EDA on Furniture store Transaction Dataset

Aashish Telgote

2023-02-10

## Introduction

### In this project, we will do little cleaning and exploratory data analysis on “Furniture Transactions Data”. I used a small dataset in this project, but you can follow similar steps on a larger dataset and you will still get accurate results.

## ASK PHASE

### We will find answers to certain questions that will be useful for business decisions.

#### 1. What is the total revenue generated by each product?

#### 2. How many units of each product were sold?

#### 3. From which customer have we made the most revenue?

#### 4. How many products did each customer buy?

#### 5. Which color is most preferred by customers in product named “Fan”?

#### 6. Which color is most preferred by customers in product named “Couch”?

#### 7. Which color is most preferred by customers in product named “Rug”?

#### 8. Which color is most preferred by customers in product named “Desk”?

## PREPARE PHASE :

#### This is a practice dataset from **Google data analytics professional specialization course**.

#### To view the dataset, [click here](https://drive.google.com/file/d/1iLwrjh05klmLS4tDVkU97bSI3J14CQ7X/view?usp=share_link)

### Now, let’s install some required R packages to start our work.

#### We will start with tidyverse package.

#### Tidyverse is a collection of packages in R with a common design philosophy for data manipulation, exploration and visualization.

#### Usually, Tidyverse package is all we need for data analysis.

install.packages("tidyverse")

## Installing package into '/cloud/lib/x86\_64-pc-linux-gnu-library/4.2'  
## (as 'lib' is unspecified)

library("tidyverse")

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──  
## ✔ ggplot2 3.4.0 ✔ purrr 1.0.1  
## ✔ tibble 3.1.8 ✔ dplyr 1.1.0  
## ✔ tidyr 1.3.0 ✔ stringr 1.5.0  
## ✔ readr 2.1.3 ✔ forcats 1.0.0  
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library("readr")

### Let’s import our dataset in Rmarkdown. So that, we can knit it to create a final document

Store\_Transactions <- read.csv("Store\_Transactions.csv", header = TRUE, sep = ',')

### Now, we will install and load “Janitor package”. It has functions for cleaning data.

install.packages("janitor")

## Installing package into '/cloud/lib/x86\_64-pc-linux-gnu-library/4.2'  
## (as 'lib' is unspecified)

library("janitor")

##   
## Attaching package: 'janitor'

## The following objects are masked from 'package:stats':  
##   
## chisq.test, fisher.test

### Now, we will install “dplyr package” as will be using some of it’s functions.

install.packages("dplyr")

## Installing package into '/cloud/lib/x86\_64-pc-linux-gnu-library/4.2'  
## (as 'lib' is unspecified)

library("dplyr")

### Now, lets install “skimr package”. It let’s us summarize the data and skim through it quickly.

install.packages("skimr")

## Installing package into '/cloud/lib/x86\_64-pc-linux-gnu-library/4.2'  
## (as 'lib' is unspecified)

library("skimr")

### Now, let’s see the summary and basic statistics of the dataset

skim\_without\_charts(Store\_Transactions)

Data summary

|  |  |
| --- | --- |
| Name | Store\_Transactions |
| Number of rows | 29 |
| Number of columns | 10 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| character | 5 |
| numeric | 5 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | None |

**Variable type: character**

| skim\_variable | n\_missing | complete\_rate | min | max | empty | n\_unique | whitespace |
| --- | --- | --- | --- | --- | --- | --- | --- |
| date | 0 | 1 | 15 | 15 | 0 | 24 | 0 |
| product | 0 | 1 | 0 | 8 | 2 | 11 | 0 |
| product\_code | 0 | 1 | 8 | 8 | 0 | 12 | 0 |
| product\_color | 0 | 1 | 4 | 6 | 0 | 9 | 0 |
| revenue | 0 | 1 | 7 | 10 | 0 | 15 | 0 |

