Lab_2_Financial_Analysis

Hunter Davis

2024-11-26

Financial Analysis and Decision Making with R

Step One: Setting up and Preparing Data

Initialized my work space with necessary packages.

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggplot2)
library(readr)
library(dplyr)
```

Imported the Sales data CSV into RStudio.

```
# Import data set: financial_data.csv
financial_data <- read_csv("~/AMU/Dats200/Week_3/financial_data.csv")
```

```
## Rows: 200 Columns: 7
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (2): ProductName, Category
## dbl  (4): ProductID, UnitPrice, UnitsSold, TotalSales
## date (1): Date
##
## 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.
```

```
financial_data
```

```
## # A tibble: 200 x 7
##    Date       ProductID ProductName Category    UnitPrice UnitsSold TotalSales
##    <date>         <dbl> <chr>       <chr>           <dbl>     <dbl>      <dbl>
##  1 2023-01-17       161 Product_9   Furniture        46.9        19       891.
##  2 2023-01-05       113 Product_3   Clothing         97.3        37      3600.
##  3 2023-01-01       133 Product_7   Groceries        42.8         3       128.
##  4 2023-01-25       165 Product_3   Electronics      75.2        45      3386.
##  5 2023-01-10       194 Product_6   Clothing         73.2        19      1392.
##  6 2023-01-04       131 Product_4   Toys             12.2        50       612.
##  7 2023-01-18       126 Product_10  Clothing         95.1        20      1903.
##  8 2023-01-26       109 Product_2   Furniture        58.0        31      1798.
##  9 2023-01-17       156 Product_7   Groceries        72.7        14      1017.
## 10 2023-01-15       127 Product_1   Furniture        14.1        34       478.
## # i 190 more rows
```

Explored the data.

```
# Check header names for each column
colnames(financial_data)
```

```
## [1] "Date"        "ProductID"   "ProductName" "Category"    "UnitPrice"
## [6] "UnitsSold"   "TotalSales"
```

```
# Look at the top/bottom six rows of the data
head(financial_data)
```

```
## # A tibble: 6 x 7
##   Date       ProductID ProductName Category    UnitPrice UnitsSold TotalSales
##   <date>         <dbl> <chr>       <chr>           <dbl>     <dbl>      <dbl>
## 1 2023-01-17       161 Product_9   Furniture        46.9        19       891.
## 2 2023-01-05       113 Product_3   Clothing         97.3        37      3600.
## 3 2023-01-01       133 Product_7   Groceries        42.8         3       128.
## 4 2023-01-25       165 Product_3   Electronics      75.2        45      3386.
## 5 2023-01-10       194 Product_6   Clothing         73.2        19      1392.
## 6 2023-01-04       131 Product_4   Toys             12.2        50       612.
```

```
tail(financial_data)
```

```
## # A tibble: 6 x 7
##   Date       ProductID ProductName Category    UnitPrice UnitsSold TotalSales
##   <date>         <dbl> <chr>       <chr>           <dbl>     <dbl>      <dbl>
## 1 2023-01-24       109 Product_5   Clothing         72.7        18      1308.
## 2 2023-01-02       170 Product_2   Electronics      70.6        43      3037.
## 3 2023-01-25       191 Product_5   Groceries        26.2         9       236.
## 4 2023-01-28       181 Product_5   Furniture        75.9         8       607.
## 5 2023-01-11       124 Product_10  Toys             39.4        41      1615.
## 6 2023-01-29       173 Product_1   Electronics      41.7         8       334.
```

```r
# Examine the dimensions of the data
dim(financial_data)
```

```
## [1] 200   7
```

```r
# More in depth look at the data
summary(financial_data)
```

```
##       Date              ProductID     ProductName          Category
##  Min.   :2023-01-01   Min.   :100.0   Length:200         Length:200
##  1st Qu.:2023-01-07   1st Qu.:126.0   Class :character   Class :character
##  Median :2023-01-16   Median :150.0   Mode  :character   Mode  :character
##  Mean   :2023-01-15   Mean   :149.6
##  3rd Qu.:2023-01-24   3rd Qu.:178.0
##  Max.   :2023-01-31   Max.   :200.0
##    UnitPrice       UnitsSold       TotalSales
##  Min.   :10.51   Min.   : 1.00   Min.   :  21.02
##  1st Qu.:36.61   1st Qu.:14.00   1st Qu.: 602.34
##  Median :57.77   Median :25.50   Median :1242.52
##  Mean   :56.42   Mean   :25.73   Mean   :1431.04
##  3rd Qu.:78.47   3rd Qu.:37.00   3rd Qu.:2021.09
##  Max.   :99.52   Max.   :50.00   Max.   :4261.50
```

```r
# Shows us how many of each object in the variable and Sort the table
sort(table(financial_data$Category), decreasing = TRUE)
```

```
##
## Electronics    Clothing   Furniture   Groceries        Toys
##          47          43          43          42          25
```

Data Cleaning

I checked for any Na's in the columns.

