Data 622 - Machine Learning and Big Data - HW #1 - Exploratory Analysis and Essay

Enid Roman

2024-03-09

```
# Upload the libraries needed.
library(rpart.plot)
library(rpart)
library(RColorBrewer)
library(corrplot)
library(GGally)
library(scales)
library(tidyr)
library(tidyverse)
library(dplyr)
library(ggplot2)
library(tabulate)
library(knitr)
library(kableExtra)
library(summarytools)
library(stats)
library(class)
```

```
# Import the data from github.

urlfile1 <- "https://raw.githubusercontent.com/enidroman/Data-622-Machine-Learning-and-Big-Data/main/10
urlfile2 <- "https://raw.githubusercontent.com/enidroman/Data-622-Machine-Learning-and-Big-Data/main/10
table_small_df <- read.csv(urlfile1)
table_large_df <- read.csv(urlfile2)
```

In the realm of business analytics, a thorough exploration of our dataset reveals intriguing patterns and variations in key numerical indicators. These metrics not only serve as numerical snapshots of our business operations but also offer valuable insights into the underlying dynamics.

Small Dataset

Content of the Tables:

```
# head() displays the first few rows of a dataframe, giving you a quick look at the data.
head(table_small_df)
```

Head of the Dataframe:

```
##                              Region               Country       Item.Type
## 1             Australia and Oceania                Tuvalu       Baby Food
## 2 Central America and the Caribbean               Grenada          Cereal
## 3                            Europe                Russia Office Supplies
## 4               Sub-Saharan Africa Sao Tome and Principe          Fruits
## 5               Sub-Saharan Africa                Rwanda Office Supplies
## 6             Australia and Oceania       Solomon Islands       Baby Food
##   Sales.Channel Order.Priority Order.Date  Order.ID Ship.Date Units.Sold
## 1       Offline              H  5/28/2010 669165933 6/27/2010       9925
## 2        Online              C  8/22/2012 963881480 9/15/2012       2804
## 3       Offline              L   5/2/2014 341417157  5/8/2014       1779
## 4        Online              C  6/20/2014 514321792  7/5/2014       8102
## 5       Offline              L   2/1/2013 115456712  2/6/2013       5062
## 6        Online              C   2/4/2015 547995746 2/21/2015       2974
##   Unit.Price Unit.Cost Total.Revenue Total.Cost Total.Profit
## 1     255.28    159.42    2533654.00 1582243.50    951410.50
## 2     205.70    117.11     576782.80  328376.44    248406.36
## 3     651.21    524.96    1158502.59  933903.84    224598.75
## 4       9.33      6.92      75591.66   56065.84     19525.82
## 5     651.21    524.96    3296425.02 2657347.52    639077.50
## 6     255.28    159.42     759202.72  474115.08    285087.64
```

```
# summary() provides a summary of the central tendency, dispersion, and distribution of the data, inclu

summary(table_small_df)
```

Summary Statistic:

```
##     Region             Country           Item.Type         Sales.Channel
##  Length:100         Length:100         Length:100         Length:100
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##  Order.Priority      Order.Date           Order.ID          Ship.Date
##  Length:100         Length:100         Min.   :114606559   Length:100
##  Class :character   Class :character   1st Qu.:338922488   Class :character
##  Mode  :character   Mode  :character   Median :557708561   Mode  :character
##                                        Mean   :555020412
##                                        3rd Qu.:790755081
##                                        Max.   :994022214
##    Units.Sold     Unit.Price       Unit.Cost      Total.Revenue
##  Min.   : 124   Min.   :  9.33   Min.   :  6.92   Min.   :   4870
##  1st Qu.:2836   1st Qu.: 81.73   1st Qu.: 35.84   1st Qu.: 268721
##  Median :5382   Median :179.88   Median :107.28   Median : 752314
##  Mean   :5129   Mean   :276.76   Mean   :191.05   Mean   :1373488
##  3rd Qu.:7369   3rd Qu.:437.20   3rd Qu.:263.33   3rd Qu.:2212045
##  Max.   :9925   Max.   :668.27   Max.   :524.96   Max.   :5997055
##    Total.Cost       Total.Profit
##  Min.   :   3612   Min.   :   1258
```

```
##  1st Qu.: 168868   1st Qu.: 121444
##  Median : 363566   Median : 290768
##  Mean   : 931806   Mean   : 441682
##  3rd Qu.:1613870   3rd Qu.: 635829
##  Max.   :4509794   Max.   :1719922
```

The dataset contains 100 observations for each variable, and the variables have different types (character, integer, numeric). The descriptive statistics give insights into the distribution and central tendency of the numeric variables.

```r
# The str() function provides an overview of the structure of an R object, showing the data type and st
str(table_small_df)
```

Structure of Tables:

```
## 'data.frame':    100 obs. of  14 variables:
##  $ Region        : chr  "Australia and Oceania" "Central America and the Caribbean" "Europe" "Sub-Sa
##  $ Country       : chr  "Tuvalu" "Grenada" "Russia" "Sao Tome and Principe" ...
##  $ Item.Type     : chr  "Baby Food" "Cereal" "Office Supplies" "Fruits" ...
##  $ Sales.Channel : chr  "Offline" "Online" "Offline" "Online" ...
##  $ Order.Priority: chr  "H" "C" "L" "C" ...
##  $ Order.Date    : chr  "5/28/2010" "8/22/2012" "5/2/2014" "6/20/2014" ...
##  $ Order.ID      : int  669165933 963881480 341417157 514321792 115456712 547995746 135425221 871543
##  $ Ship.Date     : chr  "6/27/2010" "9/15/2012" "5/8/2014" "7/5/2014" ...
##  $ Units.Sold    : int  9925 2804 1779 8102 5062 2974 4187 8082 6070 6593 ...
##  $ Unit.Price    : num  255.28 205.7 651.21 9.33 651.21 ...
##  $ Unit.Cost     : num  159.42 117.11 524.96 6.92 524.96 ...
##  $ Total.Revenue : num  2533654 576783 1158503 75592 3296425 ...
##  $ Total.Cost    : num  1582244 328376 933904 56066 2657348 ...
##  $ Total.Profit  : num  951411 248406 224599 19526 639078 ...
```

Again, the dataset consists of 100 observations and 14 variables. The variables include categorical features such as Region, Country, Item Type, Sales Channel, Order Priority, Order Date, and Ship Date. Additionally, there are numerical features, including Order ID, Units Sold, Unit Price, Unit Cost, Total Revenue, Total Cost, and Total Profit.

The categorical features provide information about the geographic and logistical aspects of each transaction, while the numerical features quantify the financial details, such as sales quantities, pricing, and associated costs. Notably, the dataset covers a diverse range of regions, countries, and product types, making it suitable for exploring sales trends and financial performance.

Dataset Characteristics (Structure, Size, Dependencies, Labels, etc.):

```r
# nrow() and ncol() give the number of rows and columns, respectively.
number_of_rows <- nrow(table_small_df)
print(paste("Number of rows:", number_of_rows))
```

Number of Rows and Columns:

```
## [1] "Number of rows: 100"
```

```
number_of_columns <- ncol(table_small_df)
print(paste("Number of rows:", number_of_columns))
```

```
## [1] "Number of rows: 14"
```

```
# names() shows the variable (column) names.
names(table_small_df)
```

```
##  [1] "Region"         "Country"        "Item.Type"      "Sales.Channel"
##  [5] "Order.Priority" "Order.Date"     "Order.ID"       "Ship.Date"
##  [9] "Units.Sold"     "Unit.Price"     "Unit.Cost"      "Total.Revenue"
## [13] "Total.Cost"     "Total.Profit"
```

```
# Get the total number of NA's in the entire dataframe
total_na_count <- sum(is.na(table_small_df))
print("Total number of NA's in the entire dataframe:")
```

Number of NA's:

```
## [1] "Total number of NA's in the entire dataframe:"
```

```
print(total_na_count)
```

```
## [1] 0
```

```
# Get the number of NA's in each column
na_per_column <- colSums(is.na(table_small_df))
print("Number of NA's in each column:")
```

```
## [1] "Number of NA's in each column:"
```

```
print(na_per_column)
```

```
##         Region        Country      Item.Type  Sales.Channel Order.Priority
##              0              0              0              0              0
##     Order.Date       Order.ID      Ship.Date     Units.Sold     Unit.Price
##              0              0              0              0              0
##      Unit.Cost  Total.Revenue     Total.Cost   Total.Profit
##              0              0              0              0
```

There are no NA's in this dataset.

```r
# Exclude "Order.ID" and non-numeric variables
numeric_table_small_df <- table_small_df[sapply(table_small_df, is.numeric) & colnames(table_small_df)

# Check if there are any missing values
if (any(is.na(numeric_table_small_df))) {
  cat("Warning: There are missing values in the numeric columns. Consider handling missing values before
} else {
  # Calculate standard deviation and variance for numeric variables
  std_dev <- apply(numeric_table_small_df, 2, sd)
  variance <- apply(numeric_table_small_df, 2, var)

  # Combine results into a data frame
  result_table_small_df <- data.frame(Variable = names(numeric_table_small_df), Standard_Deviation = st

 # Print the result as a nicely formatted table
  result_table <- kable(result_table_small_df, "html") %>%
    kable_styling()

  # Display the table
  print(result_table)
}
```

Standard Deviation and Variance:

```
## <table class="table" style="margin-left: auto; margin-right: auto;">
##  <thead>
##   <tr>
##    <th style="text-align:left;">   </th>
##    <th style="text-align:left;"> Variable </th>
##    <th style="text-align:right;"> Standard_Deviation </th>
##    <th style="text-align:right;"> Variance </th>
##   </tr>
##  </thead>
## <tbody>
##   <tr>
##    <td style="text-align:left;"> Units.Sold </td>
##    <td style="text-align:left;"> Units.Sold </td>
##    <td style="text-align:right;"> 2794.4846 </td>
##    <td style="text-align:right;"> 7.809144e+06 </td>
##   </tr>
##   <tr>
##    <td style="text-align:left;"> Unit.Price </td>
##    <td style="text-align:left;"> Unit.Price </td>
##    <td style="text-align:right;"> 235.5922 </td>
##    <td style="text-align:right;"> 5.550370e+04 </td>
##   </tr>
##   <tr>
##    <td style="text-align:left;"> Unit.Cost </td>
##    <td style="text-align:left;"> Unit.Cost </td>
##    <td style="text-align:right;"> 188.2082 </td>
##    <td style="text-align:right;"> 3.542232e+04 </td>
##   </tr>
```

```
##   <tr>
##    <td style="text-align:left;"> Total.Revenue </td>
##    <td style="text-align:left;"> Total.Revenue </td>
##    <td style="text-align:right;"> 1460028.7068 </td>
##    <td style="text-align:right;"> 2.131684e+12 </td>
##   </tr>
##   <tr>
##    <td style="text-align:left;"> Total.Cost </td>
##    <td style="text-align:left;"> Total.Cost </td>
##    <td style="text-align:right;"> 1083938.2522 </td>
##    <td style="text-align:right;"> 1.174922e+12 </td>
##   </tr>
##   <tr>
##    <td style="text-align:left;"> Total.Profit </td>
##    <td style="text-align:left;"> Total.Profit </td>
##    <td style="text-align:right;"> 438537.9071 </td>
##    <td style="text-align:right;"> 1.923155e+11 </td>
##   </tr>
## </tbody>
## </table>
```

Examining the 'Units Sold' metric, we observe a considerable range in the quantity of items sold. This variance suggests a diverse sales landscape, possibly influenced by factors such as product popularity, market demand, or seasonal fluctuations.

'Unit Price,' the stability in its values indicates a consistent pricing strategy across different products or time periods. This uniformity in unit prices may be a deliberate choice or could signify a need for reassessment in pricing strategies.

'Unit Cost' displays a moderate level of variability. This metric, representing the cost incurred for each unit, could be influenced by factors like raw material prices, production efficiency, or supplier negotiations.

Shifting our focus to broader financial indicators, 'Total Revenue' showcases substantial fluctuations. This variability in overall income suggests that our business experiences diverse sales performances, potentially tied to external market dynamics or internal factors.

'Total Cost' also demonstrates notable variations, encompassing various expenses associated with our business operations. Understanding these fluctuations is vital for effective cost management and resource allocation.