**Variable type: numeric**

| skim\_variable | n\_missing | complete\_rate | mean | sd | p0 | p25 | p50 | p75 | p100 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| transaction\_id | 0 | 1 | 27283.28 | 15388.50 | 1675.00 | 12560.00 | 24785.00 | 44700 | 49430 |
| customer\_id | 0 | 1 | 5456.66 | 3077.70 | 335.00 | 2512.00 | 4957.00 | 8940 | 9886 |
| product\_price | 0 | 1 | 413.39 | 429.06 | 9.99 | 58.89 | 169.95 | 1000 | 1000 |
| purchase\_size | 0 | 1 | 1.45 | 0.91 | 1.00 | 1.00 | 1.00 | 2 | 5 |
| purchase\_price | 0 | 1 | 434.64 | 414.52 | 13.99 | 89.85 | 234.50 | 1000 | 1000 |

### Let’s see the structure of the dataset and datatype of each column.

str(Store\_Transactions)

## 'data.frame': 29 obs. of 10 variables:  
## $ date : chr "29/08/2020 0:00" "01/05/2020 0:00" "12/12/2020 0:00" "16/02/2020 0:00" ...  
## $ transaction\_id: int 9900 12315 9890 46915 44700 44700 12560 9640 22620 49430 ...  
## $ customer\_id : int 1980 2463 1978 9383 8940 8940 2512 1928 4524 9886 ...  
## $ product : chr "fan" "fan" "fan" "fan" ...  
## $ product\_code : chr "SKU83503" "SKU83503" "SKU83503" "SKU83503" ...  
## $ product\_color : chr "brass" "brass" "white" "black" ...  
## $ product\_price : num 14 14 14 14 14 ...  
## $ purchase\_size : int 2 2 1 1 2 5 1 1 1 1 ...  
## $ purchase\_price: num 28 28 14 14 28 ...  
## $ revenue : chr "$27.98 " "$27.98 " "$13.99 " "$13.99 " ...

### Now, we will take a glimpse of the dataset

glimpse(Store\_Transactions)

## Rows: 29  
## Columns: 10  
## $ date <chr> "29/08/2020 0:00", "01/05/2020 0:00", "12/12/2020 0:00"…  
## $ transaction\_id <int> 9900, 12315, 9890, 46915, 44700, 44700, 12560, 9640, 22…  
## $ customer\_id <int> 1980, 2463, 1978, 9383, 8940, 8940, 2512, 1928, 4524, 9…  
## $ product <chr> "fan", "fan", "fan", "fan", "fan", "lamp", "bed", "couc…  
## $ product\_code <chr> "SKU83503", "SKU83503", "SKU83503", "SKU83503", "SKU835…  
## $ product\_color <chr> "brass", "brass", "white", "black", "brass", "brass", "…  
## $ product\_price <dbl> 13.99, 13.99, 13.99, 13.99, 13.99, 45.99, 799.99, 1000.…  
## $ purchase\_size <int> 2, 2, 1, 1, 2, 5, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 2, 1…  
## $ purchase\_price <dbl> 27.980, 27.980, 13.990, 13.990, 27.980, 160.965, 799.99…  
## $ revenue <chr> "$27.98 ", "$27.98 ", "$13.99 ", "$13.99 ", "$27.98 ", …

### Now, if we want we can only check all the column names

colnames(Store\_Transactions)

## [1] "date" "transaction\_id" "customer\_id" "product"   
## [5] "product\_code" "product\_color" "product\_price" "purchase\_size"   
## [9] "purchase\_price" "revenue"

### Let’s preview the dataset to know how it looks in tabular format.

head(Store\_Transactions)

## date transaction\_id customer\_id product product\_code product\_color  
## 1 29/08/2020 0:00 9900 1980 fan SKU83503 brass  
## 2 01/05/2020 0:00 12315 2463 fan SKU83503 brass  
## 3 12/12/2020 0:00 9890 1978 fan SKU83503 white  
## 4 16/02/2020 0:00 46915 9383 fan SKU83503 black  
## 5 28/12/2020 0:00 44700 8940 fan SKU83503 brass  
## 6 28/12/2020 0:00 44700 8940 lamp SKU95363 brass  
## product\_price purchase\_size purchase\_price revenue  
## 1 13.99 2 27.980 $27.98   
## 2 13.99 2 27.980 $27.98   
## 3 13.99 1 13.990 $13.99   
## 4 13.99 1 13.990 $13.99   
## 5 13.99 2 27.980 $27.98   
## 6 45.99 5 160.965 $229.95

## 

## PROCESS PHASE

### In this phase, we will do some data cleaning.

#### Let’s rename the “product” and “purchase size” column to Product\_name and Units\_purchased respectively for better understanding of underlying data in the column.