```r
# Looking for Na's
colSums(is.na(financial_data))
```

```
##        Date   ProductID ProductName    Category   UnitPrice   UnitsSold
##           0           0           0           0           0           0
##  TotalSales
##           0
```

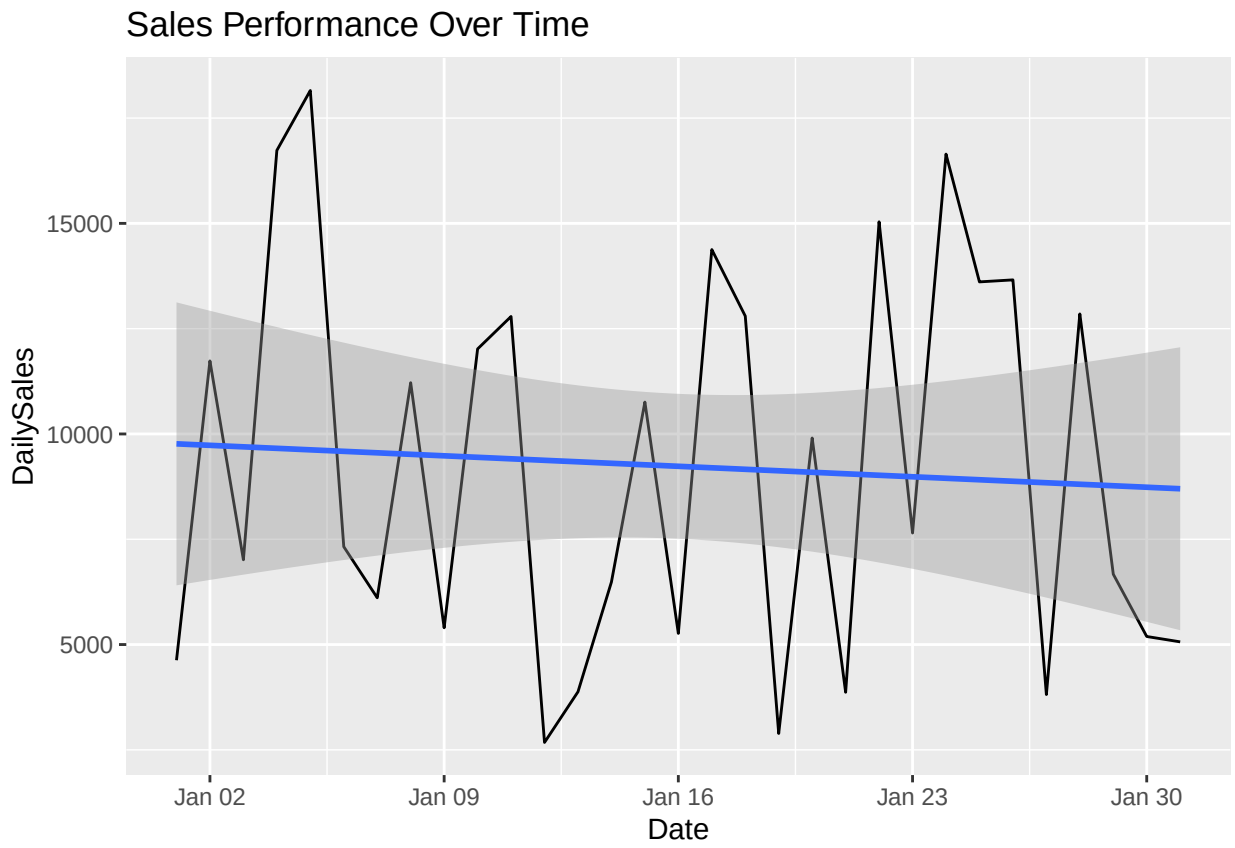
Step Two: Data Analysis

Sales performance analysis and identifying high and low-demand days.