'Total Profit' metric, reflecting the net financial outcome, displays changes that warrant further investigation. Unraveling the factors influencing these profit variations will be pivotal in devising strategies to enhance overall business performance.

Relationships between numeric variables using a correlation matrix.

```r
# Identify numeric columns
numeric_columns <- sapply(table_small_df, is.numeric)

# Correlation matrix for numeric columns
cor_matrix <- cor(table_small_df[, numeric_columns])
cor_matrix
```

See below visualization for Correlation.

```
##                 Order.ID  Units.Sold  Unit.Price   Unit.Cost Total.Revenue
## Order.ID       1.0000000 -0.22290682 -0.19094121 -0.21320058    -0.3146876
## Units.Sold    -0.2229068  1.00000000 -0.07048559 -0.09223245     0.4477845
## Unit.Price    -0.1909412 -0.07048559  1.00000000  0.98726981     0.7523596
## Unit.Cost     -0.2132006 -0.09223245  0.98726981  1.00000000     0.7156226
## Total.Revenue -0.3146876  0.44778449  0.75235960  0.71562263     1.0000000
## Total.Cost    -0.3289444  0.37474592  0.78790543  0.77489522     0.9839277
## Total.Profit  -0.2346378  0.56455046  0.55736525  0.46721391     0.8973269
##               Total.Cost Total.Profit
## Order.ID      -0.3289444   -0.2346378
## Units.Sold     0.3747459    0.5645505
## Unit.Price     0.7879054    0.5573652
## Unit.Cost      0.7748952    0.4672139
## Total.Revenue  0.9839277    0.8973269
## Total.Cost     1.0000000    0.8040911
## Total.Profit   0.8040911    1.0000000
```

The summarized correlation matrix indicates the relationships between various pairs of variables in the dataset. Here are some key observations:

"Units Sold" and "Total Profit" have a strong positive correlation of 0.5645, suggesting that higher unit sales tend to be associated with increased total profits.

"Unit Price" and "Unit Cost" exhibit a strong positive correlation of 0.9873, indicating a nearly linear relationship between these two variables.

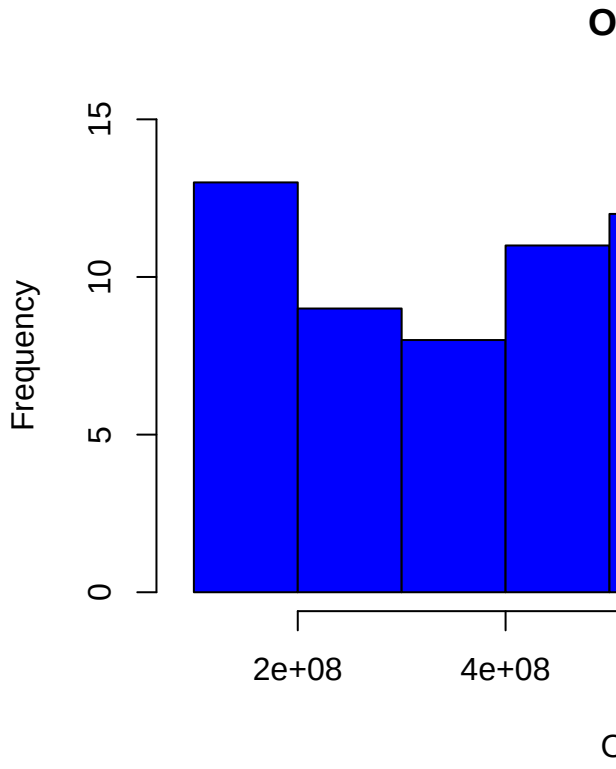
"Total Revenue" and "Total Cost" have a high positive correlation of 0.9839, implying that these variables are strongly related, as expected.

"Order ID" has a relatively weak negative correlation with most variables, indicating limited linear relationships with other features in the dataset.

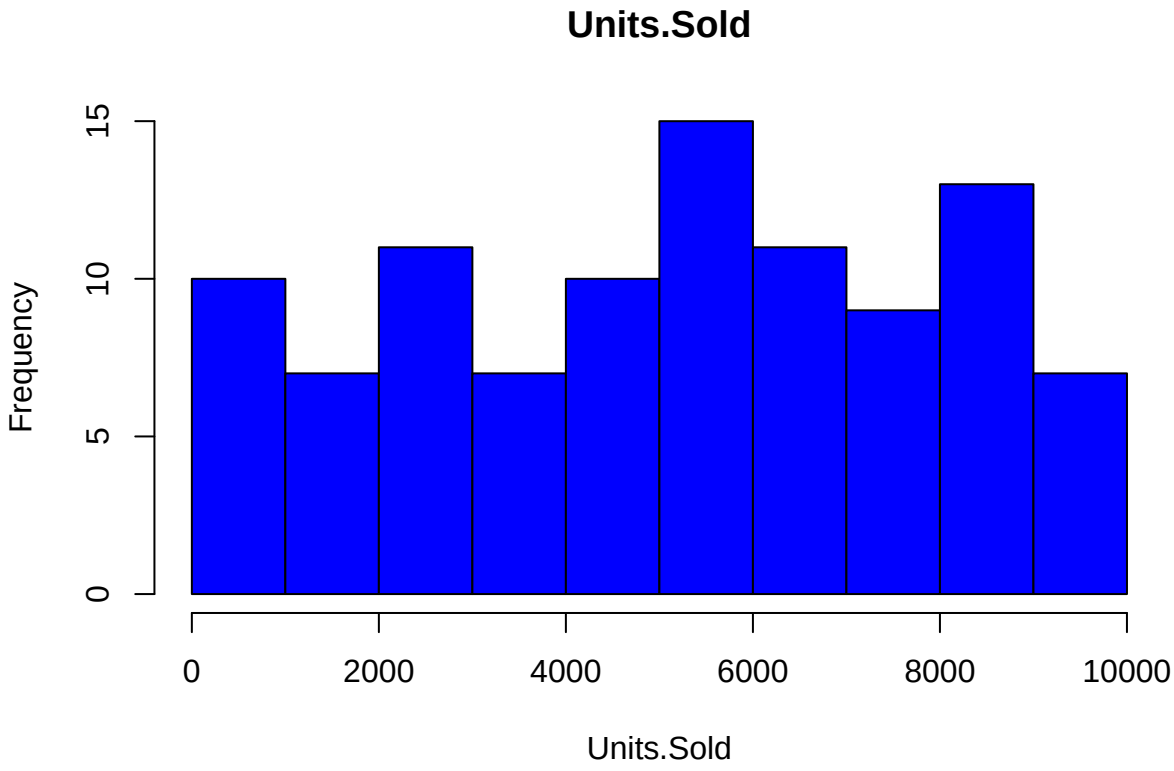
The correlation matrix provides valuable insights into the linear associations between different pairs of variables. Positive correlations suggest a positive relationship, while negative correlations indicate an inverse relationship. The strength of correlation is indicated by the magnitude of the correlation coefficient, with values closer to 1 or -1 indicating stronger associations.

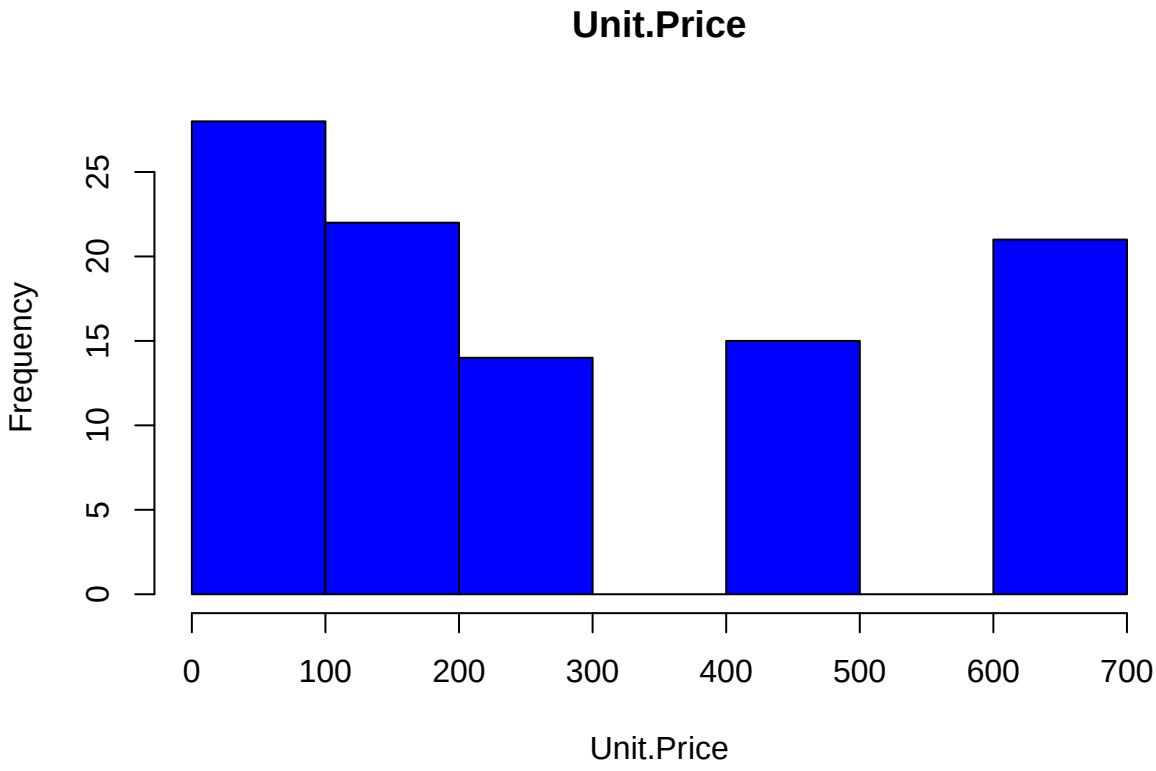
Visualization

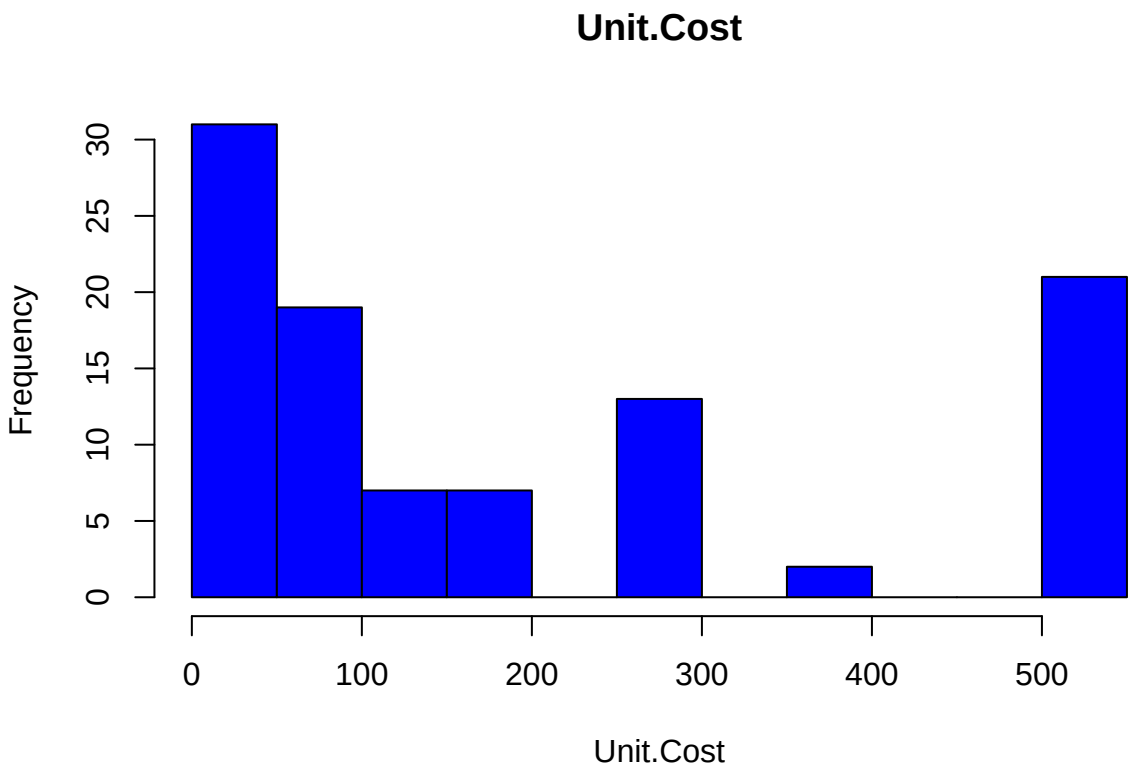
```r
# Identify numeric columns
numeric_columns <- sapply(table_small_df, is.numeric)

# Create histograms for numeric columns with blue color
for (col in names(table_small_df)[numeric_columns]) {
  hist(table_small_df[[col]], main = col, xlab = col, col = "blue")
}
```

