Group3 sales forecasting

Group3

2024-06-13

```r
if (!requireNamespace("dplyr", quietly = TRUE)) {
  install.packages("dplyr")
}
if (!requireNamespace("ggplot2", quietly = TRUE)) {
  install.packages("ggplot2")
}
if (!requireNamespace("readxl", quietly = TRUE)) {
  install.packages("readxl")
}

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(ggplot2)
library(readxl)

# Load the dataset
salesforecasting <- read_excel("C:\\Users\\srira\\Downloads\\salesforecasting.xlsx")
```

About Dataset This dataset offers a valuable resource for businesses operating in the retail furniture sector. By analyzing historical sales data from the superstore dataset, users can gain insights into future sales patterns and trends. This information can be utilized to optimize inventory management strategies, anticipate customer demand, and enhance overall operational efficiency. Whether for retail managers, analysts, or data scientists, this dataset provides a foundation for informed decision-making, helping businesses maintain stability and drive sustained growth in the dynamic retail environment.

```r
str(salesforecasting)
```

```
## tibble [558 x 21] (S3: tbl_df/tbl/data.frame)
```

Git link: https://github.com/Sriram2795/Group3_R_assignment

```
##  $ Row ID          : num [1:558] 1 2 4 6 11 24 25 28 30 37 ...
##  $ Order ID        : chr [1:558] "CA-2016-152156" "CA-2016-152156" "US-2015-108966" "CA-2014-115812" ..
##  $ Order Date      : chr [1:558] "42593" "42593" "42318" "41888" ...
##  $ Ship Date       : chr [1:558] "42685" "42685" "10/18/2015" "6/14/2014" ...
##  $ Ship Mode       : chr [1:558] "Second Class" "Second Class" "Standard Class" "Standard Class" ...
##  $ Customer ID     : chr [1:558] "CG-12520" "CG-12520" "SO-20335" "BH-11710" ...
##  $ Customer Name: chr [1:558] "Claire Gute" "Claire Gute" "Sean O'Donnell" "Brosina Hoffman" ...
##  $ Segment         : chr [1:558] "Consumer" "Consumer" "Consumer" "Consumer" ...
##  $ Country         : chr [1:558] "United States" "United States" "United States" "United States" ...
##  $ City            : chr [1:558] "Henderson" "Henderson" "Fort Lauderdale" "Los Angeles" ...
##  $ State           : chr [1:558] "Kentucky" "Kentucky" "Florida" "California" ...
##  $ Postal Code     : num [1:558] 42420 42420 33311 90032 90032 ...
##  $ Region          : chr [1:558] "South" "South" "South" "West" ...
##  $ Product ID      : chr [1:558] "FUR-BO-10001798" "FUR-CH-10000454" "FUR-TA-10000577" "FUR-FU-10001487
##  $ Category        : chr [1:558] "Furniture" "Furniture" "Furniture" "Furniture" ...
##  $ Sub-Category : chr [1:558] "Bookcases" "Chairs" "Tables" "Furnishings" ...
##  $ Product Name : chr [1:558] "Bush Somerset Collection Bookcase" "Hon Deluxe Fabric Upholstered Sta
##  $ Sales           : num [1:558] 262 731.9 957.6 48.9 1706.2 ...
##  $ Quantity        : num [1:558] 2 3 5 7 9 2 3 7 3 5 ...
##  $ Discount        : num [1:558] 0 0 0.45 0 0.2 0.3 0 0.5 0.2 0.6 ...
##  $ Profit          : num [1:558] 41.9 219.6 -383 14.2 85.3 ...
```

```
names(salesforecasting)
```

```
##  [1] "Row ID"         "Order ID"       "Order Date"     "Ship Date"
##  [5] "Ship Mode"      "Customer ID"    "Customer Name"  "Segment"
##  [9] "Country"        "City"           "State"          "Postal Code"
## [13] "Region"         "Product ID"     "Category"       "Sub-Category"
## [17] "Product Name"   "Sales"          "Quantity"       "Discount"
## [21] "Profit"
```

Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.