```r
#Sale Performance

# Visualize Sales performance over time
financial_data %>%
  group_by(Date) %>%
  summarize(DailySales = sum(TotalSales)) %>%
  ggplot(aes(x = Date, y = DailySales)) +
  geom_line() +
  geom_smooth(method = "lm")+
  labs(title = "Sales Performance Over Time")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



The chart above shows us the daily sales over the month.

```r
# Product analysis
# determine the top-performing and the
# under performing products

# %>% short cut: shift+Ctrl+m
Top_Products <- financial_data %>%
  group_by(ProductName) %>%
  summarize(TotalSales = sum(TotalSales),
            AvgUnitPrice = round(mean(UnitPrice), 2),
            TotalUnitsSold = sum(UnitsSold)) %>%
  arrange(desc(TotalSales))
```

```
Top_Products
```

```
## # A tibble: 10 x 4
##    ProductName TotalSales AvgUnitPrice TotalUnitsSold
##    <chr>            <dbl>        <dbl>          <dbl>
##  1 Product_10      53029.         59.2            993
##  2 Product_3       43446.         61.5            687
##  3 Product_7       32118.         50.3            635
##  4 Product_6       27090.         61.2            443
##  5 Product_5       25585.         59.0            466
##  6 Product_4       24099.         52.8            452
##  7 Product_2       23458.         52.0            460
##  8 Product_9       23329.         55.4            393
##  9 Product_8       19846.         62.8            302
## 10 Product_1       14209.         45.7            316
```

The table above sorts the most profitable product to the least.

```
# Analyze by category
Top_Category <- financial_data %>%
  group_by(Category) %>%
  summarize(TotalSales = sum(TotalSales),