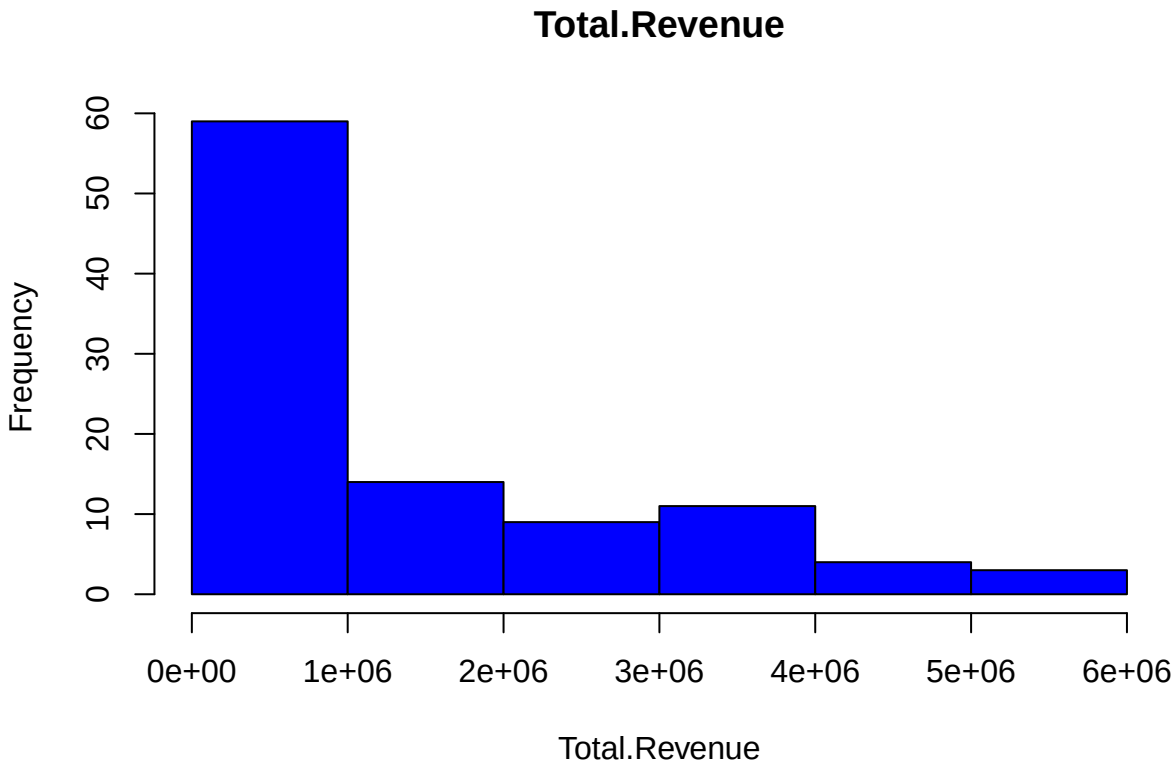


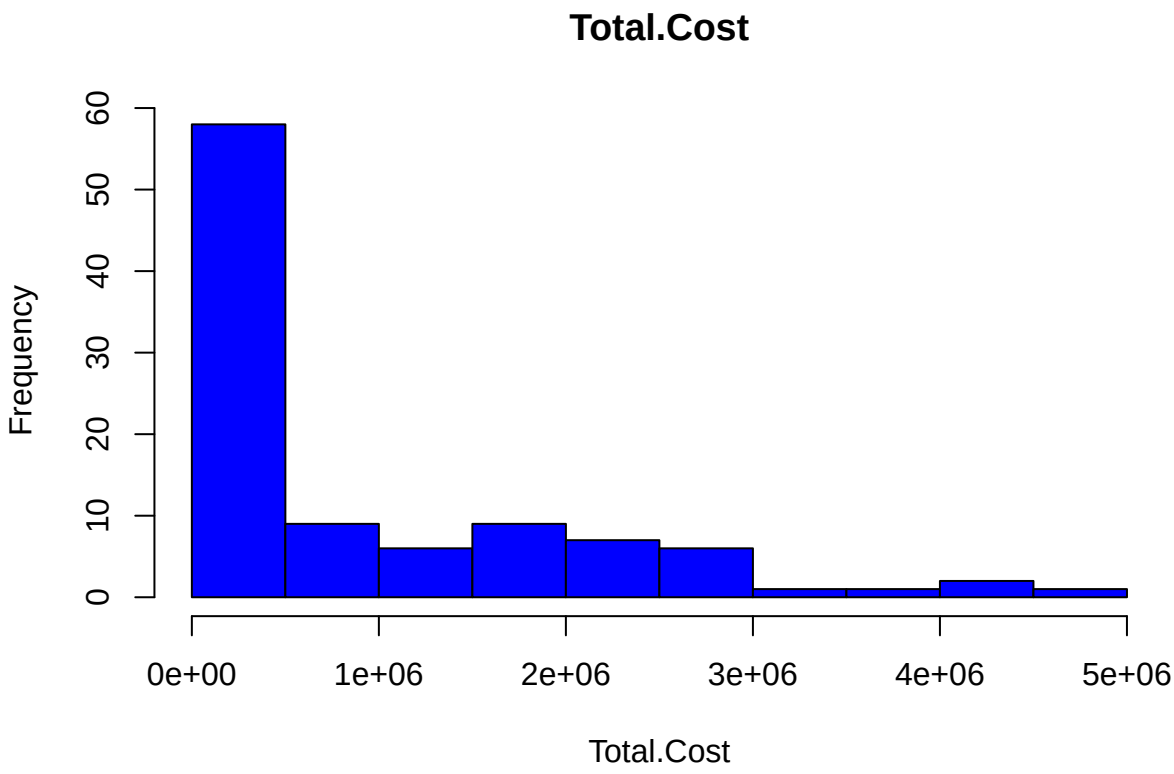
Histograms to visualize the distribution of numeric variables.

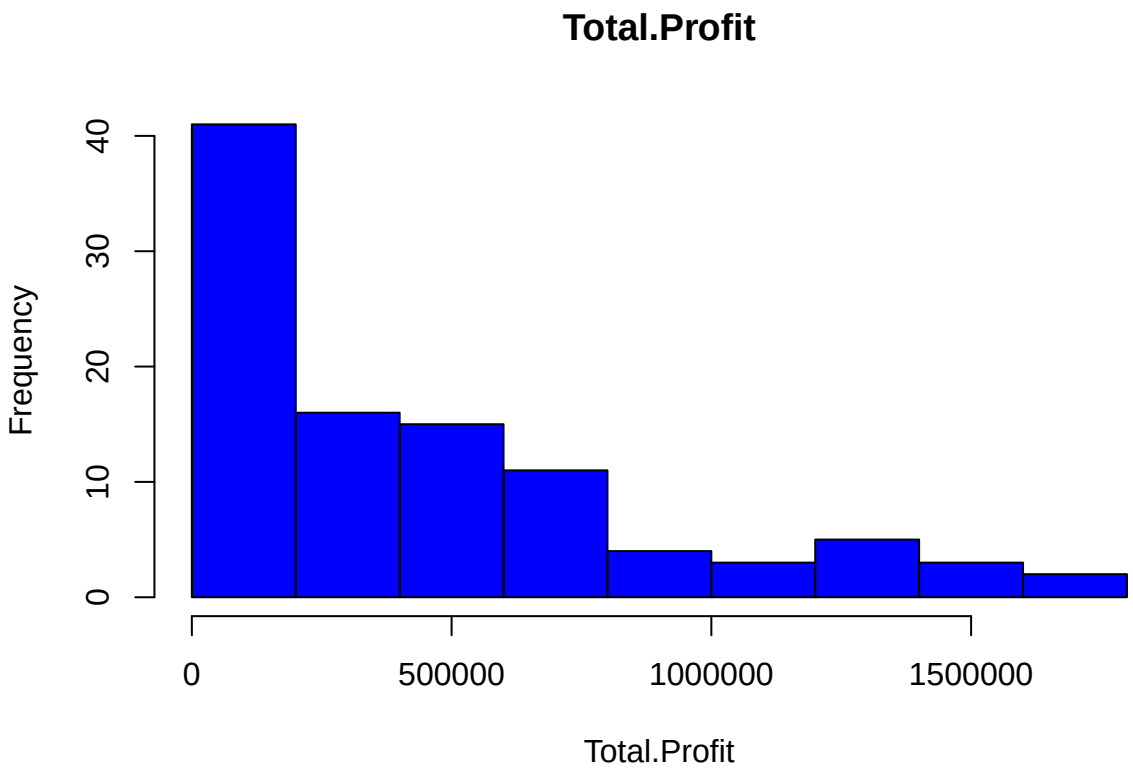



Unit.Price
Frequency
0
5
10
15
20
25
0
100
200
300
400
500
600
700
Unit.Price


Unit.Cost
Frequency
0
5
10
15
20
25
30
0
100
200
300
400
500
Unit.Cost

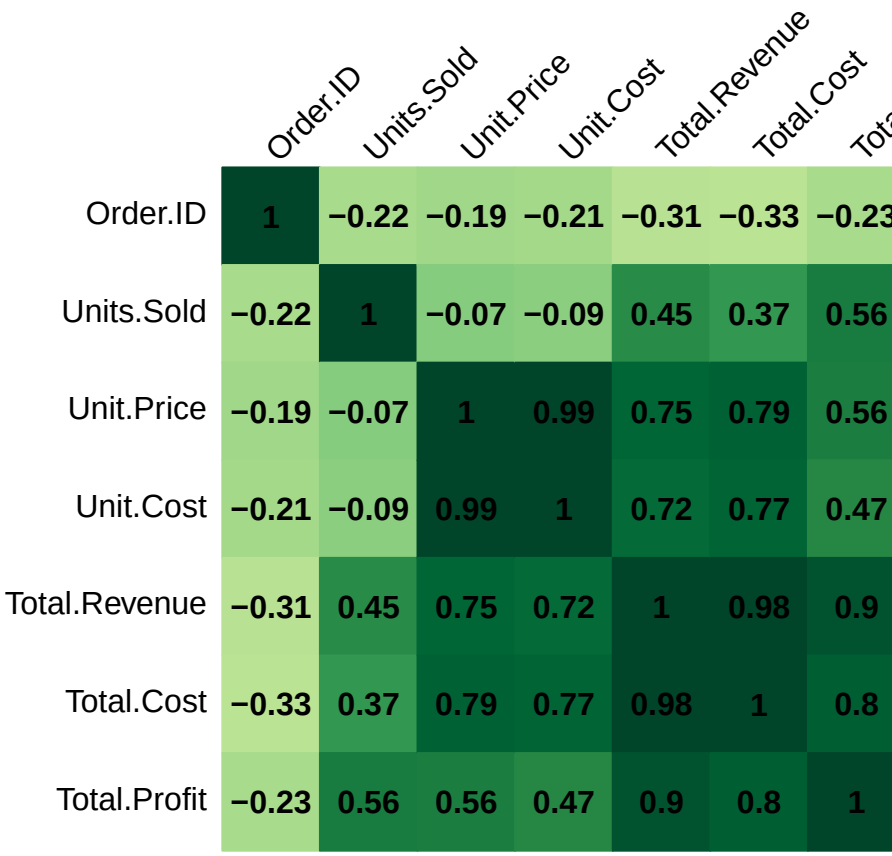

Total.Revenue
Frequency
0
10
20
30
40
50
60
0e+00
1e+06
2e+06
3e+06
4e+06
5e+06
6e+06
Total.Revenue


Total.Cost
Frequency
0
10
20
30
40
50
60
0e+00
1e+06
2e+06
3e+06
4e+06
5e+06
Total.Cost