```
head(salesforecasting, n=15)
```

```
## # A tibble: 15 x 21
##     `Row ID` `Order ID`     `Order Date` `Ship Date` `Ship Mode`    `Customer ID`
##        <dbl> <chr>          <chr>        <chr>       <chr>          <chr>
##  1         1 CA-2016-152156 42593        42685       Second Class   CG-12520
##  2         2 CA-2016-152156 42593        42685       Second Class   CG-12520
##  3         4 US-2015-108966 42318        10/18/2015  Standard Class SO-20335
##  4         6 CA-2014-115812 41888        6/14/2014   Standard Class BH-11710
##  5        11 CA-2014-115812 41888        6/14/2014   Standard Class BH-11710
##  6        24 US-2017-156909 7/16/2017    7/18/2017   Second Class   SF-20065
##  7        25 CA-2015-106320 9/25/2015    9/30/2015   Standard Class EB-13870
##  8        28 US-2015-150630 9/17/2015    9/21/2015   Standard Class TB-21520
##  9        30 US-2015-150630 9/17/2015    9/21/2015   Standard Class TB-21520
## 10        37 CA-2016-117590 42594        42655       First Class    GH-14485
## 11        39 CA-2015-117415 12/27/2015   12/31/2015  Standard Class SN-20710
## 12        40 CA-2015-117415 12/27/2015   12/31/2015  Standard Class SN-20710
## 13        52 CA-2015-115742 4/18/2015    4/22/2015   Standard Class DP-13000
## 14          53 CA-2015-115742 4/18/2015     4/22/2015     Standard Class DP-13000
## 15          58 CA-2016-111682 6/17/2016     6/18/2016     First   Class     TB-21055
## # i 15 more variables: 'Customer Name' <chr>, Segment <chr>, Country <chr>,
## #   City <chr>, State <chr>, 'Postal Code' <dbl>, Region <chr>,
## #   'Product ID' <chr>, Category <chr>, 'Sub-Category' <chr>,
## #   'Product Name' <chr>, Sales <dbl>, Quantity <dbl>, Discount <dbl>,
## #   Profit <dbl>
```

```
calculate_total_sales_by_subcategory <- function(df, subcategory) {
  total_sales <- df %>%
    filter(salesforecasting$`Sub-Category` == subcategory) %>%
    summarize(TotalSales = sum(Sales, na.rm = TRUE))
  return(total_sales)
}

# Call the function with the dataset and the sub-category "Chairs"
total_sales_chairs <- calculate_total_sales_by_subcategory(salesforecasting, "Chairs")
print(total_sales_chairs)
```

```
## # A tibble: 1 x 1
##   TotalSales
##        <dbl>
## 1     82423.
```

```
# Define a simple user-defined function to filter the data by state and calculate total profit
total_profit_by_state <- function(df, state) {
  # Filter the data for the given state
  filtered_data <- df %>% filter(salesforecasting$State == state)

  # Calculate the total profit for the filtered data
  total_profit <- sum(filtered_data$Profit, na.rm = TRUE)

  return(total_profit)
}

# Call the function with the dataset and the state "California"
total_profit_california <- total_profit_by_state(salesforecasting, "California")
print(total_profit_california)
```

```
## [1] 1534.985
```

```
# Select relevant columns for the new data frame
selected_columns <- salesforecasting %>% select(`Order ID`,`Customer ID`,Segment,Category,`Sub-Category

# Summarize the data by Segment and Category
summary_df <- selected_columns %>%
group_by(Segment, Category) %>%
  summarize(Total_Profit = sum(Profit, na.rm = TRUE),
            Average_Sales = mean(Sales, na.rm = TRUE),
            Total_Quantity = sum(Quantity, na.rm = TRUE))
```