            AvgUnitPrice = round(mean(UnitPrice), 2),
            TotalUnitsSold = sum(UnitsSold)) %>%
  arrange(desc(TotalSales))

Top_Category
```

```
## # A tibble: 5 x 4
##   Category    TotalSales AvgUnitPrice TotalUnitsSold
##   <chr>            <dbl>        <dbl>          <dbl>
## 1 Furniture       69732.         53.0           1382
## 2 Electronics     69276.         64.6           1075
## 3 Clothing        64382.         56.1           1033
## 4 Groceries       49434.         55.4            935
## 5 Toys            33385.         49.2            722
```

Same analysis but with category of product.

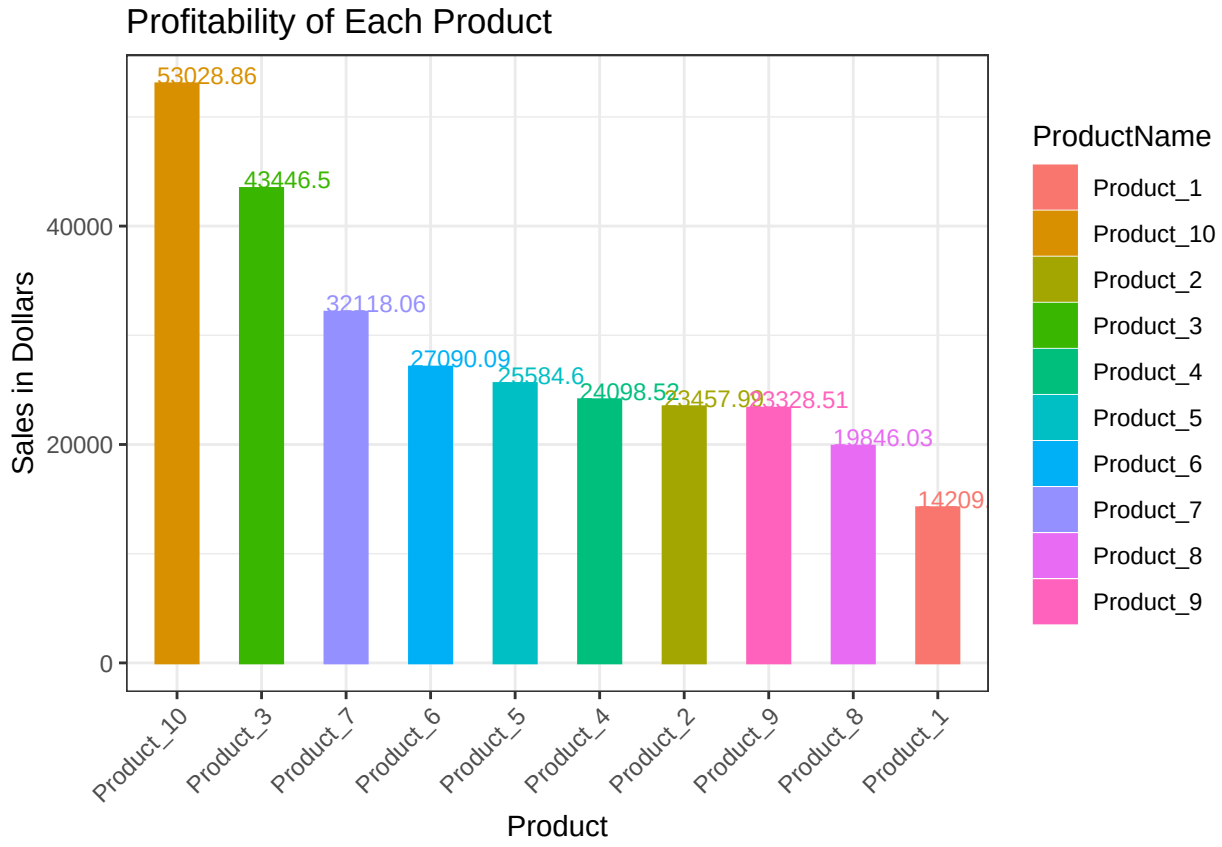
```
# Profitability Analysis
# Calculate the profitability of the products
# consider sales volume, unit price, and other relevant costs

# Create a plot and assign it to a variable
# This way we can use the theme function to manipulate it
Product_Graph <- ggplot(data = Top_Products ,
             mapping = aes(x = reorder(ProductName, -TotalSales),
                           y = TotalSales,
                           color = ProductName))+
        geom_bar(aes(fill = ProductName),
                 stat = "identity",
```

```
                width = 0.5)+
        geom_text(aes(label = TotalSales),
                  angle = 0,
                  vjust = 0,
                  hjust = 0.2,
                  size = 3)+
        labs(title = "Profitability of Each Product",
             x = "Product",
             y = "Sales in Dollars")+
        theme_bw()

# Manipulation with the theme function
# and update variable with changes
Product_Graph <- Product_Graph +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1))

Product_Graph
```