Store\_Transactions <- Store\_Transactions %>%  
 rename(product\_name=product) %>%  
 rename(units\_purchased=purchase\_size)

#### To highlight column names more clearly. Let’s capitalize column names

Store\_Transactions <- rename\_with(Store\_Transactions, toupper)

#### Let’s preview to see if the changes occured

head(Store\_Transactions)

## DATE TRANSACTION\_ID CUSTOMER\_ID PRODUCT\_NAME PRODUCT\_CODE  
## 1 29/08/2020 0:00 9900 1980 fan SKU83503  
## 2 01/05/2020 0:00 12315 2463 fan SKU83503  
## 3 12/12/2020 0:00 9890 1978 fan SKU83503  
## 4 16/02/2020 0:00 46915 9383 fan SKU83503  
## 5 28/12/2020 0:00 44700 8940 fan SKU83503  
## 6 28/12/2020 0:00 44700 8940 lamp SKU95363  
## PRODUCT\_COLOR PRODUCT\_PRICE UNITS\_PURCHASED PURCHASE\_PRICE REVENUE  
## 1 brass 13.99 2 27.980 $27.98   
## 2 brass 13.99 2 27.980 $27.98   
## 3 white 13.99 1 13.990 $13.99   
## 4 black 13.99 1 13.990 $13.99   
## 5 brass 13.99 2 27.980 $27.98   
## 6 brass 45.99 5 160.965 $229.95

#### Let’s load another package to make changes related to date

library("lubridate")

##   
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':  
##   
## date, intersect, setdiff, union

#### let’s see the format type of “date” column

class(Store\_Transactions$DATE)

## [1] "character"

#### Thus, to be able to perform operations on the date lets’convert **date from char to date**

Store\_Transactions$DATE <- ymd(Store\_Transactions$DATE)

## Warning: All formats failed to parse. No formats found.

#### Now, lets see if the change has occured

class(Store\_Transactions$DATE)

## [1] "Date"

#### Now, we will remove all rows with N.A values in columns. Otherwise, they would cause problem while analysing data.

#### We will save the results in new table, as Store\_Transaction

Store\_Transaction <- Store\_Transactions[!is.na(Store\_Transactions$PRODUCT\_NAME), ]

#### OR In the code, we can also mention particular rows we want to remove.

Store\_Transaction <- Store\_Transactions[-c(28,29),]

#### Let’s create another column “NEW\_REVENUE” to calculate revenue of each transaction and cross check it with column named “PURCHASE\_PRICE”

Store\_Transaction <- Store\_Transaction %>% mutate(Store\_Transaction, NEW\_REVENUE= PRODUCT\_PRICE\*UNITS\_PURCHASED)

#### Now, we will remove all columns that we don’t require for our analysis.

#### We will also be removing “purchase price” column as we have newly created accurate column named “new\_revenue” in place of it

Store\_Transaction <- Store\_Transaction %>% select(-DATE,-PRODUCT\_CODE,-PURCHASE\_PRICE)

#### Now, let’s check again if the changes we made occured or not

head(Store\_Transaction)

## TRANSACTION\_ID CUSTOMER\_ID PRODUCT\_NAME PRODUCT\_COLOR PRODUCT\_PRICE  
## 1 9900 1980 fan brass 13.99  
## 2 12315 2463 fan brass 13.99  
## 3 9890 1978 fan white 13.99  
## 4 46915 9383 fan black 13.99  
## 5 44700 8940 fan brass 13.99  
## 6 44700 8940 lamp brass 45.99  
## UNITS\_PURCHASED REVENUE NEW\_REVENUE  
## 1 2 $27.98 27.98  
## 2 2 $27.98 27.98  
## 3 1 $13.99 13.99  
## 4 1 $13.99 13.99  
## 5 2 $27.98 27.98  
## 6 5 $229.95 229.95

## ANALYSIS PHASE

### It’s time for us to analyse the data and find what insights we can get from it.

#### Every transformation we will make in orignal dataset to pull out insights, we will be saving those transformations in new tables in order to make visuals from them later.