```
## 'summarise()' has grouped output by 'Segment'. You can override using the
## '.groups' argument.
```

```r
# Print the new data frame
print(summary_df)
```

```
## # A tibble: 3 x 5
## # Groups:   Segment [3]
##   Segment     Category  Total_Profit Average_Sales Total_Quantity
## <chr>         <chr>            <dbl>         <dbl>          <dbl>
## 1 Consumer    Furniture        -94.6          379.           1171
## 2 Corporate   Furniture        953.           373.            595
## 3 Home Office Furniture       1295.           363.            360
```

```r
# Remove rows with missing values
if(any(is.na(data))) {
  # Remove rows with missing values
  data_clean <- data %>% na.omit()

  # Display the cleaned data
  print("Data after removing rows with missing values:")
  print(data_clean)
} else {
  print("No missing values")
}
```

```
## Warning in is.na(data): is.na() applied to non-(list or vector) of type
## ’closure’
```

```
## [1] "No missing values"
```

```r
# Identify duplicated rows based on selected columns
duplicated_rows <- salesforecasting %>%
  group_by(`Order ID`, `Order Date`, `Ship Date`, `Customer ID`) %>%
  filter(!duplicated(`Order ID`))

# Print cleaned dataset
print(duplicated_rows)
```

```
## # A tibble:  475 x 21
## # Groups:    Order ID, Order Date, Ship Date, Customer ID [475]
##     ‘Row ID‘ ‘Order ID‘      ‘Order Date‘ ‘Ship Date‘ ‘Ship Mode‘    ‘Customer ID‘
##        <dbl> <chr>           <chr>        <chr>       <chr>          <chr>
##  1         1 CA-2016-152156  42593        42685       Second Class   CG-12520
##  2         4 US-2015-108966  42318        10/18/2015  Standard Class SO-20335
##  3         6 CA-2014-115812  41888        6/14/2014   Standard Class BH-11710
##  4        24 US-2017-156909  7/16/2017    7/18/2017   Second Class   SF-20065
##  5        25 CA-2015-106320  9/25/2015    9/30/2015   Standard Class EB-13870
##  6        28 US-2015-150630  9/17/2015    9/21/2015   Standard Class TB-21520
##  7        37 CA-2016-117590  42594        42655       First Class    GH-14485
##  8        39 CA-2015-117415  12/27/2015   12/31/2015  Standard Class SN-20710
##  9        52 CA-2015-115742  4/18/2015    4/22/2015   Standard Class DP-13000
## 10        58 CA-2016-111682  6/17/2016    6/18/2016   First Class    TB-21055
## # i 465 more rows
## # i 15 more variables: ‘Customer Name‘ <chr>, Segment <chr>, Country <chr>,
## #   City <chr>, State <chr>, ‘Postal Code‘ <dbl>, Region <chr>,
## #   ‘Product ID‘ <chr>, Category <chr>, ‘Sub-Category‘ <chr>,
## #   ‘Product Name‘ <chr>, Sales <dbl>, Quantity <dbl>, Discount <dbl>,
## #   Profit <dbl>
```