A graph to show exactly how much each product earned in profit during the moth of Janurary.

```
# calculations to find the Gross profit margin
Total_sales <- sum(Top_Products$TotalSales)

Sales_by_Product <- vector()

Sales_by_Product <- Top_Products$TotalSales
```

```r
Gross_Profit_by_Product <- vector()

Gross_Profit_by_Product <- data.frame(round(Sales_by_Product/Total_sales, 4))

Gross_Profit_by_Product
```

```
##    round.Sales_by_Product.Total_sales..4.
## 1                                  0.1853
## 2                                  0.1518
## 3                                  0.1122
## 4                                  0.0947
## 5                                  0.0894
## 6                                  0.0842
## 7                                  0.0820
## 8                                  0.0815
## 9                                  0.0693
## 10                                 0.0496
```

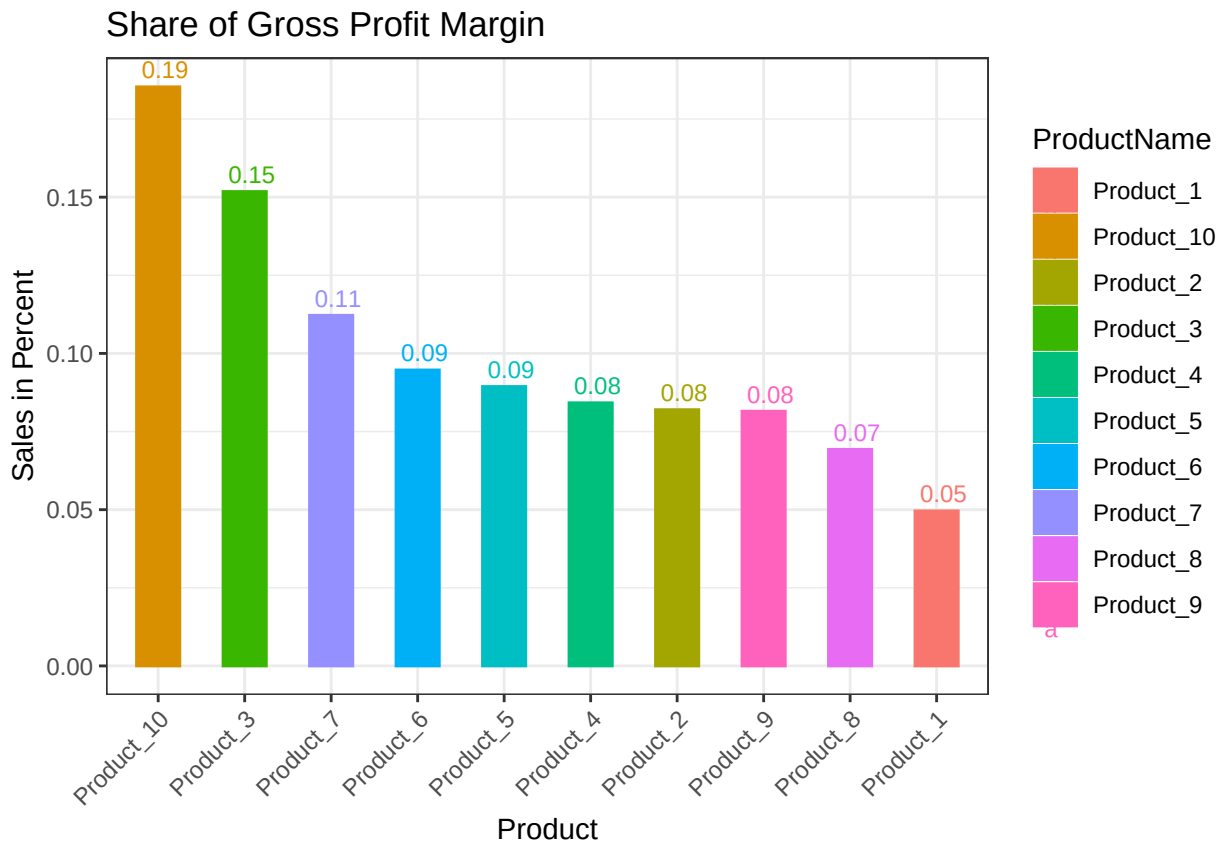
```r
Top_Products <- cbind(Top_Products, Gross_Profit_by_Product)
colnames(Top_Products)[5] = "GPM"
Top_Products
```

```
##    ProductName TotalSales AvgUnitPrice TotalUnitsSold    GPM
## 1   Product_10   53028.86        59.20            993 0.1853
## 2    Product_3   43446.50        61.50            687 0.1518
## 3    Product_7   32118.06        50.26            635 0.1122
## 4    Product_6   27090.09        61.16            443 0.0947
## 5    Product_5   25584.60        58.97            466 0.0894
## 6    Product_4   24098.52        52.78            452 0.0842
## 7    Product_2   23457.99        52.02            460 0.0820
## 8    Product_9   23328.51        55.35            393 0.0815
## 9    Product_8   19846.03        62.81            302 0.0693
## 10   Product_1   14209.48        45.73            316 0.0496
```

```r
# Graph of gross profit margin
Percent_Graph <- ggplot(data = Top_Products ,
                        mapping = aes(x = reorder(ProductName, -GPM),
                                      y = GPM,
                                      color = ProductName))+
  geom_bar(aes(fill = ProductName),
           stat = "identity",
           width = 0.5)+
  geom_text(aes(label = round(GPM, 2)),
            angle = 0,
            vjust = -0.5,
            hjust = .35,
            size = 3)+
  labs(title = "Share of Gross Profit Margin",
       x = "Product",
       y = "Sales in Percent")+
  theme_bw()
```

```
Percent_Graph <- Percent_Graph +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1))

Percent_Graph
```