#### First, we will find out how much revenue each product generated

# Grouping and summarizing in order to find total Revenue generated from each product  
Products\_vs\_Revenue <- Store\_Transaction %>% group\_by(PRODUCT\_NAME) %>%  
 summarize(Total\_revenue\_of\_each\_product = sum(NEW\_REVENUE))   
head(Products\_vs\_Revenue)

## # A tibble: 6 × 2  
## PRODUCT\_NAME Total\_revenue\_of\_each\_product  
## <chr> <dbl>  
## 1 bed 800.   
## 2 bookcase 58.9  
## 3 chair 234.   
## 4 couch 9000   
## 5 desk 510.   
## 6 fan 112.

#### Now, we will see how many units of each product were sold.

# Grouping and summarizing in order to find how many units of each product were sold.  
Products\_vs\_units <- Store\_Transaction %>% group\_by(PRODUCT\_NAME) %>%  
 summarize(Total\_units\_sold\_of\_each\_product = sum(UNITS\_PURCHASED))  
head(Products\_vs\_units)

## # A tibble: 6 × 2  
## PRODUCT\_NAME Total\_units\_sold\_of\_each\_product  
## <chr> <int>  
## 1 bed 1  
## 2 bookcase 1  
## 3 chair 1  
## 4 couch 9  
## 5 desk 3  
## 6 fan 8

#### Now, let’s see the revenue generated from each customer

# Grouping and summarizing in order to find total revenue generated from each customer  
Customer\_vs\_revenue <- Store\_Transaction %>% group\_by(CUSTOMER\_ID) %>%  
 summarize(Total\_revenue\_by\_each\_customer = sum(NEW\_REVENUE))  
head(Customer\_vs\_revenue)

## # A tibble: 6 × 2  
## CUSTOMER\_ID Total\_revenue\_by\_each\_customer  
## <int> <dbl>  
## 1 335 1000   
## 2 1268 170.   
## 3 1928 1000   
## 4 1978 14.0  
## 5 1980 1028.   
## 6 2463 28.0

#### Now, we will see number of units bought by each customer.

# Grouping and summarizing in order to find total units bought by each customer   
Customer\_vs\_units\_purchased <- Store\_Transaction %>% group\_by(CUSTOMER\_ID) %>%  
 summarize(Total\_units\_bought\_by\_each\_customer = sum(UNITS\_PURCHASED))  
head(Customer\_vs\_units\_purchased)

## # A tibble: 6 × 2  
## CUSTOMER\_ID Total\_units\_bought\_by\_each\_customer  
## <int> <int>  
## 1 335 1  
## 2 1268 1  
## 3 1928 1  
## 4 1978 1  
## 5 1980 3  
## 6 2463 2

#### Now, we will analyse revenue from individual products which are available with different colours.

#### First, let’s see which colour of product “Fan” made the most revenue

# Filtering to pull out products named "FAN"  
PRODUCT\_FAN <- Store\_Transaction %>% filter(PRODUCT\_NAME=='fan')  
# Creating a new column by uniting 2 columns.  
PRODUCT\_FAN <- unite(PRODUCT\_FAN,'PRODUCT\_NAME\_and\_COLOR', PRODUCT\_NAME,PRODUCT\_COLOR, sep = ' ')  
# Grouping and summarizing in order to find revenue of product generated by each of its colour variations  
PRODUCT\_FAN <- PRODUCT\_FAN %>% group\_by(PRODUCT\_NAME\_and\_COLOR) %>%  
 summarize(Total\_revenue\_by\_each\_color = sum(NEW\_REVENUE))

head(PRODUCT\_FAN)