For both "Order ID" and "Units Sold," the histograms show a uniform distribution. This suggests that values are spread relatively evenly across the range, and there's no clear tendency for values to cluster around specific points. The histogram for "Unit Price" is described as right-skewed. Right-skewed distributions have a tail extending to the right, indicating that there are fewer higher values but with a few extremely high values (outliers) pulling the distribution to the right. The histograms for "Total Revenue," "Total Cost," and "Total Profit" are all described as right-skewed. Similar to the Unit Price, this indicates that these variables have a distribution with a tail extending to the right, implying the presence of relatively fewer higher values.

```
# Create a full correlation plot with a lighter color palette
corrplot(cor_matrix, method = "color", addCoef.col = "black", tl.col = "black", tl.srt = 45, col = colo
```



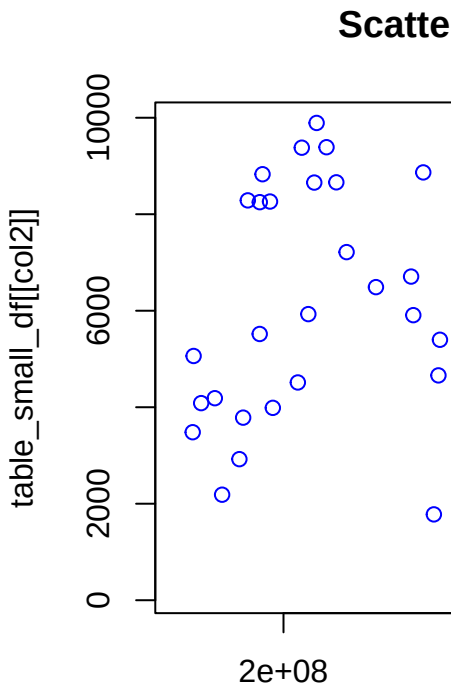
Correlation between variables.
The correlation coefficients in the provided dataset offer valuable insights into the relationships between various key variables. Notably, a correlation coefficient of 0.99 between Unit Cost and Unit Price indicates an exceptionally strong positive correlation. This implies that as the cost of producing a unit increases, there is a proportional increase in its selling price. This tight relationship suggests a direct link between production costs and pricing.

Moving on to Total Cost and Total Revenue, a correlation coefficient of 0.98 suggests a robust positive correlation. This implies that an increase in the total costs incurred in production corresponds to a proportional increase in the total revenue generated. This strong positive correlation underscores the interconnectedness of production costs and overall revenue.

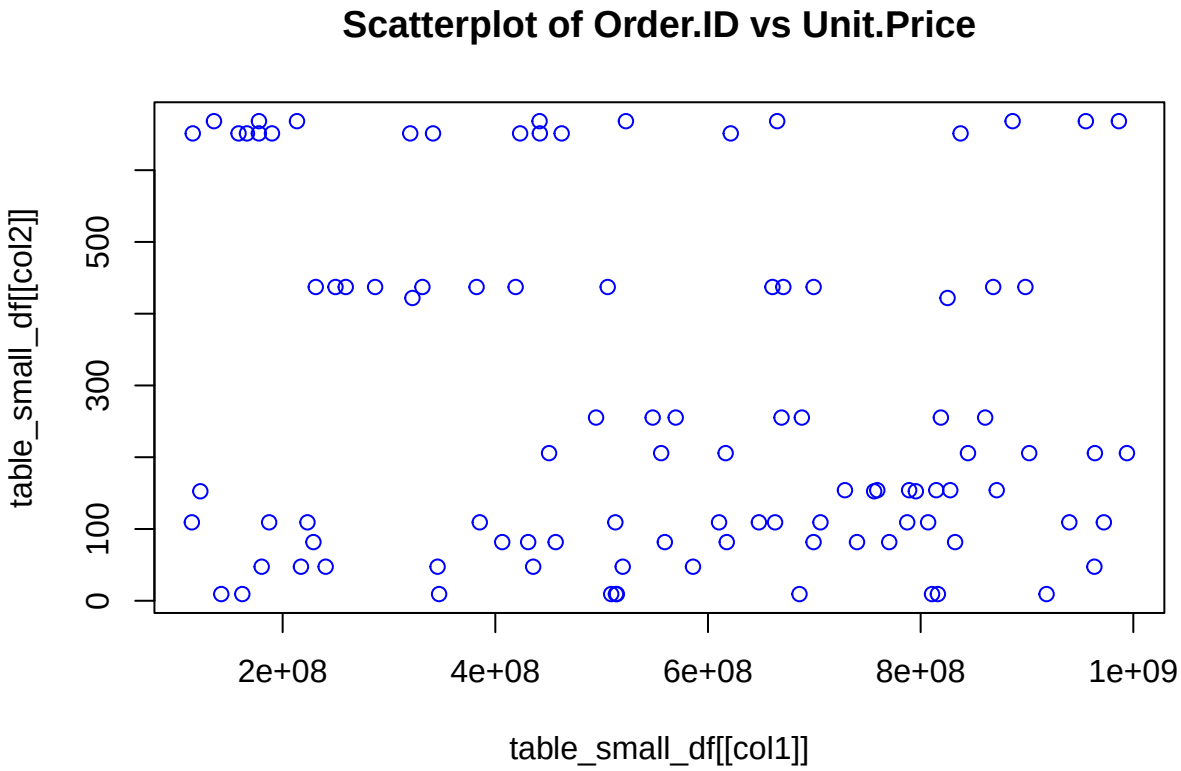
The correlation coefficient of 0.90 between Total Profit and Total Revenue indicates a strong positive correlation, albeit slightly less intense than the previous examples. This suggests that as total profits increase, there is a tendency for total revenue to increase as well. While not as extreme, this correlation emphasizes a meaningful relationship between overall profitability and revenue.

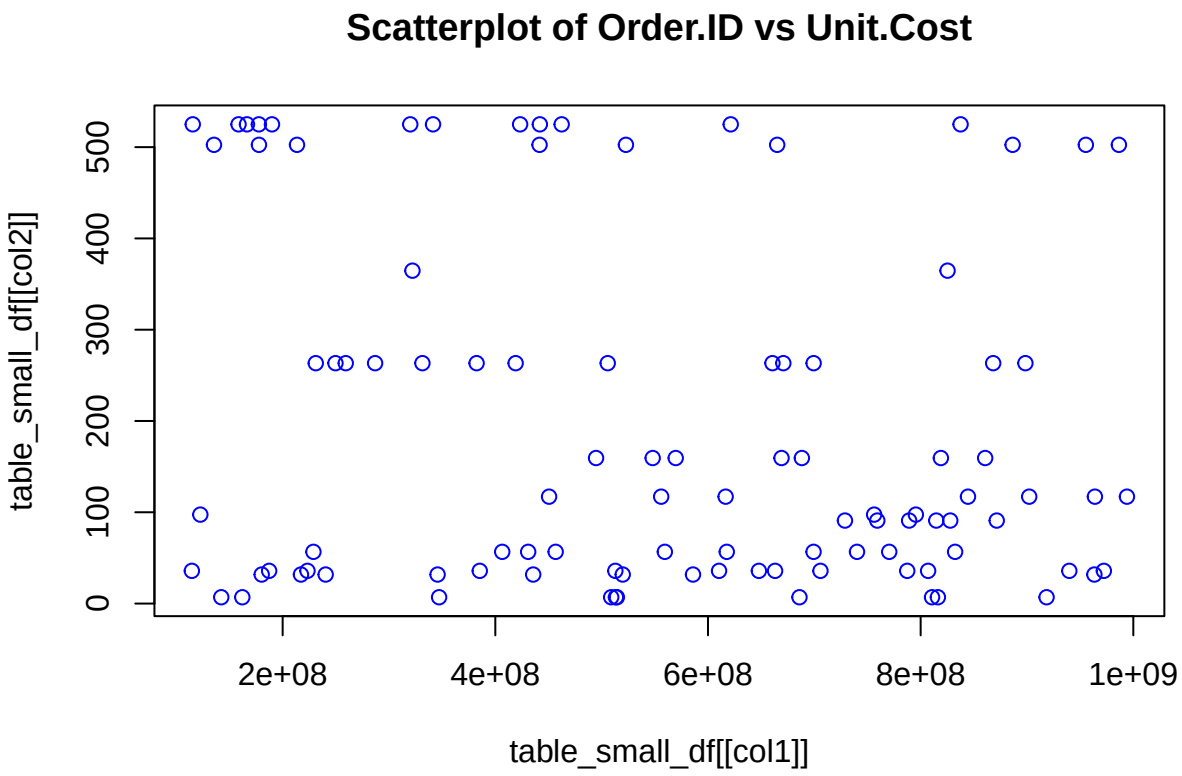
```r
# Assuming your dataframe is named 'table_small_df'
# Identify numeric columns
numeric_columns <- sapply(table_small_df, is.numeric)

# Create individual scatterplots for pairs of numeric columns using plot
for (col1 in names(table_small_df)[numeric_columns]) {
  for (col2 in names(table_small_df)[numeric_columns]) {
    if (col1 != col2) {
      plot(table_small_df[[col1]], table_small_df[[col2]], main = paste("Scatterplot of", col1, "vs", co
```

