product_name  : chr  "apples" "bread" "bread" "BBQ meat" ...
##  $ Quantity_stock: num  41 143 143 69 86 88 82 87 87 135 ...
##  $ Supplier      : chr  "Chacon, Jonathan" "el-Basher, Shaahir" "el-Basher, Shaahir" "Lewis, Raycell
##  $ Order_No      : num  NA 41837 44366 44790 48976 ...
##  $ date          : Date, format: NA "2022-04-11" ...
##  $ Sales_Quantity: num  NA 25 21 20 21 NA NA 2 6 23 ...
##  $ Price         : num  NA 6.86 11.35 1.97 4.92 ...
##  $ Discount      : num  NA 1.14 9.38 0.23 8.55 NA NA 5.97 9.38 4.01 ...
##  $ Payment_method: int  NA 3 1 3 1 NA NA 3 4 1 ...
##  $ Total_Sales   : num  NA 171.5 238.3 39.4 103.3 ...
```

```
names(merged1)
```

```
##  [1] "Product_id"     "Category"       "Product_name"   "Quantity_stock"
##  [5] "Supplier"       "Order_No"       "date"           "Sales_Quantity"
##  [9] "Price"          "Discount"       "Payment_method" "Total_Sales"
```

```
#converting to factor
# converting payment methods from INT to Factor
unique(Payment_method)
```

```
## [1] 4 3 2 1
```

```
merged1$Payment_method <- merged1$Payment_method %>%
  factor(.,levels = c( 1,2,3,4),
         labels = c("credit card","gift card","online payment","cash"))
class(merged1$Payment_method)
```

```
## [1] "factor"
```

```
levels(merged1$Payment_method)
```

```
## [1] "credit card"    "gift card"      "online payment" "cash"
```

```
# converting Category from CHR to Factor
unique(Category)
```

```
##  [1] "Fruits and vegetables"    "bakery"
##  [3] "poultry"                  "dairy eggs & fridge"
##  [5] "Pantry"                   "Freezer"
##  [7] "Beauty"                   "Home & Lifestyle"
##  [9] "Cleaning and Maintenance" "Pet"
```

```
merged1$Category <- merged1$Category%>%
  factor(.,levels = c("Fruits and vegetables","bakery","poultry" ,
                  "dairy eggs & fridge","Pantry","Freezer",
                    "Beauty","Home & Lifestyle","Cleaning and Maintenance",
                    "Pet" ))
class(merged1$Category)
```

```
## [1] "factor"
```

```r
levels(merged1$Category)
```

```
##  [1] "Fruits and vegetables"    "bakery"
##  [3] "poultry"                  "dairy eggs & fridge"
##  [5] "Pantry"                   "Freezer"
##  [7] "Beauty"                   "Home & Lifestyle"
##  [9] "Cleaning and Maintenance" "Pet"
```

```r
# converting product_name from CHR to Factor
unique(merged1$Product_name)
```

```
##  [1] "apples"                      "bread"
##  [3] "BBQ meat"                    "cheese"
##  [5] "oil"                         "frozen pizza"
##  [7] "hair care"                   "Dining"
##  [9] "hardware"                    "Bird, Fish & Small & pets"
## [11] "garlic"                      "croissant"
## [13] "meat"                        "Ice cream"
## [15] "dental care"                 "Home Decor"
## [17] "kitchen"                     "banana"
## [19] "eggs"                        "tea & coffee"
## [21] "frozen meals"                "sun protection"
## [23] "Toys"                        "Dog & puppy"
## [25] "cupcakes"                    "seafood"
## [27] "Storage"                     "laundry"
## [29] "cakes"                       "Herbs"
## [31] "pest control"                "cookies"
## [33] "milk"                        "pasta"
## [35] "onions"                      "poultry"
## [37] "frozen meat"                 "party Supplies"
## [39] "skin care"                   "tomato"
## [41] "yoghurt"                     "frozen fruits"
## [43] "cat & kitten"
```