## # A tibble: 3 × 2  
## PRODUCT\_NAME\_and\_COLOR Total\_revenue\_by\_each\_color  
## <chr> <dbl>  
## 1 fan black 14.0  
## 2 fan brass 83.9  
## 3 fan white 14.0

#### Now, let’s see which colour of product “Couch” made the most revenue

# Filtering to pull out products named "COUCH"  
PRODUCT\_COUCH <- Store\_Transaction %>% filter(PRODUCT\_NAME=='couch')  
# Creating a new column by uniting 2 columns.  
PRODUCT\_COUCH <- unite(PRODUCT\_COUCH,'PRODUCT\_NAME\_and\_COLOR', PRODUCT\_NAME,PRODUCT\_COLOR, sep = ' ')  
# Grouping and summarizing in order to find revenue of product generated by each of its colour variations  
PRODUCT\_COUCH <- PRODUCT\_COUCH %>% group\_by(PRODUCT\_NAME\_and\_COLOR) %>%  
 summarize(Total\_revenue\_by\_each\_color = sum(NEW\_REVENUE))

head(PRODUCT\_COUCH)

## # A tibble: 6 × 2  
## PRODUCT\_NAME\_and\_COLOR Total\_revenue\_by\_each\_color  
## <chr> <dbl>  
## 1 couch black 1000  
## 2 couch blue 1000  
## 3 couch brown 1000  
## 4 couch grey 3000  
## 5 couch purple 1000  
## 6 couch white 2000

#### Now, let’s see which colour of product “Rug” made the most revenue

# Filtering to pull out products named "RUG"  
PRODUCT\_RUG <- Store\_Transaction %>% filter(PRODUCT\_NAME=='rug')  
# Creating a new column by uniting 2 columns.  
PRODUCT\_RUG <- unite(PRODUCT\_RUG,'PRODUCT\_NAME\_and\_COLOR', PRODUCT\_NAME,PRODUCT\_COLOR, sep = ' ')  
# Grouping and summarizing in order to find revenue of product generated by each of its colour variations  
PRODUCT\_RUG <- PRODUCT\_RUG %>% group\_by(PRODUCT\_NAME\_and\_COLOR) %>%  
 summarize(Total\_revenue\_by\_each\_color = sum(NEW\_REVENUE))

head(PRODUCT\_RUG)

## # A tibble: 2 × 2  
## PRODUCT\_NAME\_and\_COLOR Total\_revenue\_by\_each\_color  
## <chr> <dbl>  
## 1 rug beige 539.  
## 2 rug grey 270.

#### Now, let’s see which colour of product “Desk” made the most revenue

# Filtering to pull out products named "DESK"  
PRODUCT\_DESK <- Store\_Transaction %>% filter(PRODUCT\_NAME=='desk')  
# Creating a new column by uniting 2 columns.  
PRODUCT\_DESK <- unite(PRODUCT\_DESK,'PRODUCT\_NAME\_and\_COLOR', PRODUCT\_NAME,PRODUCT\_COLOR, sep = ' ')  
# Grouping and summarizing in order to find revenue of product generated by each of its colour variations  
PRODUCT\_DESK <- PRODUCT\_DESK %>% group\_by(PRODUCT\_NAME\_and\_COLOR) %>%  
 summarize(Total\_revenue\_by\_each\_color = sum(NEW\_REVENUE))

head(PRODUCT\_DESK)

## # A tibble: 2 × 2  
## PRODUCT\_NAME\_and\_COLOR Total\_revenue\_by\_each\_color  
## <chr> <dbl>  
## 1 desk brown 340.  
## 2 desk white 170.