```
# Reorder rows based on Sales in descending order
data_ordered <- salesforecasting %>%
  arrange(desc(Sales))

# Print the reordered data
print(data_ordered)
```

```
## # A tibble: 558 x 21
##    ‘Row ID‘ ‘Order ID‘     ‘Order Date‘ ‘Ship Date‘ ‘Ship Mode‘    ‘Customer ID‘
##       <dbl> <chr>          <chr>        <chr>       <chr>          <chr>
##  1     1247 CA-2014-168494 41985        12/14/2014  Second Class   NP-18700
##  2       28 US-2015-150630 9/17/2015    9/21/2015   Standard Class TB-21520
##  3     1792 CA-2014-120474 41651        41710       First Class    RP-19390
##  4     2568 CA-2017-123967 42746        42805       Second Class   SF-20200
##  5     1439 CA-2015-139731 10/15/2015   10/15/2015  Same Day       JE-15745
##  6      400 CA-2016-108987 42591        42652       Second Class   AG-10675
##  7     1156 CA-2014-136567 12/20/2014   12/21/2014  First Class    PS-19045
##  8      950 US-2017-110576 11/28/2017   42778       Standard Class RB-19795
##  9      245 CA-2014-131926 41645        41796       Second Class   DW-13480
## 10      150 CA-2016-114489 42502        42625       Standard Class JE-16165
## # i 548 more rows
## # i 15 more variables: ‘Customer Name‘ <chr>, Segment <chr>, Country <chr>,
## #   City <chr>, State <chr>, ‘Postal Code‘ <dbl>, Region <chr>,
## #   ‘Product ID‘ <chr>, Category <chr>, ‘Sub-Category‘ <chr>,
## #   ‘Product Name‘ <chr>, Sales <dbl>, Quantity <dbl>, Discount <dbl>,
## #   Profit <dbl>
```

```
# Rename specific columns
data_renamed <- salesforecasting %>%
  rename(
    Row_ID = `Row ID`,
    Order_ID = `Order ID`,
    Order_Date = `Order Date`,
    Ship_Date = `Ship Date`,
  )

# Print the data with renamed columns
print(data_renamed)
```

```
## # A tibble: 558 x 21
##    Row_ID Order_ID       Order_Date Ship_Date  ‘Ship Mode‘    ‘Customer ID‘
##     <dbl> <chr>          <chr>      <chr>      <chr>          <chr>
##  1      1 CA-2016-152156 42593      42685      Second Class   CG-12520
##  2      2 CA-2016-152156 42593      42685      Second Class   CG-12520
##  3      4 US-2015-108966 42318      10/18/2015 Standard Class SO-20335
##  4      6 