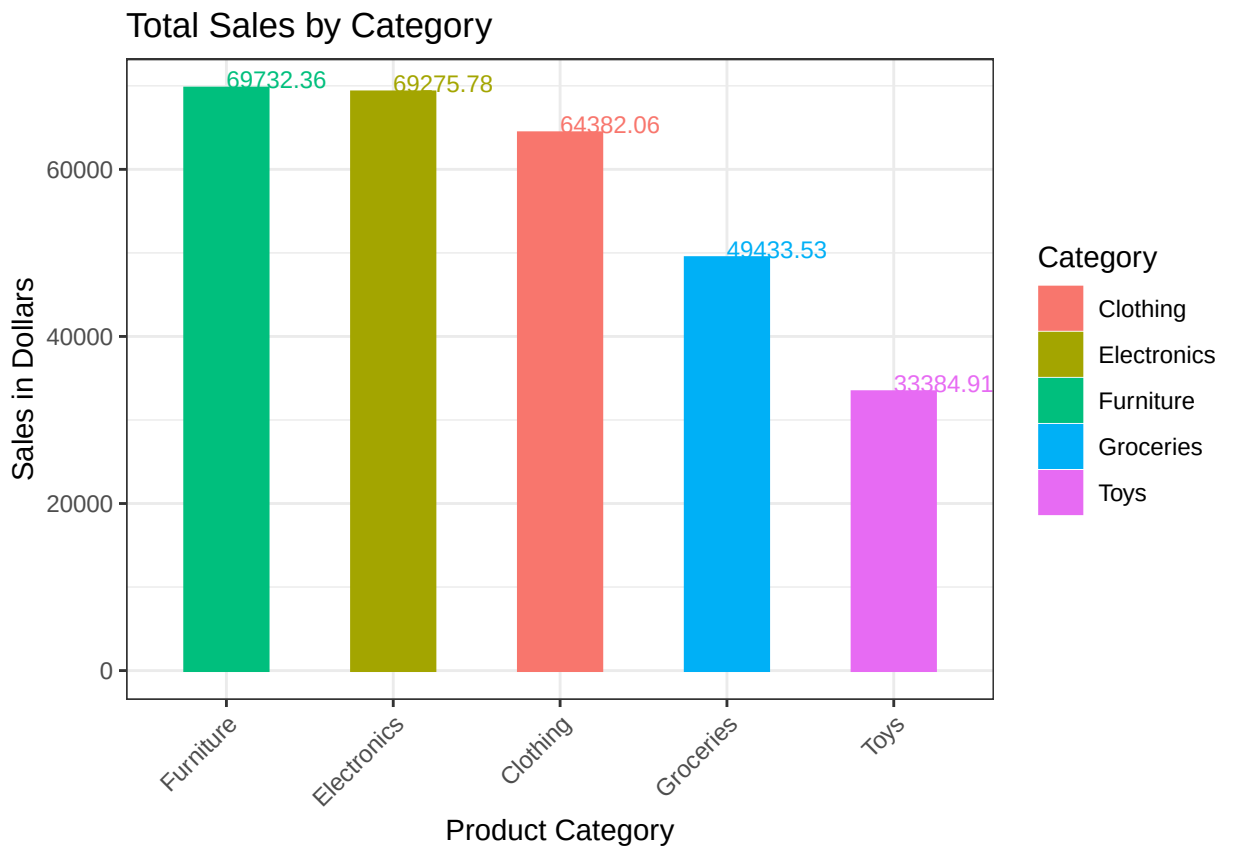


Shows the percentage of gross profits by product type.

```
Category_Graph <- ggplot(data = Top_Category ,
                         mapping = aes(x = reorder(Category, -TotalSales),
                                       y = TotalSales,
                                       color = Category))+
  geom_bar(aes(fill = Category),
           stat = "identity",
           width = 0.5)+
  geom_text(aes(label = TotalSales),
            angle = 0,
            vjust = 0,
            hjust = 0,
            size = 3)+
  labs(title = "Total Sales by Category",
       x = "Product Category",
       y = "Sales in Dollars")+
  theme_bw()

Category_Graph <- Category_Graph +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1))
```

```
Category_Graph
```



Shows the sales data on each category during the month.

```
# Percent graph by category
Sales_by_Category <- vector()

Sales_by_Category <- Top_Category$TotalSales

Gross_Profit_by_Category <- vector()

Gross_Profit_by_Category <- data.frame(round(Sales_by_Category/Total_sales, 4))
Gross_Profit_by_Category
```

```
##   round.Sales_by_Category.Total_sales..4.
## 1                                  0.2436
## 2                                  0.2420
## 3                                  0.2249
## 4                                  0.1727
## 5                                  0.1166
```