## SHARE PHASE

#### In this phase, we will present the insights we found from our analysis by using visualisations.

Note :*I will be sharing the code for how to create visuals in Rstudio. But, because they were difficult to understand for stakeholder’s, I will be sharing the visuals that I created using Google sheets. They provide a accurate, detailed understanding of the insights we pulled from data.*

### 1. What is the total revenue generated by each product?

# ggplot(data = Products\_vs\_Revenue) +   
# geom\_bar(mapping =aes(x=Total\_revenue\_of\_each\_product, fill=PRODUCT\_NAME))

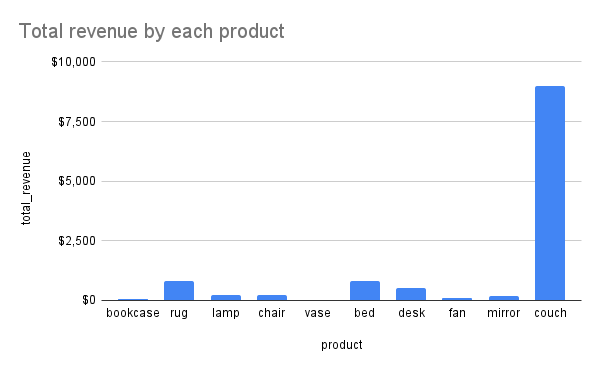


Fig.a

#### It’s surprising to see that the product “couch” generated the most revenue for our store as compared to other products. The revenue is literally around 9000 $, while we couldn’t even generate minimum 2500 $ for any of the other products. This possibly has multiple reasons such as, we sell couches with the most variety in colors. So, customers prefer to buy couch from our store as there are many varieties available with respect to color. Another reason we made most revenue from “couch” is because it’s also the most expensive product in our furniture shop, each one costing 1000$.

### 2. How many units of each product were sold?

# ggplot(data = Products\_vs\_units) +  
# geom\_bar(mapping = aes(x=PRODUCT\_NAME, fill=Total\_units\_sold\_of\_each\_product))

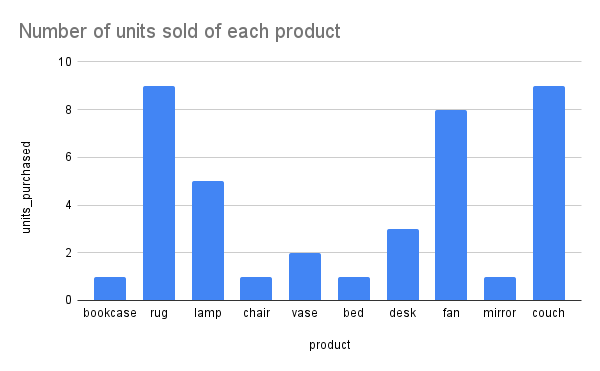


Fig.b

#### It’s clear from the above figure that the total units sold of products “FAN, RUG and COUCH” are highest compared to other products. The number of units sold of this products were minimum 8. This states that most customers are in need of FAN, RUG & COUCH than other products.

### 

### 3. From which customer have we made the most revenue?

# ggplot(data = Customer\_vs\_revenue) +  
# geom\_bar(mapping = aes(x=CUSTOMER\_ID, fill=Total\_revenue\_by\_each\_customer))

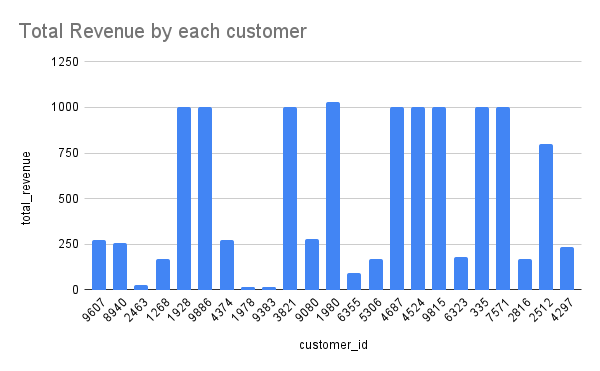


Fig.c

#### Looking at this graph and looking back to our earlier findings, we can say that those customers who bought “couches” from our store generated the most revenue for us and this graph indirectly suggests the same.