CA-2014-115812 41888      6/14/2014  Standard Class BH-11710
##  5     11 CA-2014-115812 41888      6/14/2014  Standard Class BH-11710
##  6     24 US-2017-156909 7/16/2017  7/18/2017  Second Class   SF-20065
## 7       25 CA-2015-106320 9/25/2015  9/30/2015  Standard Class EB-13870
## 8       28 US-2015-150630 9/17/2015  9/21/2015  Standard Class TB-21520
## 9       30 US-2015-150630 9/17/2015  9/21/2015  Standard Class TB-21520
## 10      37 CA-2016-117590 42594      42655      First Class    GH-14485
## # i 548 more rows
## # i 15 more variables: 'Customer Name' <chr>, Segment <chr>, Country <chr>,
## #   City <chr>, State <chr>, 'Postal Code' <dbl>, Region <chr>,
## #   'Product ID' <chr>, Category <chr>, 'Sub-Category' <chr>,
## #   'Product Name' <chr>, Sales <dbl>, Quantity <dbl>, Discount <dbl>,
## #   Profit <dbl>
```

```r
# Add a new variable by multiplying an existing column by 2
data_modified <- salesforecasting %>%
  mutate(SalesTwice = Sales * 2)

# Print the modified data frame with the new variable
print(data_modified)
```

```
## # A tibble: 558 x 22
##    'Row ID' 'Order ID'     'Order Date' 'Ship Date' 'Ship Mode'    'Customer ID'
##       <dbl> <chr>          <chr>        <chr>       <chr>          <chr>
##  1        1 CA-2016-152156 42593        42685       Second Class   CG-12520
##  2        2 CA-2016-152156 42593        42685       Second Class   CG-12520
##  3        4 US-2015-108966 42318        10/18/2015  Standard Class SO-20335
##  4        6 CA-2014-115812 41888        6/14/2014   Standard Class BH-11710
##  5       11 CA-2014-115812 41888        6/14/2014   Standard Class BH-11710
##  6       24 US-2017-156909 7/16/2017    7/18/2017   Second Class   SF-20065
##  7       25 CA-2015-106320 9/25/2015    9/30/2015   Standard Class EB-13870
##  8       28 US-2015-150630 9/17/2015    9/21/2015   Standard Class TB-21520
##  9       30 US-2015-150630 9/17/2015    9/21/2015   Standard Class TB-21520
## 10       37 CA-2016-117590 42594        42655       First Class    GH-14485
## # i 548 more rows
## # i 16 more variables: 'Customer Name' <chr>, Segment <chr>, Country <chr>,
## #   City <chr>, State <chr>, 'Postal Code' <dbl>, Region <chr>,
## #   'Product ID' <chr>, Category <chr>, 'Sub-Category' <chr>,
## #   'Product Name' <chr>, Sales <dbl>, Quantity <dbl>, Discount <dbl>,
## #   Profit <dbl>, SalesTwice <dbl>
```

```r
set.seed(123)