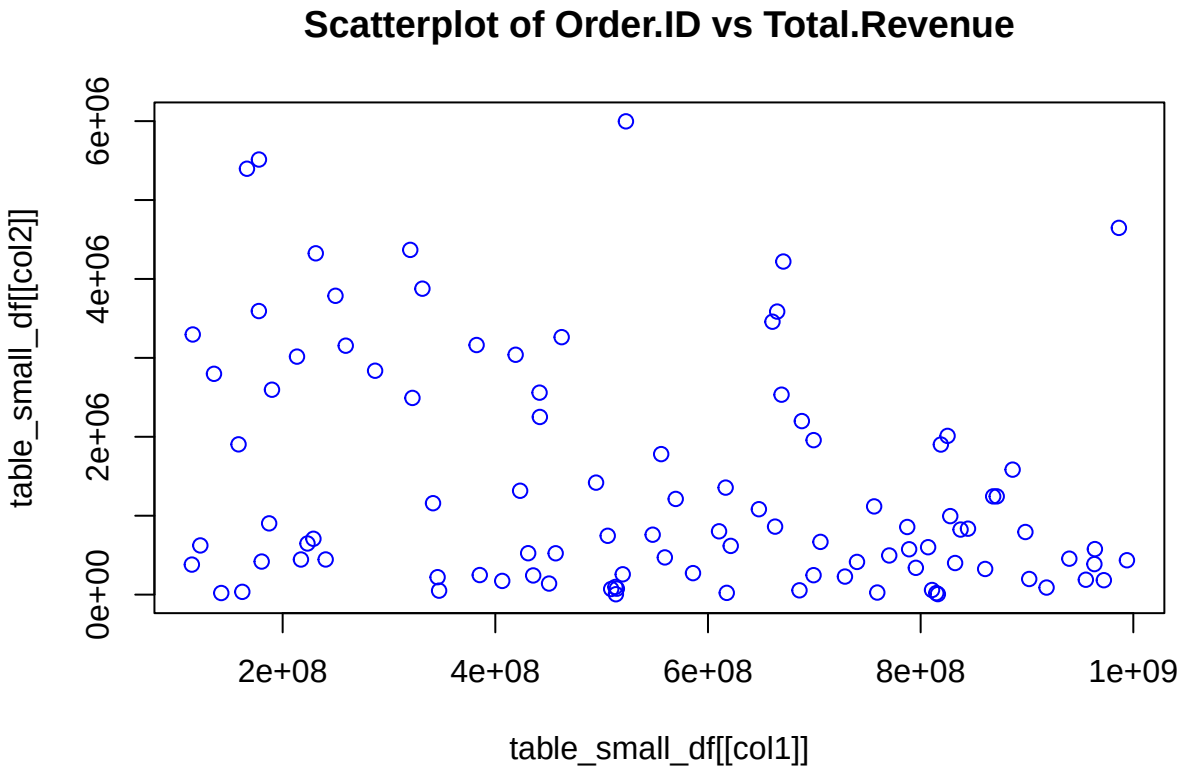
```
    }
  }
}
```

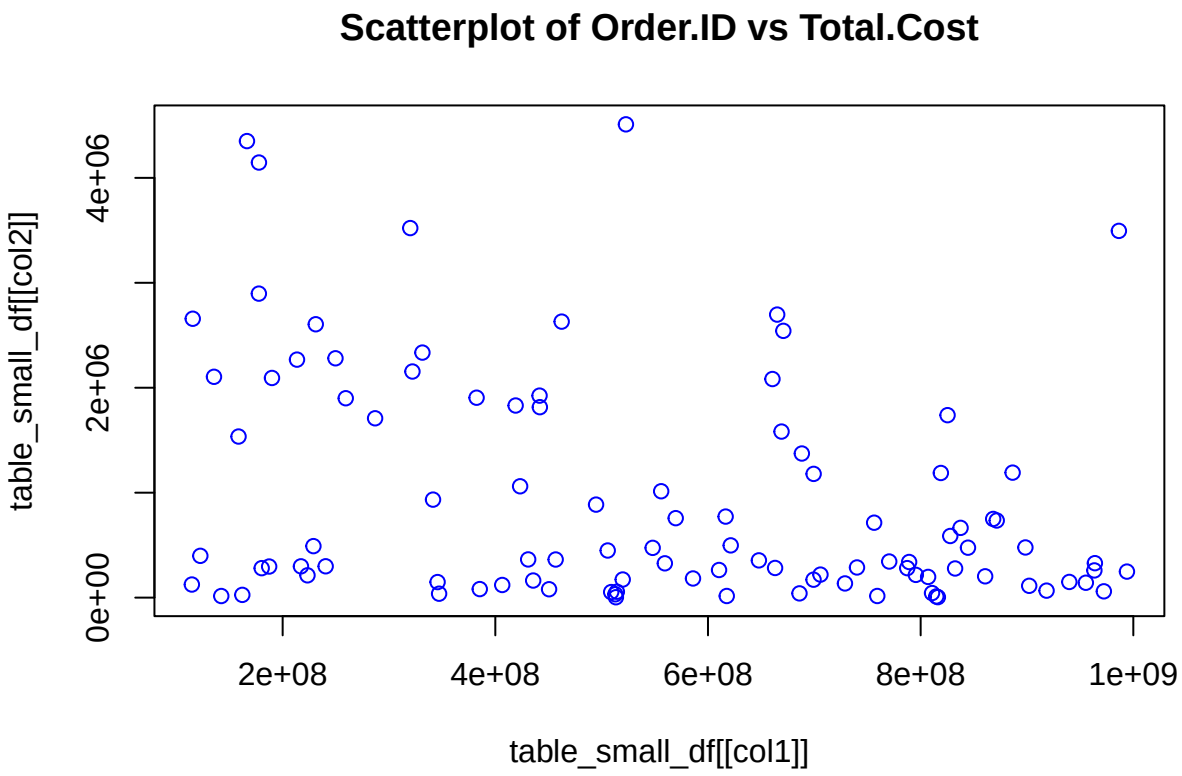


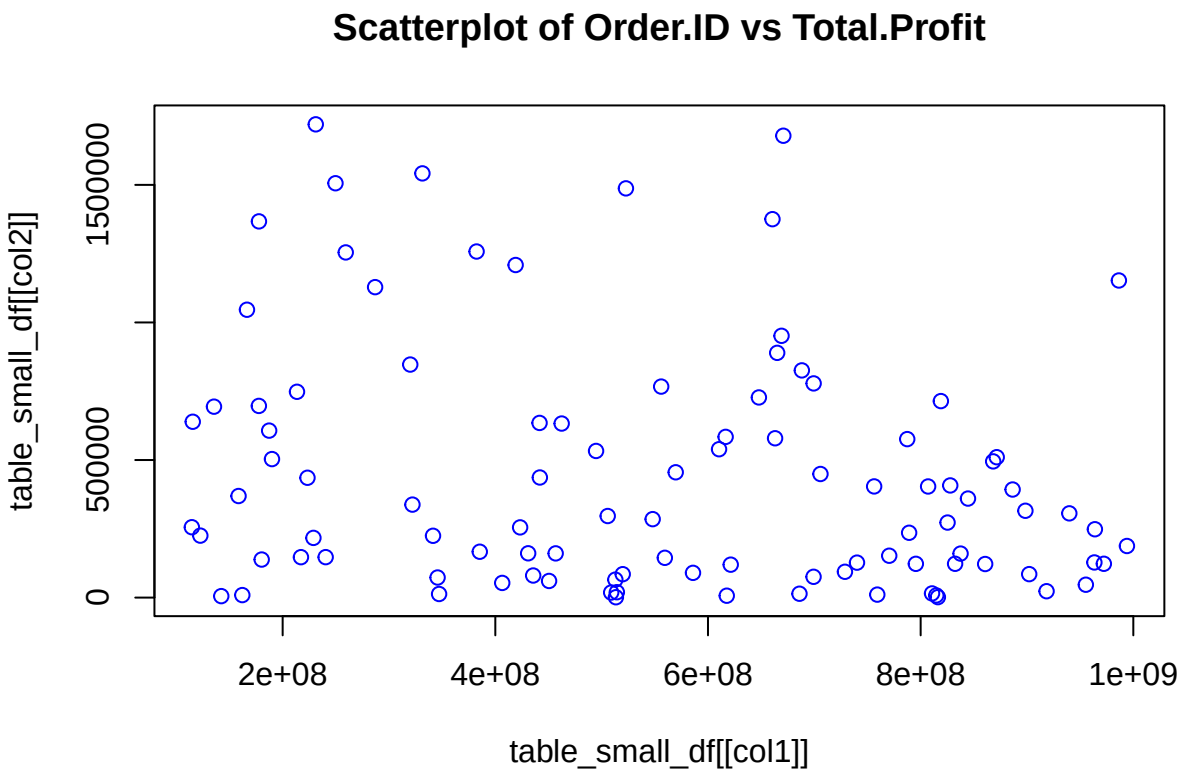
Individual Scatter Plots to explore relationships for each pair of variables.

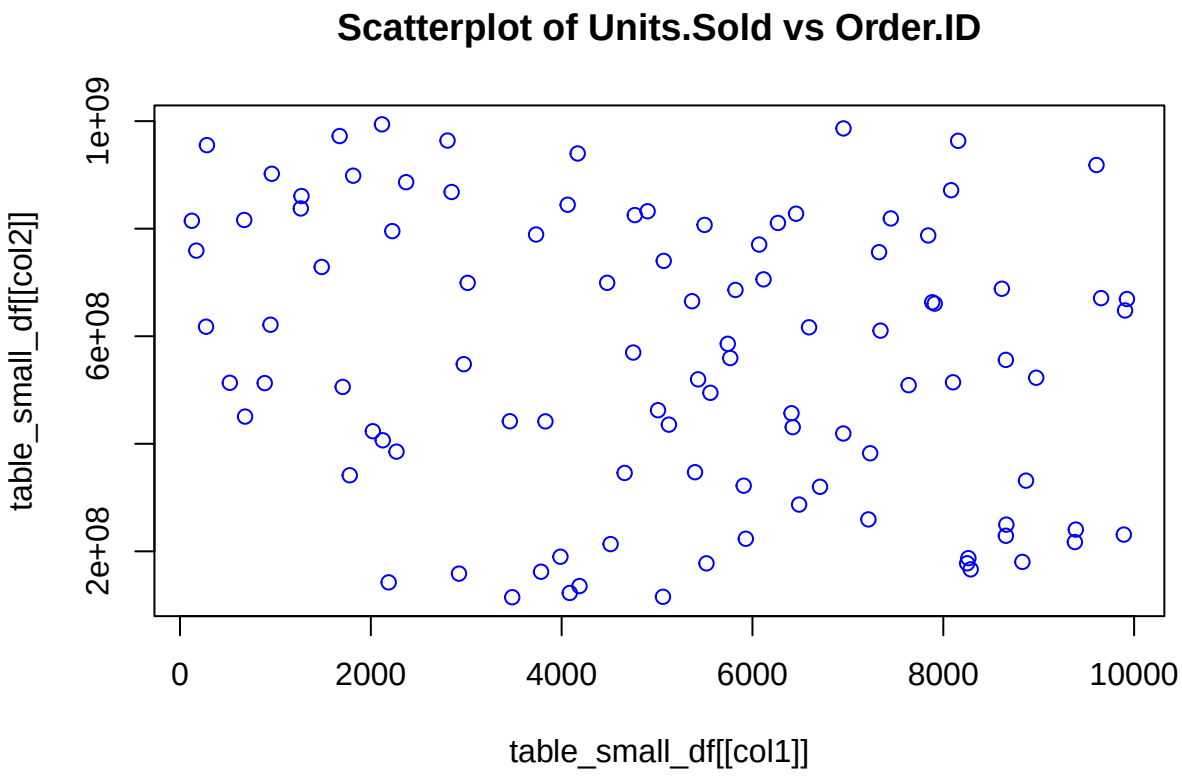


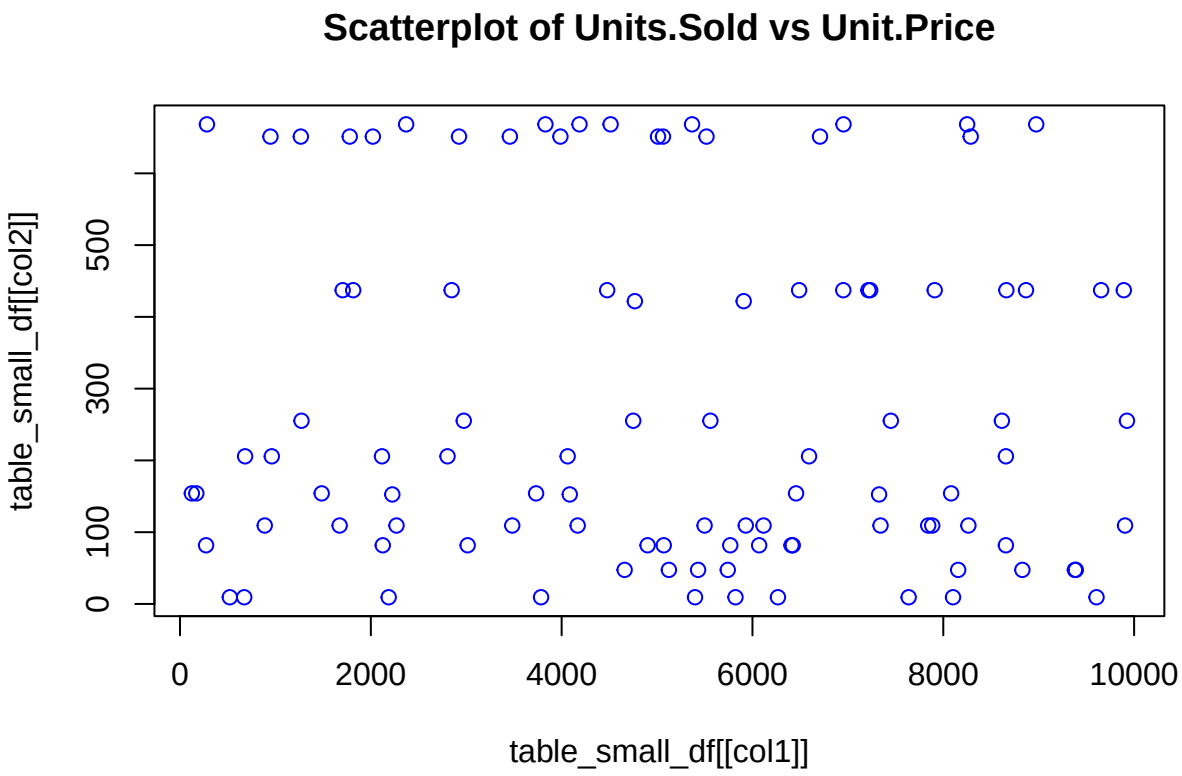

Scatterplot of Order.ID vs Unit.Cost
table_small_df[[col2]]
0
100
200
300
400
500
2e+08
4e+08
6e+08
8e+08
1e+09
table_small_df[[col1]]