```r
merged1$Product_name <- merged1$Product_name%>%
  factor(.,levels = c( "apples","bread","seafood","cheese","Herbs","frozen pizza",
                       "sun protection","party Supplies","kitchen","Bird, Fish & Small & pets","garlic"
                       "meat","baking","frozen meat","skin care","Dining","cat & kitten",
                       "banana", "cookies","milk","oil","frozen meals","Home Decor",
                       "hardware","Dog & puppy","onions","BBQ meat","dips","tea & coffee",
                       "dental care","Storage","cakes","yoghurt","Toys","poultry",
                       "pasta","Ice cream","hair care","pest control","tomato","frozen fruits"))

class(merged1$Product_name)
```

```
## [1] "factor"
```

```r
levels(merged1$Product_name)
```

```
##  [1] "apples"                      "bread"
##  [3] "seafood"                     "cheese"
##  [5] "Herbs"                        "frozen pizza"
##  [7] "sun protection"               "party Supplies"
##  [9] "kitchen"                      "Bird, Fish & Small & pets"
## [11] "garlic"                       "croissant"
## [13] "meat"                         "baking"
## [15] "frozen meat"                  "skin care"
## [17] "Dining"                       "cat & kitten"
## [19] "banana"                       "cookies"
## [21] "milk"                         "oil"
## [23] "frozen meals"                 "Home Decor"
## [25] "hardware"                     "Dog & puppy"
## [27] "onions"                       "BBQ meat"
## [29] "dips"                         "tea & coffee"
## [31] "dental care"                  "Storage"
## [33] "cakes"                        "yoghurt"
## [35] "Toys"                         "poultry"
## [37] "pasta"                        "Ice cream"
## [39] "hair care"                    "pest control"
## [41] "tomato"                       "frozen fruits"
```

```
str(merged1)
```

```
## 'data.frame':    137 obs. of  12 variables:
##  $ Product_id    : num  1 2 2 3 4 5 6 7 7 8 ...
##  $ Category      : Factor w/ 10 levels "Fruits and vegetables",..: 1 2 2 3 4 5 6 7 7 8 ...
##  $ Product_name  : Factor w/ 42 levels "apples","bread",..: 1 2 2 28 4 22 6 39 39 17 ...
##  $ Quantity_stock: num  41 143 143 69 86 88 82 87 87 135 ...
##  $ Supplier      : chr  "Chacon, Jonathan" "el-Basher, Shaahir" "el-Basher, Shaahir" "Lewis, Raycell
##  $ Order_No      : num  NA 41837 44366 44790 48976 ...
##  $ date          : Date, format: NA "2022-04-11" ...
##  $ Sales_Quantity: num  NA 25 21 20 21 NA NA 2 6 23 ...
##  $ Price         : num  NA 6.86 11.35 1.97 4.92 ...
##  $ Discount      : num  NA 1.14 9.38 0.23 8.55 NA NA 5.97 9.38 4.01 ...
##  $ Payment_method: Factor w/ 4 levels "credit card",..: NA 3 1 3 1 NA NA 3 4 1 ...
##  $ Total_Sales   : num  NA 171.5 238.3 39.4 103.3 ...
```

The structure of the merged dataset shows that there are 137 observations and 12 variables. The category, product name and payment methods seems like ideal candidates to convert into Factor variable. The 'names' function shows the names of the different variables

In order to convert the payment method to factor variable, we have first identified the unique values in the variable using the 'unique' function and then we have used the 'factor' function along with levels to identify the different levels 1,2,3,4 and the corresponding labels for each of these levels such as "credit card", "gift card", "online payment", "cash". Once the factor conversion is done, the 'class' function confirms that the payment method is converted into factor and the 'levels' function shows the new labels. The above process is followed for the category variable and product names.

The 'str' function shows the updated structure of the merged dataset with category, product name and payment methods being a Factor variable.

Generate summary statistics