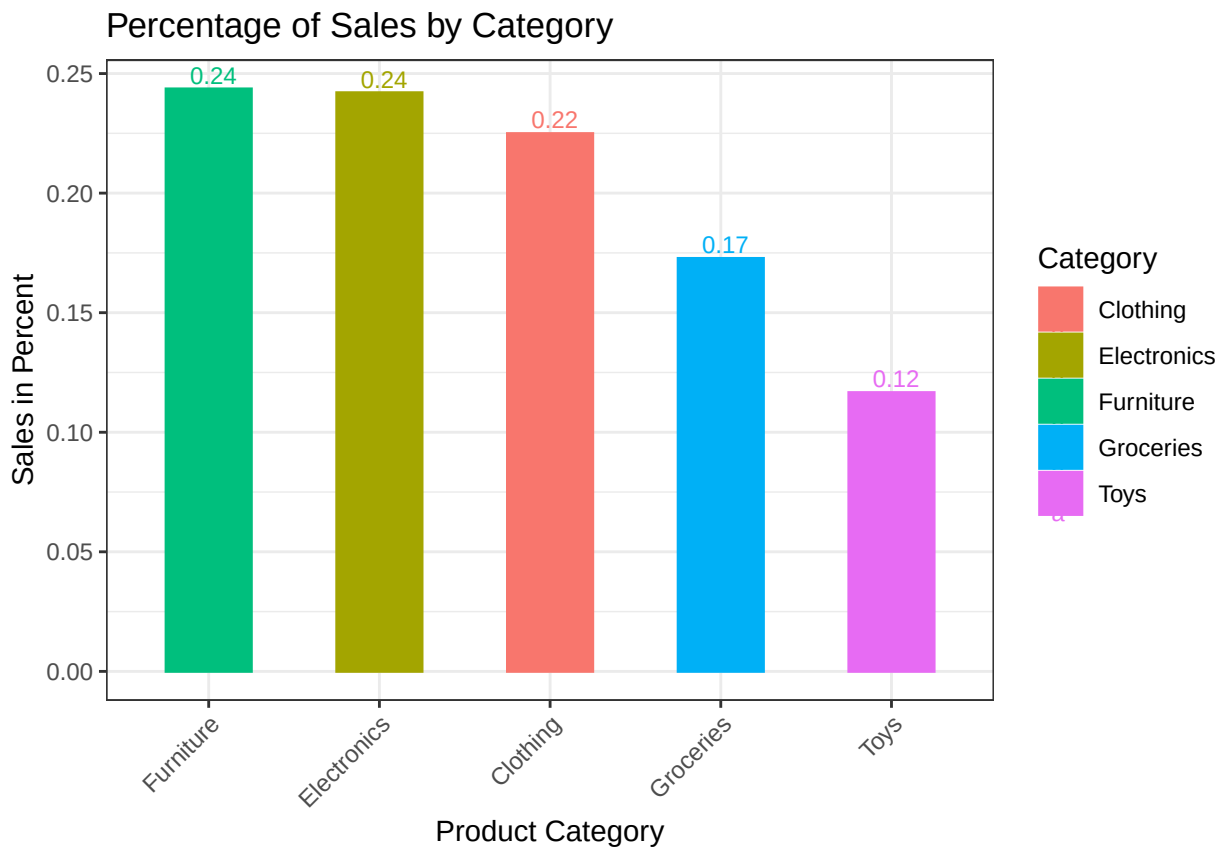
```
Top_Category <- cbind(Top_Category, Gross_Profit_by_Category)
colnames(Top_Category)[5] = "Percent_of_Sales"
Top_Category
```

```
##      Category TotalSales AvgUnitPrice TotalUnitsSold Percent_of_Sales
## 1   Furniture   69732.36        52.97           1382           0.2436
## 2 Electronics   69275.78        64.61           1075           0.2420
## 3    Clothing   64382.06        56.07           1033           0.2249
## 4   Groceries   49433.53        55.44            935           0.1727
## 5        Toys   33384.91        49.15            722           0.1166
```

```r
Category_Percent_Graph <- ggplot(data = Top_Category ,
                         mapping = aes(x = reorder(Category, -Percent_of_Sales),
                                       y = Percent_of_Sales,
                                       color = Category))+
  geom_bar(aes(fill = Category),
           stat = "identity",
           width = 0.5)+
  geom_text(aes(label = round(Percent_of_Sales, 2)),
            angle = 0,
            vjust = -0.3,
            hjust = 0.4,
            size = 3)+
  labs(title = "Percentage of Sales by Category",
       x = "Product Category",
       y = "Sales in Percent")+
  theme_bw()

Category_Percent_Graph <- Category_Percent_Graph +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1))

Category_Percent_Graph
```



Shows the percent of total sales by category.

```
Graph <- ggplot(data = Top_Products_by_date,
       mapping = aes(x = +Date,
                     y = TotalSales,
                     color = ProductName))+
  geom_bar(aes(fill = ProductName),
           stat = "identity",
           width = 0.25)+
  geom_text(aes(label = TotalSales),
            angle = 0,
            vjust = 0,
            hjust = -3,
            size = 3)+
  labs(title = "Sales Over Time by Product",
       x = "Date",
       y = "Sales in Dollars")+
  coord_flip()+
  theme_bw()

Graph
```

I used the insert picture method for above code because I deleted the code that produced the data frame for this graph and was being lazy.

This graph breaks downs daily sales by the share that each product contributed to total daily sales for the month of January.