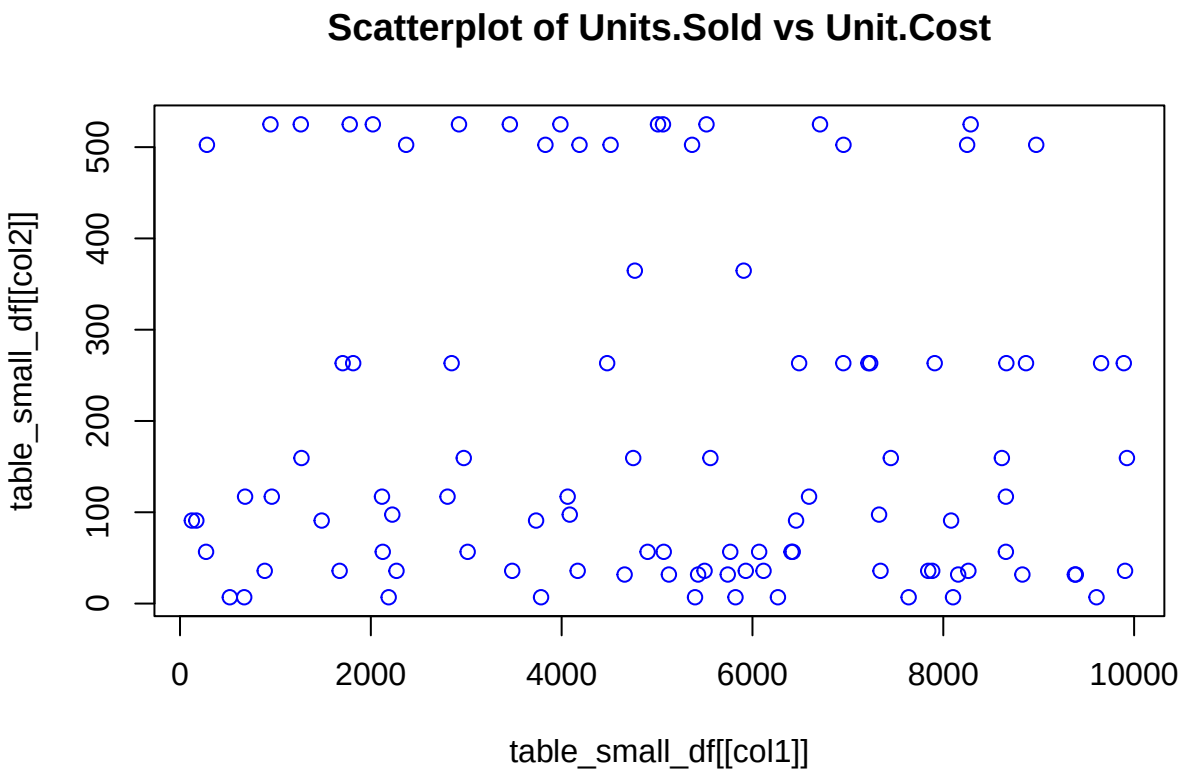

Scatterplot of Order.ID vs Total.Revenue
table_small_df[[col2]]
0e+00
2e+06
4e+06
6e+06
2e+08
4e+08
6e+08
8e+08
1e+09
table_small_df[[col1]]

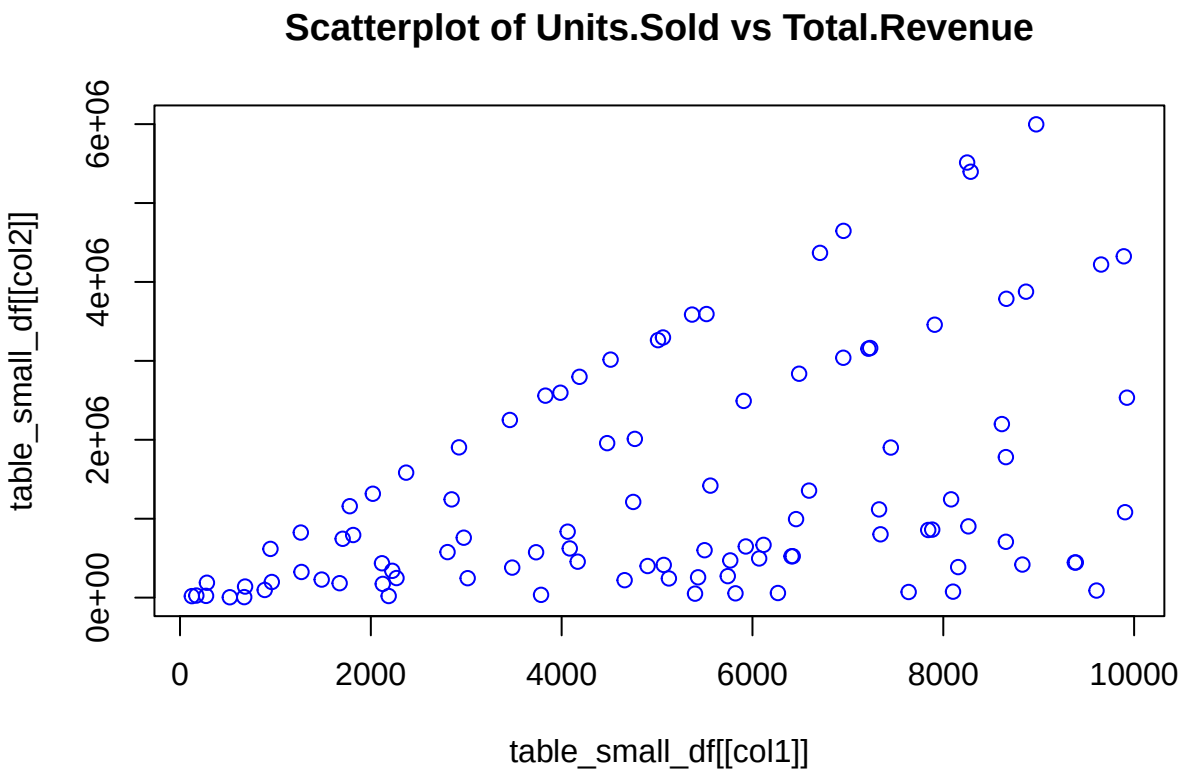

Scatterplot of Order.ID vs Total.Cost
table_small_df[[col2]]
0e+00
2e+06
4e+06
2e+08
4e+08
6e+08
8e+08
1e+09
table_small_df[[col1]]

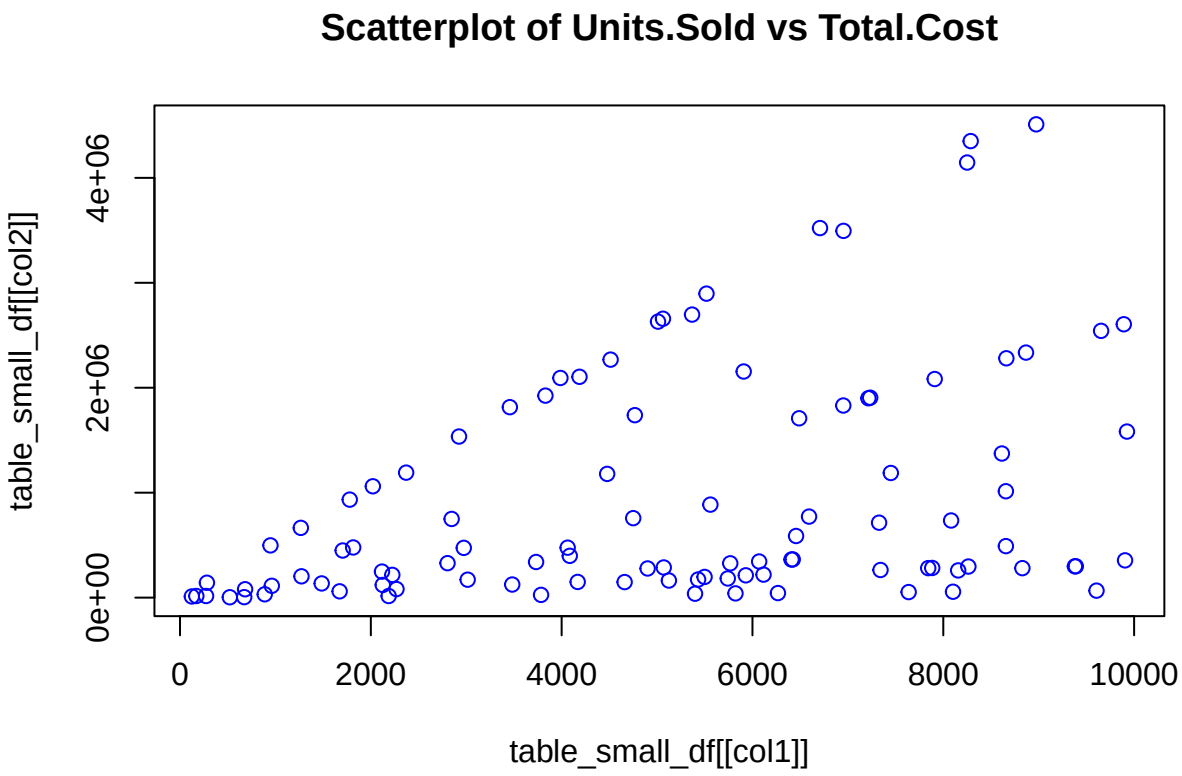

Scatterplot of Order.ID vs Total.Profit
table_small_df[[col2]]
0
500000
1500000
2e+08
4e+08
6e+08
8e+08
1e+09
table_small_df[[col1]]

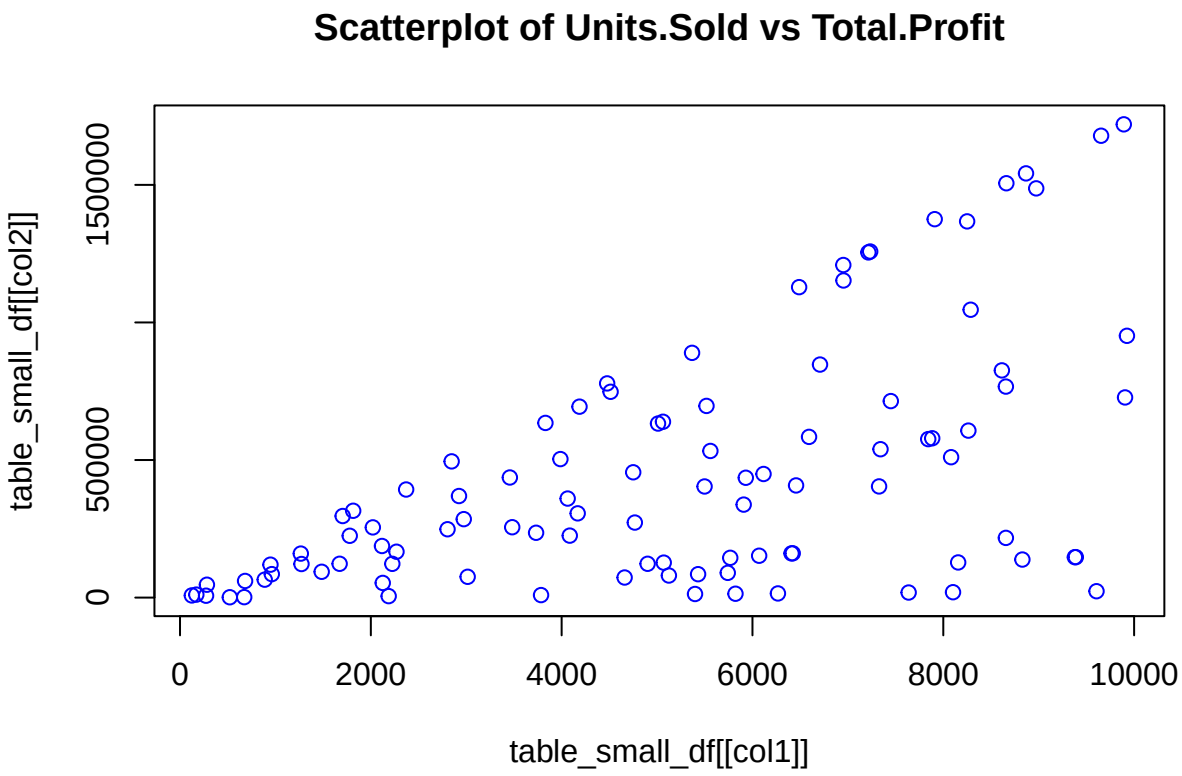

Scatterplot of Units.Sold vs Order.ID
table_small_df[[col2]]
1e+09
6e+08
2e+08
0
2000
4000
6000
8000
10000
table_small_df[[col1]]