# Number of rows in the training set
train_size <- 0.7 * nrow(salesforecasting)  # Adjust 0.7 to your desired proportion

# Generate indices for training set
train_indices <- sample(seq_len(nrow(salesforecasting)), size = train_size, replace = FALSE)

# Create training set
train_set <- salesforecasting[train_indices, ]

# Print the first few rows of the training set
print(head(train_set))
```

```
## # A tibble: 6 x 21
##    `Row ID` `Order ID`      `Order Date` `Ship Date` `Ship Mode`     `Customer ID`
##       <dbl> <chr>           <chr>        <chr>       <chr>           <chr>
## 1      2004 CA-2017-163510 12/25/2017   12/28/2017  Second Class    JW-15955
## 2      2232 CA-2017-157091 6/26/2017    42742       Standard Class  DB-13405
## 3       848 CA-2015-114300 10/13/2015   10/17/2015  Standard Class  AF-10885
## 4      2544 US-2016-114174 42591        9/14/2016   Standard Class  AP-10720
## 5       949 US-2017-110576 11/28/2017   42778       Standard Class  RB-19795
## 6       540 CA-2015-134894 42197        42320       Standard Class  DK-12985
## # i 15 more variables: `Customer Name` <chr>, Segment <chr>, Country <chr>,
## #   City <chr>, State <chr>, `Postal Code` <dbl>, Region <chr>,
## #   `Product ID` <chr>, Category <chr>, `Sub-Category` <chr>,
## #   `Product Name` <chr>, Sales <dbl>, Quantity <dbl>, Discount <dbl>,
## #   Profit <dbl>
```

```
summary(salesforecasting)
```

```
##      Row ID          Order ID           Order Date          Ship Date
##  Min.   :   1.0   Length:558         Length:558         Length:558
##  1st Qu.: 663.8   Class :character   Class :character   Class :character
##  Median :1337.5   Mode  :character   Mode  :character   Mode  :character
##  Mean   :1339.2
##  3rd Qu.:2025.5
##  Max.   :2695.0
##   Ship Mode         Customer ID        Customer Name        Segment
##  Length:558         Length:558         Length:558         Length:558
##   Class  :character   Class  :character   Class  :character   Class  :character
##  Mode   :character   Mode   :character   Mode   :character   Mode   :character
##
##
##
##    Country             City              State            Postal Code
##  Length:558         Length:558         Length:558         Min.   : 1040
##  Class :character   Class :character   Class :character   1st Qu.:20016
##  Mode  :character   Mode  :character   Mode  :character   Median :55654
##                                                           Mean   :54526
##                                                           3rd Qu.:90004
##                                                           Max.   :99207
##     Region           Product ID          Category         Sub-Category
##  Length:558         Length:558         Length:558         Length:558
##  Class :character   Class :character   Class :character    Class :character
##  Mode  :character   Mode  :character   Mode  :character      Mode  :character
##
##
##
##  Product Name           Sales             Quantity         Discount
##  Length:558          Min.   :   1.988   Min.   : 1.00   Min.   :0.0000
##  Class :character   1st Qu.:  47.943   1st Qu.: 2.00   1st Qu.:0.0000
##  Mode  :character    Median : 199.220   Median : 3.00   Median :0.2000
##                     Mean   : 374.897   Mean   : 3.81   Mean   :0.1719
##                     3rd Qu.: 516.336   3rd Qu.: 5.00   3rd Qu.:0.2000
##                     Max.   :3610.848   Max.   :14.00   Max.   :0.7000
##      Profit
##  Min.   :-1665.052
##  1st Qu.:  -11.987
##   Median :     9.587
##   Mean   :     3.859
##   3rd Qu.:    40.736
##   Max.   :   673.882
```

```
mean_sales <- mean(salesforecasting$Sales, na.rm = TRUE)
print(mean_sales)
```

```
## [1] 374.8968
```

```
median_sales <- median(salesforecasting$Sales, na.rm = TRUE)
print(median_sales)
```

```
## [1] 199.22
```

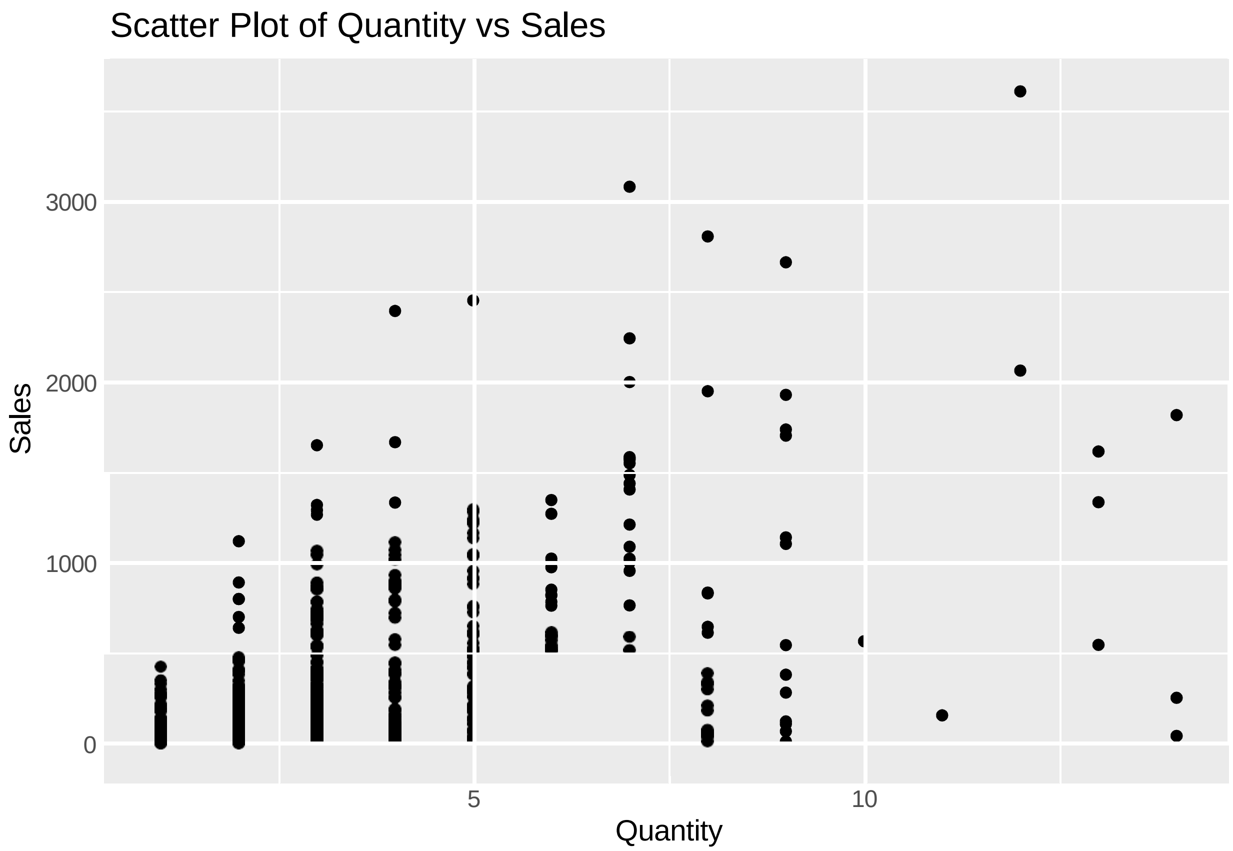
```
mode_sales <- as.numeric(names(sort(table(salesforecasting$Sales), decreasing = TRUE)[1]))
print(mode_sales)
```

```
## [1] 301.96
```

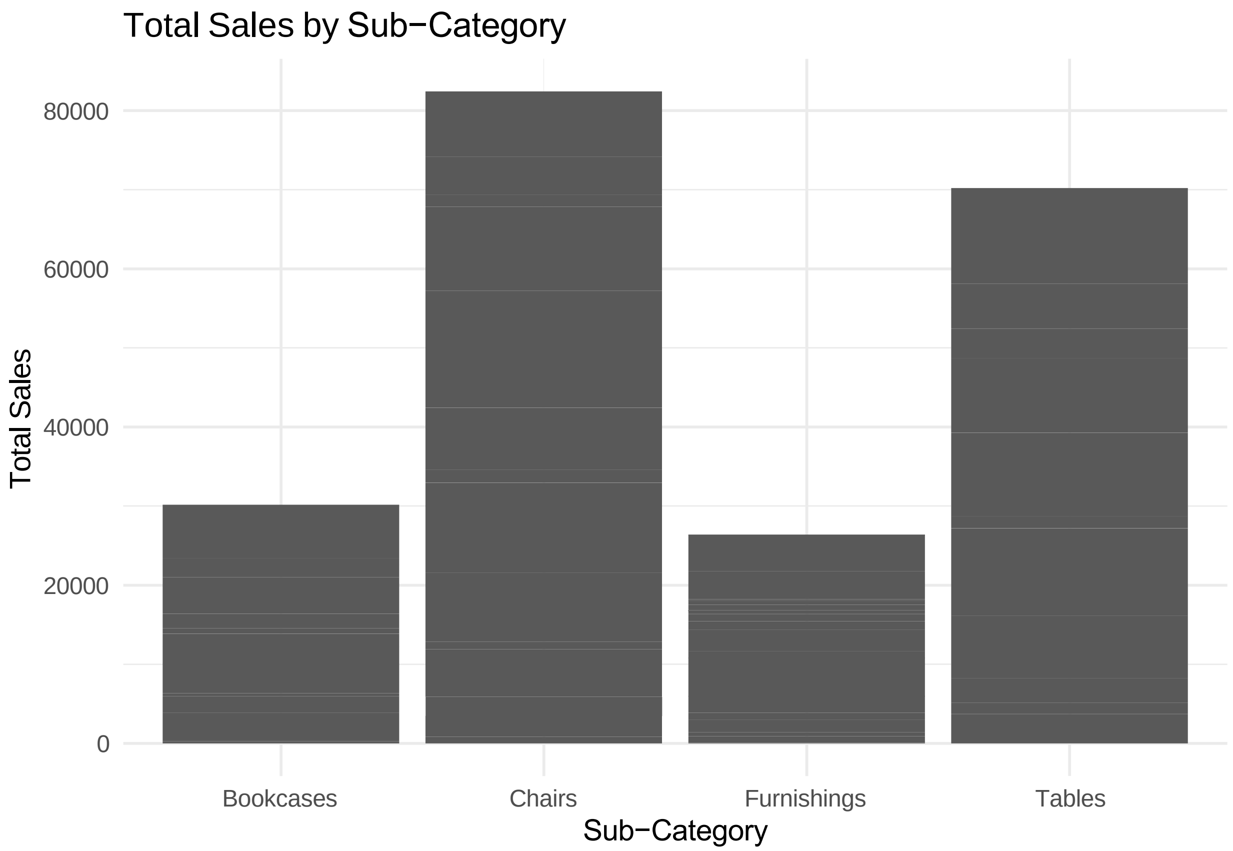
```
range_sales <- range(salesforecasting$Sales, na.rm = TRUE)
print(range_sales)
```

```
## [1]     1.988 3610.848
```

```
ggplot(salesforecasting, aes(x = Quantity, y = Sales)) + geom_point() + labs(title = "Scatter Plot of Q
```



```r
#ggplot(salesforecasting, aes(x = Category, y = Sales)) + geom_bar(stat = "identity") + labs(title = "B
ggplot(salesforecasting, aes(x = `Sub-Category`, y = Sales)) +
  geom_bar(stat = "identity") +
  labs(title = "Total Sales by Sub-Category", x = "Sub-Category", y = "Total Sales") +
  theme_minimal()
```



```r
correlation_sales_profit <- cor(salesforecasting$Sales, salesforecasting$Profit, method = "pearson")

# Print the correlation
print(correlation_sales_profit)
```

```
## [1] 0.004219722
```