### 4. How many products did each customer buy?

# ggplot(data = Customer\_vs\_units\_purchased) +  
# geom\_bar(mapping = aes(x=Total\_units\_bought\_by\_each\_customer , fill=PRODUCT\_NAME))

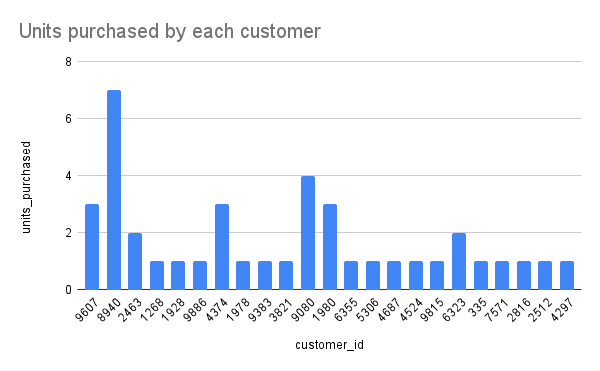


Fig.d

#### The customer with ID 8940 purchased the highest number of furniture products from our store. And the customer who bought 2nd highest number of products from our store has customer ID9080.

#### Then there are three customers who bought approximately 3 products from our store and some other two customers bought approximately 2 products from our store. Remaining customers have only bought 1 product from our store.

#### We can conclude that the top 2 customers who bought most products from our store are

**• ID8940** **• ID9080**

### 5. Which color is most preferred by customers in product named “Fan”?

# ggplot(data = PRODUCT\_FAN) +  
# geom\_bar(mapping = aes(x=Total\_revenue\_by\_each\_color, fill=PRODUCT\_NAME\_and\_COLOR))

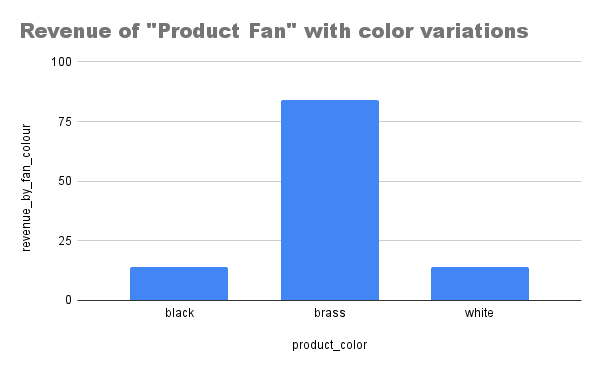


Fig.e

#### As we can see, the brass colour of product “FAN” is more preferred by customers and thus has generated revenue of above 75 $ for our Store. While the white & black colour of it generated comparatively less revenue which is under 25$.

#### It’s good to remember that all colour variants of this product are sold at the same price. But, because the ‘brass’ colour variant was sold more. Thus, it generated more revenue for our store.

### 6. Which color is most preferred by customers in product named “Couch”?

# ggplot(data = PRODUCT\_COUCH) +  
# geom\_bar(mapping = aes(x=Total\_revenue\_by\_each\_color, fill=PRODUCT\_NAME\_and\_COLOR))

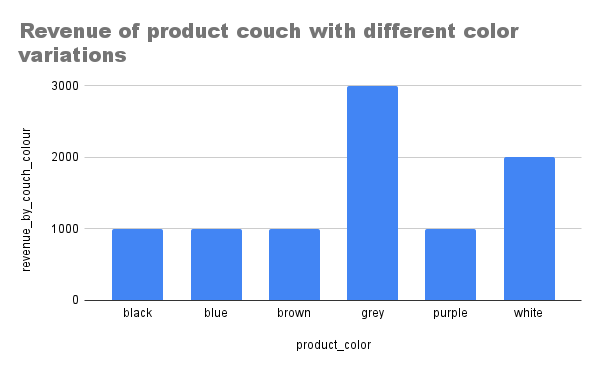


Fig.f

#### As we can see, the Grey colour of product “COUCH” is more preferred by customers and thus has generated revenue of around 3000 $ for our Store. While the white colour of it made comparatively less which is around 2000$.