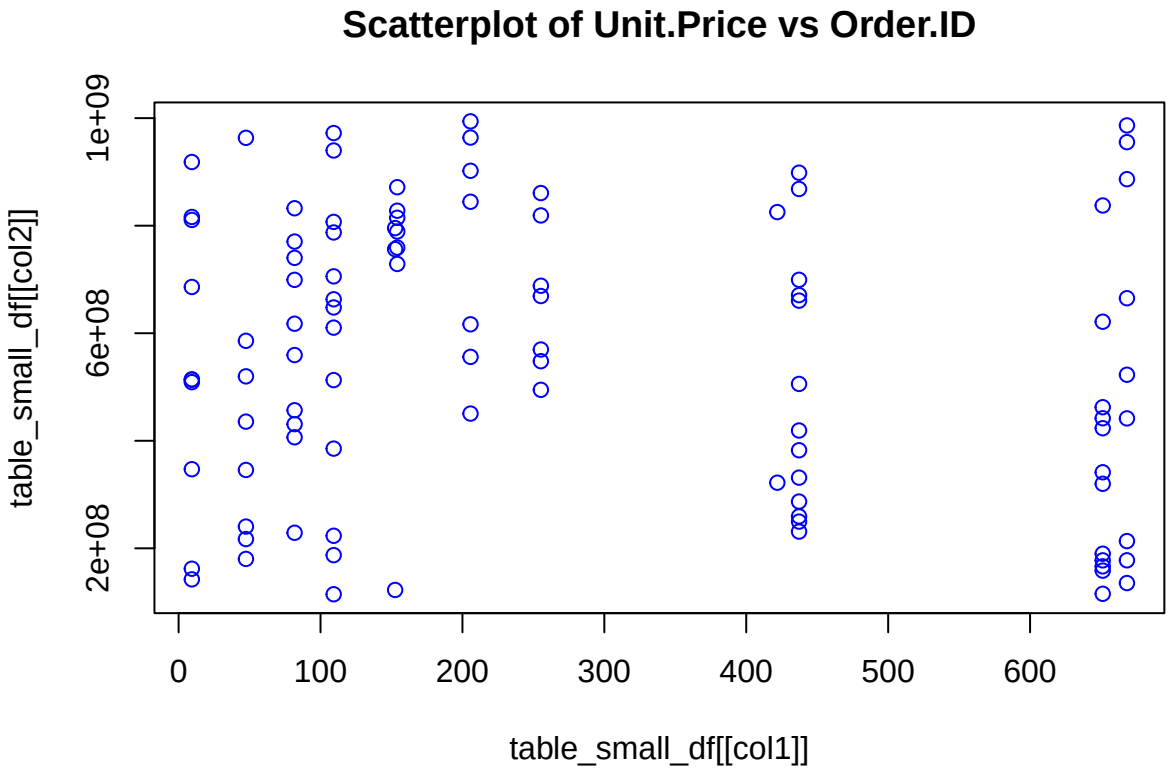

Scatterplot of Units.Sold vs Unit.Price
table_small_df[[col2]]
0
100
300
500
0
2000
4000
6000
8000
10000
table_small_df[[col1]]

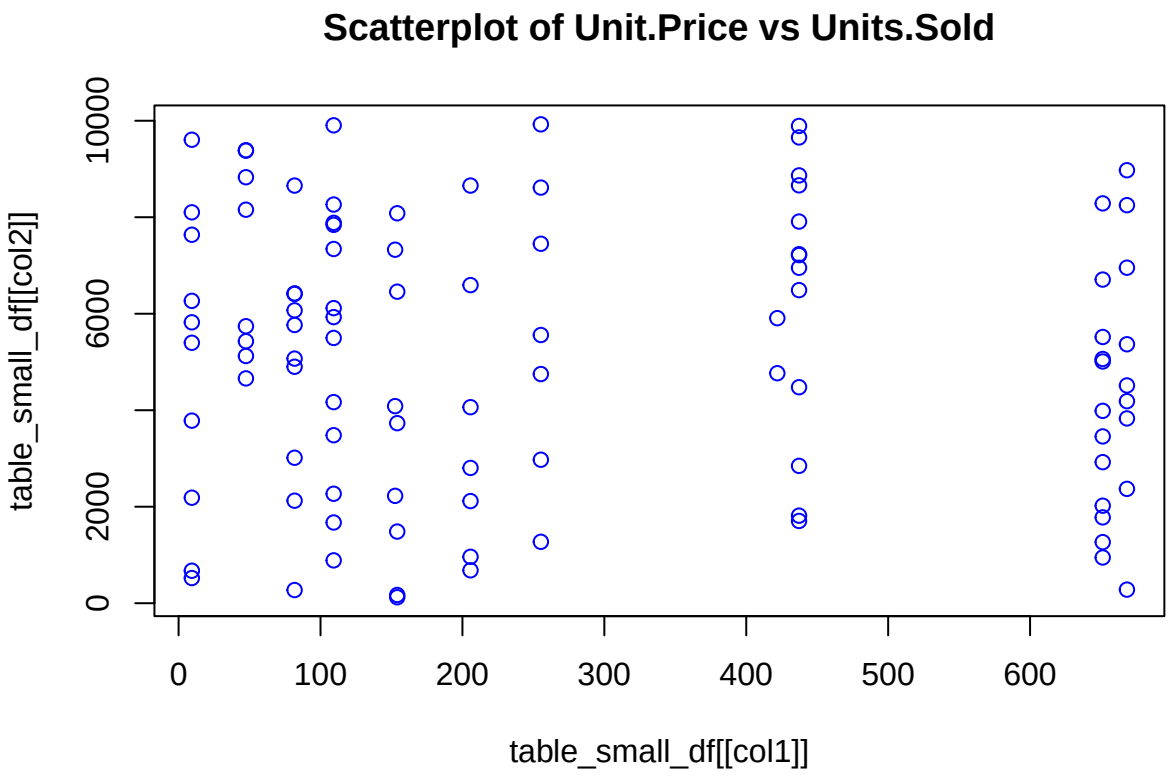

Scatterplot of Units.Sold vs Unit.Cost
table_small_df[[col2]]
0
100
200
300
400
500
0
2000
4000
6000
8000
10000
table_small_df[[col1]]

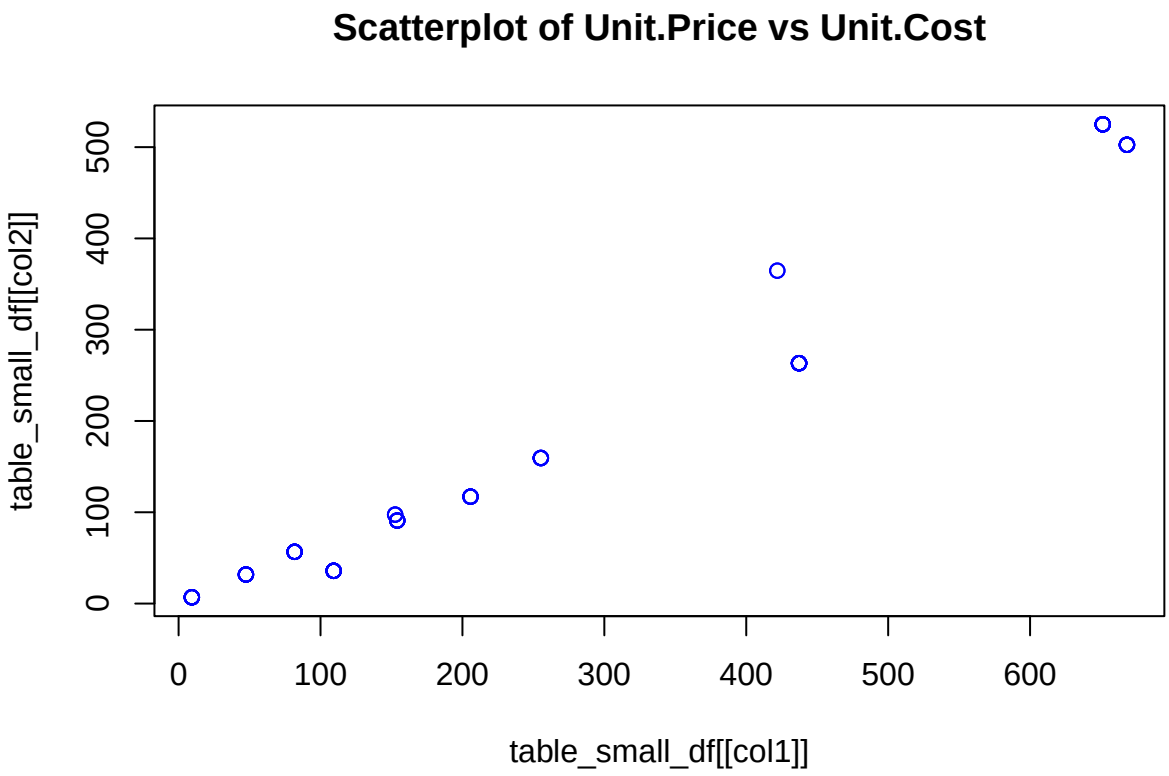

Scatterplot of Units.Sold vs Total.Revenue
table_small_df[[col2]]
0e+00
2e+06
4e+06
6e+06
0
2000
4000
6000
8000
10000
table_small_df[[col1]]

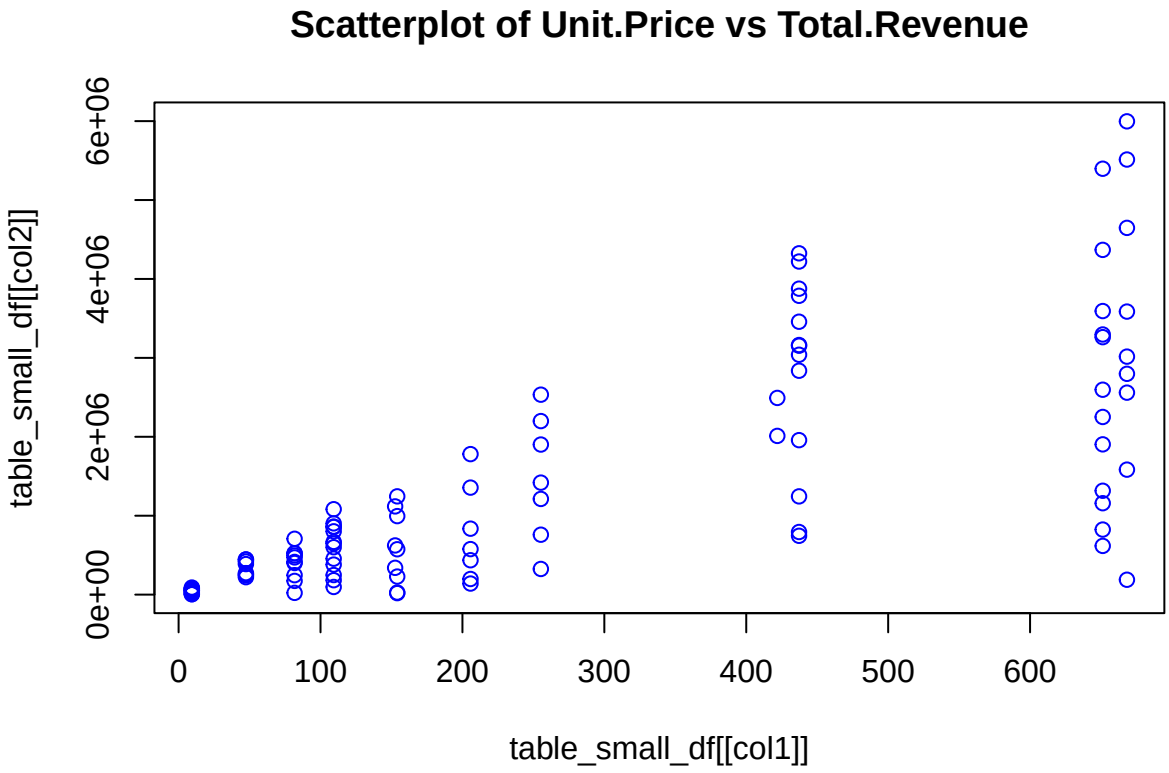

Scatterplot of Units.Sold vs Total.Cost
table_small_df[[col2]]
0e+00
2e+06
4e+06
0
2000
4000
6000
8000
10000
table_small_df[[col1]]