```r
# summarising the total sales based on category
Summary_TS <-merged1 %>%
  group_by(Category) %>%
  summarise(min = round(min(Total_Sales,na.rm = TRUE),2),
                   first_quantile = round(quantile(Total_Sales, 0.25,na.rm = TRUE),2),
                   Median = round(median(Total_Sales,na.rm = TRUE),2),
                   Mean = round(mean(Total_Sales,na.rm = TRUE),2),
                   third_quantile = round(quantile(Total_Sales,0.75,na.rm = TRUE),2),
                   max = round(max(Total_Sales,na.rm = TRUE),2),
                   stand_dev = round(sd(Total_Sales,na.rm = TRUE),2),
                   n = n(),
                   missing = sum(is.na(Total_Sales)))%>%
  ungroup()
head(Summary_TS,10)
```

```
## # A tibble: 10 x 10
##    Category       min first_quantile Median  Mean third_quantile   max stand_dev
##    <fct>        <dbl>          <dbl>  <dbl> <dbl>          <dbl> <dbl>     <dbl>
##  1 Fruits and~  10.6            41.1  114.   114.           191.  215.      88.7
##  2 bakery      172.           196.   265.   307.           428.  483.     140.
##  3 poultry      13.4            81     139.   133.           200.  234        74.8
##  4 dairy eggs~ 103.           121.   201.   247.           299.  550       170
##  5 Pantry        3.64           44.8   75.2  129.           191.  387.     125.
##  6 Freezer      10.7           109.   261.   267.           416.  548.     179.
##  7 Beauty       12.4            31.5   64.0  145.           175.  548.     169.
##  8 Home & Lif~   7.32           79.1  234.   205.           331.  396.     148.
##  9 Cleaning a~  25.0            40.9  178.   148.           237.  281.     111.
## 10 Pet          14.3            37.6  112.   161.           200.  427.     147.
## # i 2 more variables: n <int>, missing <int>
```

```r
# summarising the total sales based on Payment methods
Summary_PM <-merged1 %>%
  group_by(Payment_method) %>%
  summarise(min = round(min(Total_Sales,na.rm = TRUE),2),
            first_quantile = round(quantile(Total_Sales, 0.25,na.rm = TRUE),2),
            Median = round(median(Total_Sales,na.rm = TRUE),2),
            Mean = round(mean(Total_Sales,na.rm = TRUE),2),
            third_quantile = round(quantile(Total_Sales,0.75,na.rm = TRUE),2),
            max = round(max(Total_Sales,na.rm = TRUE),2),
            stand_dev = round(sd(Total_Sales,na.rm = TRUE),2),
            n = n(),
            missing = sum(is.na(Total_Sales)))%>% na.omit(Summary_PM) %>%
  ungroup()
```

```
## Warning: There were 2 warnings in ‘summarise()‘.
## The first warning was:
## i In argument: ‘min = round(min(Total_Sales, na.rm = TRUE), 2)‘.
## i In group 5: ‘Payment_method = NA‘.
## Caused by warning in ‘min()‘:
## ! no non-missing arguments to min; returning Inf
## i Run ‘dplyr::last_dplyr_warnings()‘ to see the 1 remaining warning.
```

```
head(Summary_PM)
```

```
## # A tibble: 4 x 10
##   Payment_method   min first_quantile Median  Mean third_quantile   max
##   <fct>          <dbl>          <dbl>  <dbl> <dbl>          <dbl> <dbl>
## 1 credit card     3.64           69.0  175.   197.           254.  548.
## 2 gift card      10.6            37.7   95.0  146.           189.  474.
## 3 online payment  7.32           41.2  201.   202.           294.  548.
## 4 cash           14.3            48.4  112.   159.           200.  550
## # i 3 more variables: stand_dev <dbl>, n <int>, missing <int>
```

In order to summarize the data, we have performed two summary statistics on Total sales based on category and Payment Methods

The summarize function is used here to generate summary statistics and saved into 'Summary__TS' data. The total sales is calculated based on the 10 category that we have used in the dataset. The following summary statistics are being used here - min, Quartile 1, Median, Mean, Quartile 3, Max, Standard Deviation, N and no of missing values. We have also used Na.rm function to remove the NA values for numerical calculations. The same process have been followed to create 'Summary__PM', a summary statistics for Total sales grouped by payment method.

On an average majority of the payment are made through online payment, followed by giftcard, credit card and cash.The maximum amount spent is using cash worth of 550$ whereas the least amount spent is 3.64$.

According to the Summary__TS data, highest average sale is from bakery category followed by Freezer.Pantry and poultry seems to be a low performing category with a mean of 129.30 and 132.83.Max sale in a single purchase is from the Beauty and Dairy department worth of 548$ and 550$

Scanning data