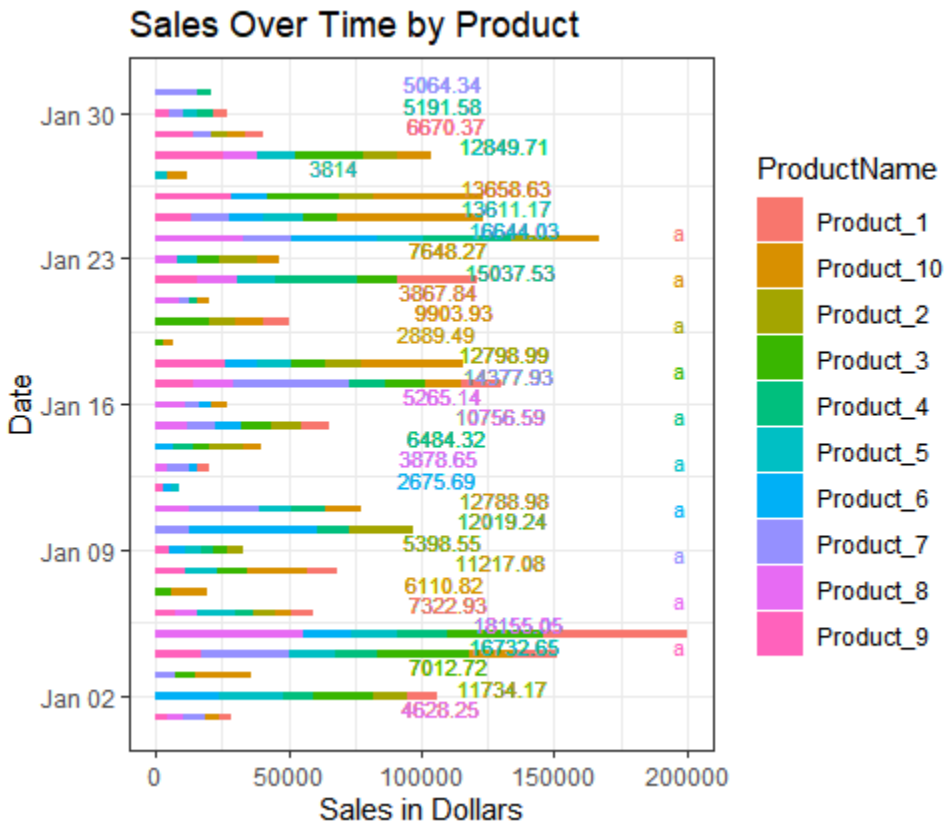


Figure 1: Graph of sales over time broken down by product

Step Three: Insight Generation and Recommendations

Insights

The First insight I would express to the company is that over all their Average sales declined by roughly $1,000.00 to $1,200.00 from the beginning to the end of the month of January.

The days that they saw the highest sales were the 4th, 5th, 11th, 17th, 18th, 22nd, 24th, 25th, 26th, and 28th with sales over $12,500.00 for the day.

The days with the least sales were the 12th, 16th, 19th, and 27th all well below $5,000.00 with the 12th being the worst day with just over $2,500.00 in sales.

The top performing product this company sells is product number ten with $53,029.86 in sales with and average price of $59.20.

The top performing category of products were furniture items totaling $69,732.36 in sales with and average price of $53.00.

Product number 3 and the electronics category were both second place in sales with $43,446.50 and $69,276.78 respectively for the month.

Product numbers one and eight are the company's worst selling products for the month of January with only $14,209.00 and $19846.03 in sales.

The company's most under performing category of items sold was toys only bringing in a total of $33,385.91 in sales.

Recommendations

Given that January the 2nd is a Monday, and that the most profitable days take place between Tuesday and Wednesday promoting a sale on Thursdays and Fridays could bring in more sales on the least profitable days.

Particularly targeting the least profitable items for sale bundles (i.e. buy one get one half off sales, etc. . .) could promote greater profitability with increase unit sales of otherwise "slow-moving stock" or "dead stock". This pertains to products one, two, four, eight, and nine as these are the bottom 50% of total sales.

Additionally, it should be considered whether or not to faze out products one and eight as they are the lowest performing products the company sales. This could be due to a competitor store selling the same product cheaper, and in that case a price adjustment (or sale) should be considered in order to draw-in more customers.

Toys are the least profitable category of product the company sales. Assuming this is a general goods store, similar to Walmart and Target, this could be due to the size of and stock of the toy section. A change of stock could boost sales by fazing out current stock for more profitable toys.