#### The other remaining 4 variants generated around 1000$ each for our store.

#### It’s good to remember that all colour variants of this product are sold at the same price. But, because the ‘Grey’ and ‘White’ colour variant were sold more. Thus, they generated more revenue for our store.

### 7. Which color is most preferred by customers in product named “Rug”?

# ggplot(data = PRODUCT\_RUG) +  
# geom\_bar(mapping = aes(x=Total\_revenue\_by\_each\_color, fill=PRODUCT\_NAME\_and\_COLOR))

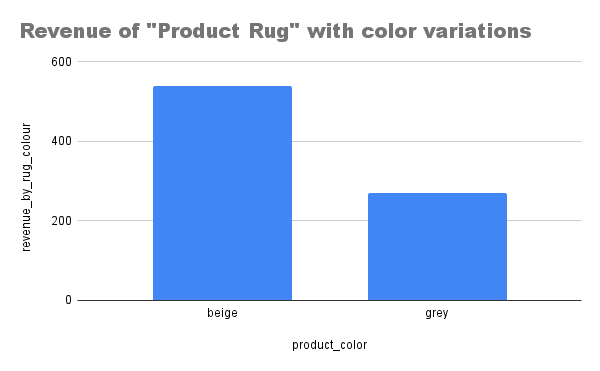


Fig.g

#### As we can see, the beige colour of product “RUG” is more preferred by customers and thus has generated revenue of above 500 $ for our Store. While the grey colour of it generated comparatively less revenue which is around 300$.

#### It’s good to remember that all colour variants of this product are sold at the same price. But, because the ‘beige’ colour variant was sold more. Thus, it generated more revenue for our store.

### 8.Which color is most preferred by customers in product named “Desk”?

# ggplot(data = PRODUCT\_DESK) +  
# geom\_bar(mapping = aes(x=Total\_revenue\_by\_each\_color, fill=PRODUCT\_NAME\_and\_COLOR))

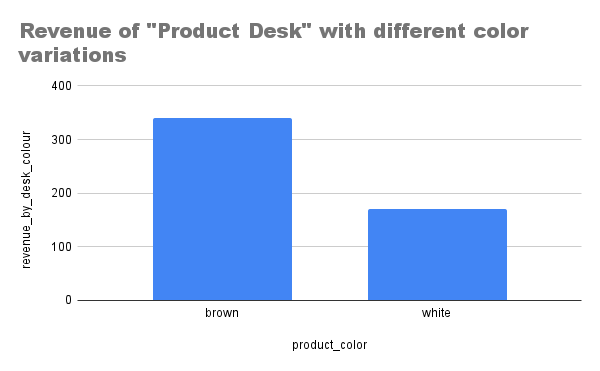


Fig.h

#### As we can see, the brown colour of product “DESK” is more preferred by customers and thus has generated revenue of above 300 $ for our Store. While the white colour of it generated comparatively less which around 150$.

#### It’s good to remember that all colour variants of this product are sold at the same price. But, because the ‘brown’ colour variant was sold more. Thus, it generated more revenue for our store.

## Recommendations :

#### 1. FAN, RUG, COUCH are the most in demand product, so we should ensure that there’s sufficient stock of this products in our inventory.

#### 2. We have 2 most loyal customers, who generally buy from our store. So, from time to time we should see if they are in need of any furniture and provide them with best offers for being a loyal customer to our shop. This will also encourage other customers to fulfill most of their furniture needs from our store.

#### 3. We should keep more variants of every single product, as people want to choose from a range of varieties. Also, we should try to keep those furniture products that are generally expensive, as they will generate the most revenue or profit for us.

#### 4. Currently, product “Couch” is generating the most revenue for us. So, it’s important to ensure that couch sales continue like this by running the business operations for product “couch” without any change for now.

#### 5. As seen earlier in products that have different color varieties. Certain color of each of this product get purchased more than others. So, we should maintain their stocks in our inventory as they are more preferred color variants.

#### In short, they are.

#### • For “COUCH” preferred colours are grey and white.

#### • For “RUG” preferred colour is beige.

#### • For “FAN” preferred colour is brass.

#### • For “DESK” preferred colour is brown.