```
# Using head to check the variables with missing values
head(merged1)
```

```
##   Product_id              Category Product_name Quantity_stock
## 1          1 Fruits and vegetables       apples             41
## 2          2                bakery        bread            143
## 3          2                bakery        bread            143
## 4          3               poultry     BBQ meat             69
## 5          4   dairy eggs & fridge       cheese             86
## 6          5                Pantry          oil             88
##             Supplier Order_No       date Sales_Quantity Price Discount
## 1   Chacon, Jonathan       NA       <NA>             NA    NA       NA
## 2 el-Basher, Shaahir    41837 2022-04-11             25  6.86     1.14
## 3 el-Basher, Shaahir    44366 2021-12-12             21 11.35     9.38
## 4     Lewis, Raycell    44790 2023-09-05             20  1.97     0.23
## 5    Kingery, Robert    48976 2022-04-24             21  4.92     8.55
## 6          Ho, David       NA       <NA>             NA    NA       NA
##   Payment_method Total_Sales
## 1           <NA>          NA
## 2 online payment      171.50
## 3    credit card      238.35
## 4 online payment       39.40
## 5    credit card      103.32
## 6           <NA>          NA
```

```
colSums(is.na(merged1))
```

```
##     Product_id       Category   Product_name Quantity_stock       Supplier
##              0              0              8              0              4
##       Order_No           date Sales_Quantity          Price       Discount
##             37             37             37             37             43
## Payment_method    Total_Sales
##             37             37
```

```
#Imputing missing values of Discount using mean vale
merged1$Discount <- round(impute(merged1$Discount,fun = mean),2)
head(merged1)
```

```
##   Product_id              Category Product_name Quantity_stock
## 1          1 Fruits and vegetables       apples             41
## 2          2                bakery        bread            143
## 3          2                bakery        bread            143
## 4          3               poultry     BBQ meat             69
## 5          4   dairy eggs & fridge       cheese             86
## 6          5                Pantry          oil             88
##             Supplier Order_No       date Sales_Quantity Price Discount
## 1   Chacon, Jonathan       NA       <NA>             NA    NA     7.77
## 2 el-Basher, Shaahir    41837 2022-04-11             25  6.86     1.14
## 3 el-Basher, Shaahir    44366 2021-12-12             21 11.35     9.38
## 4     Lewis, Raycell    44790 2023-09-05             20  1.97     0.23
## 5    Kingery, Robert    48976 2022-04-24             21  4.92     8.55
## 6          Ho, David       NA       <NA>             NA    NA     7.77
##   Payment_method Total_Sales
## 1           <NA>          NA
## 2 online payment      171.50
## 3    credit card      238.35
## 4 online payment       39.40
## 5    credit card      103.32
## 6           <NA>          NA
```

```
# bringing back Na in discount where there are no order details
merged1$Discount[is.na(merged1$Order_No)]<-NA

# deleting supplier values with missing values
merged1 <-merged1[complete.cases(merged1),]

# checking the dataset
head(merged1)
```

```
##   Product_id            Category Product_name Quantity_stock           Supplier
## 2          2              bakery        bread            143 el-Basher, Shaahir
## 3          2              bakery        bread            143 el-Basher, Shaahir
## 4          3             poultry     BBQ meat             69     Lewis, Raycell
## 5          4 dairy eggs & fridge       cheese             86    Kingery, Robert
## 8          7              Beauty    hair care             87    Valadez, Desira
## 9          7              Beauty    hair care             87    Valadez, Desira
##   Order_No       date Sales_Quantity Price Discount Payment_method Total_Sales
## 2    41837 2022-04-11             25  6.86     1.14 online payment      171.50
## 3    44366 2021-12-12             21 11.35     9.38    credit card      238.35
## 4    44790 2023-09-05             20  1.97     0.23 online payment       39.40
## 5    48976 2022-04-24             21  4.92     8.55    credit card      103.32
## 8    48494 2023-08-26              2 16.08     5.97 online payment       32.16
## 9    41227 2022-03-19              6 10.66     9.38           cash       63.96
```

```r
str(merged1)
```

```
## ’data.frame':    95 obs. of  12 variables:
##  $ Product_id    : num  2 2 3 4 7 7 8 11 12 13 ...
##  $ Category      : Factor w/ 10 levels "Fruits and vegetables",..: 2 2 3 4 7 7 8 1 2 3 ...
##  $ Product_name  : Factor w/ 42 levels "apples","bread",..: 2 2 28 4 39 39 17 11 12 13 ...
##  $ Quantity_stock: num  143 143 69 86 87 87 135 67 54 122 ...
##  $ Supplier      : chr  "el-Basher, Shaahir" "el-Basher, Shaahir" "Lewis, Raycell" "Kingery, Robert"
##  $ Order_No      : num  41837 44366 44790 48976 48494 ...
##  $ date          : Date, format: "2022-04-11" "2021-12-12" ...
##  $ Sales_Quantity: num  25 21 20 21 2 6 23 19 23 16 ...
##  $ Price         : num  6.86 11.35 1.97 4.92 16.08 ...
##  $ Discount      : ’impute’ num  1.14 9.38 0.23 8.55 5.97 9.38 4.01 9.38 8.57 1.14 ...
##   ..- attr(*, "imputed")= int [1:6] 30 36 37 38 62 90
##  $ Payment_method: Factor w/ 4 levels "credit card",..: 3 1 3 1 3 4 1 4 1 4 ...
##  $ Total_Sales   : num  171.5 238.3 39.4 103.3 32.2 ...
```

The head function is used here to take a look at at the merged data. ‘ColSums’ function is used to count the total no of missing values in each column (Is.na function is not used here because of the number of observations) The missing values in the Discount variable is dealt by imputing NAs with the mean of discount using the ‘impute’function. However when we do this we notice that the values are filled for all observations including the one with no order number. In order to remove this, we use the ’is.na’ function on the order_no variable.

The missing values in Supplier variable is removed by deleting them as they account for only <5% of the data and will not impact any of the calculations in this study. we have used the ‘complete.cases’ function to do this.

The str function shows that after removing 5% of missing values there are now 95 observations with 12 variables

References

R Core Team (2023). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/.

Wickham H, François R, Henry L, Müller K, Vaughan D (2023). *dplyr: A Grammar of Data Manipulation.* R package version 1.1.2, https://CRAN.R-project.org/package=dplyr.

Bache S, Wickham H (2022). *magrittr: A Forward-Pipe Operator for R.* R package version 2.0.3, https://CRAN.R-project.org/package=magrittr

Damian W. Betebenner (2021). randomNames: Function for Generating Random Names and a Dataset. (R package version 1.5-0.0 URL https://cran.r-project.org/package=randomNames

Harrell Jr F (2023). *Hmisc: Harrell Miscellaneous*. R package version 5.1-1, https://CRAN.R-project.org/package=Hmisc

Group, W. (2023, sept 18). Woolworths. Retrieved from Discover - Shopping Online, Accessed on 18 September 2023, https://www.woolworths.com.au/shop/discover/shopping-online/pickup?utm_source=google&utm_medium=cpc&utm_campaign=WW-0001&cq_net=g&cq_src=GOOGLE&cq_cmp=Woolies_8458_BAU_Brand_Fulfill_WW-0001&cq_med=71700000097600800&cq_plac=&cq_term=woolworths%20online&ds_

Sona Taheri (2023) Data Wrangling [Module 4, 5 Demo and Notes & “Generating Sataset Module], RMIT University,Melbourne

Studio. (n.d.). Retrieved from Canvas, accessed on 19 September 2023: https://rmit.instructure.com/accounts/1/external_tools/38?launch_type=global_navigation