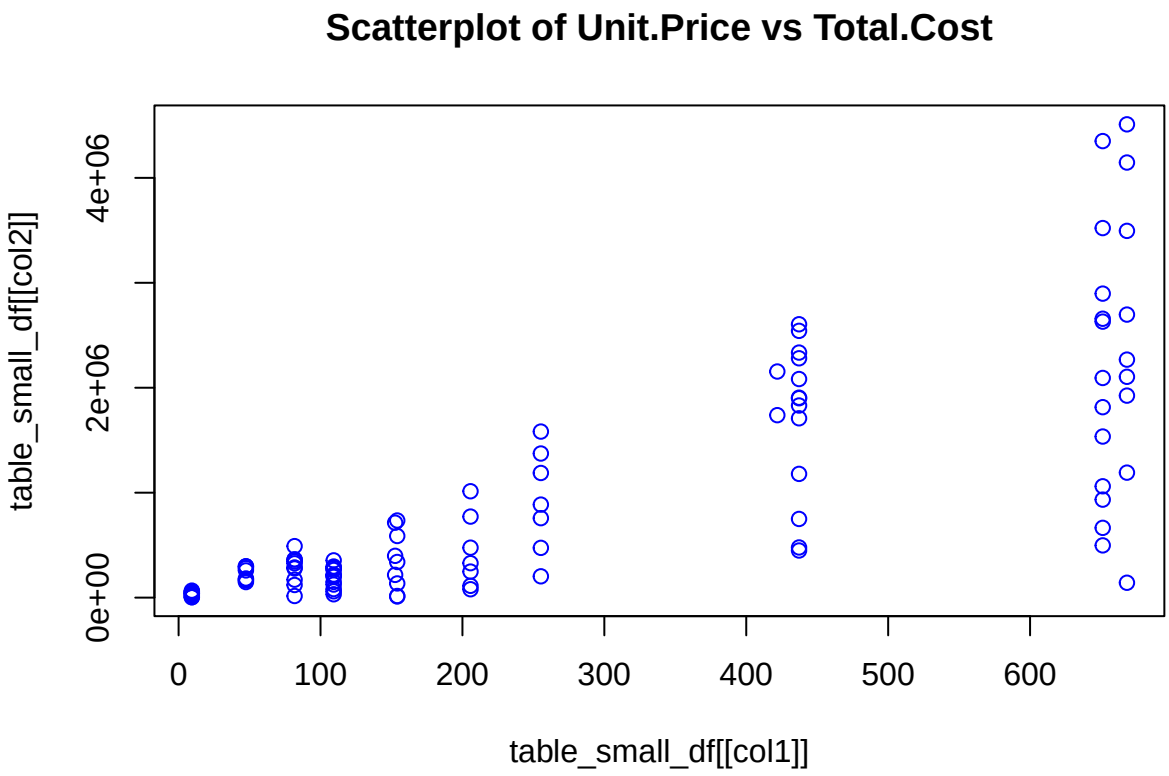

Scatterplot of Units.Sold vs Total.Profit
table_small_df[[col2]]
0
500000
1500000
0
2000
4000
6000
8000
10000
table_small_df[[col1]]

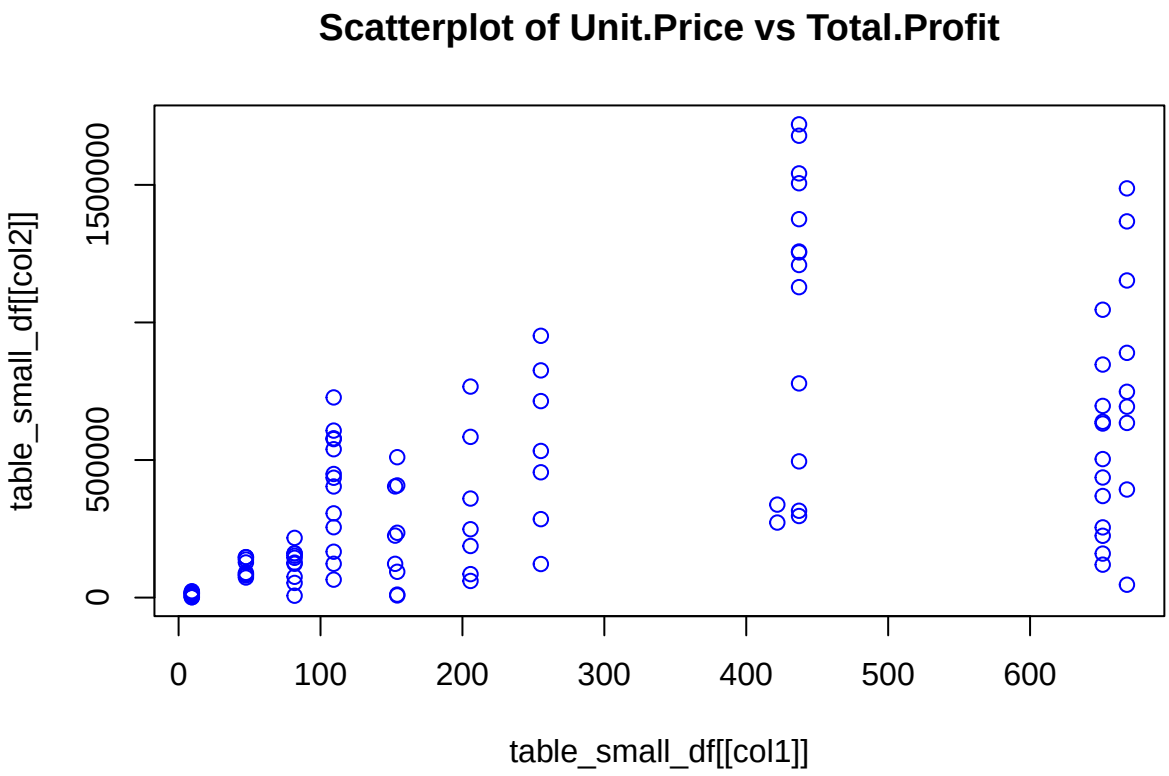

Scatterplot of Unit.Price vs Order.ID
table_small_df[[col2]]
2e+08
6e+08
1e+09
0
100
200
300
400
500
600
table_small_df[[col1]]

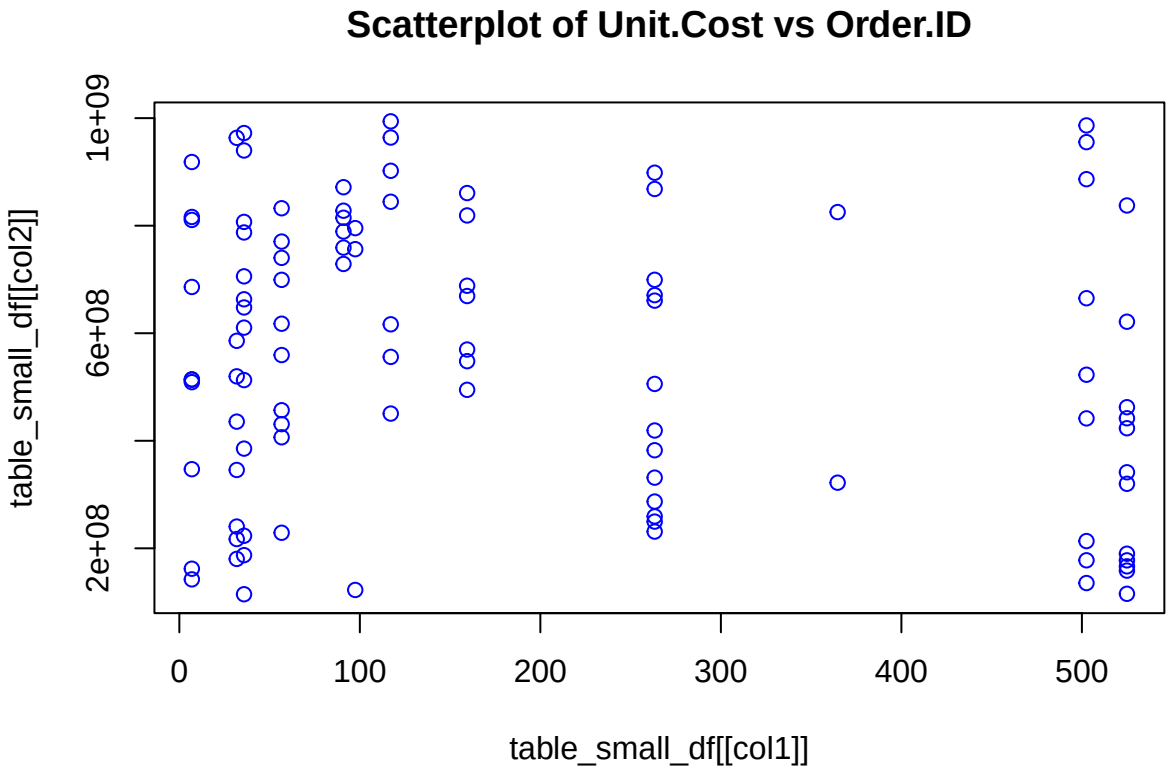

Scatterplot of Unit.Price vs Units.Sold
table_small_df[[col2]]
0
2000
6000
10000
0
100
200
300
400
500
600
table_small_df[[col1]]

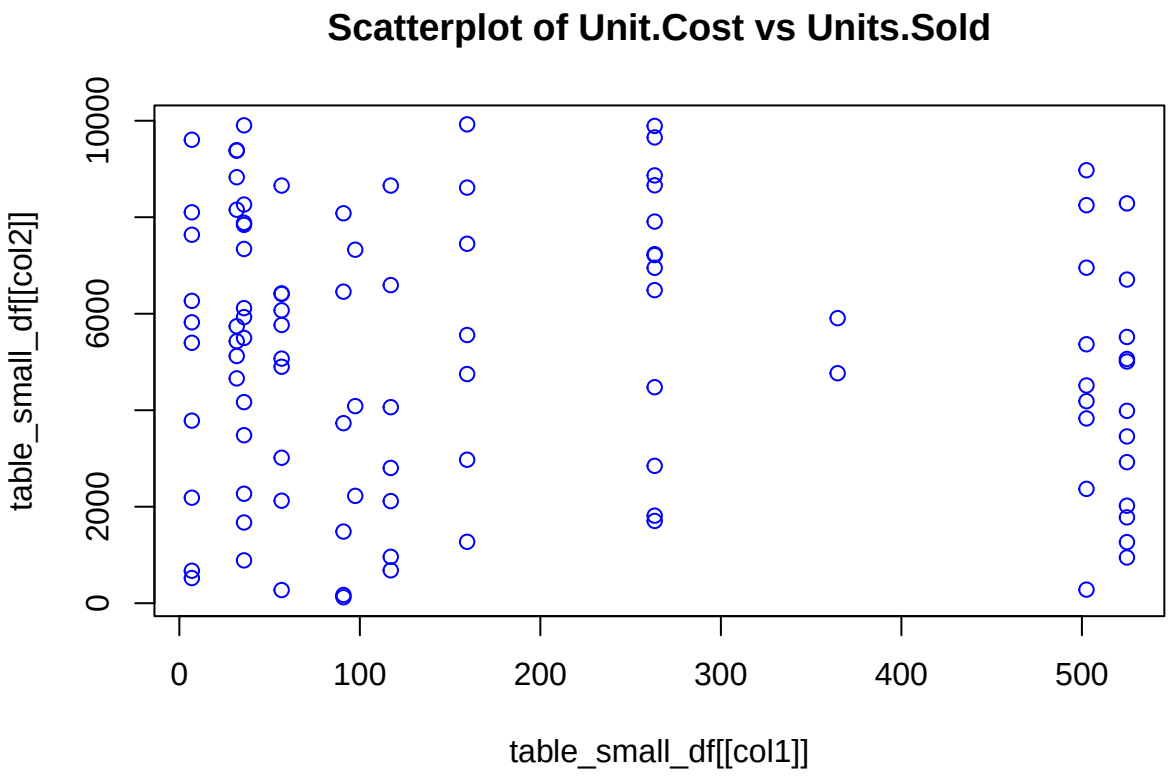

Scatterplot of Unit.Price vs Unit.Cost
table_small_df[[col2]]
0
100
200
300
400
500
table_small_df[[col1]]
0
100
200
300
400
500
600

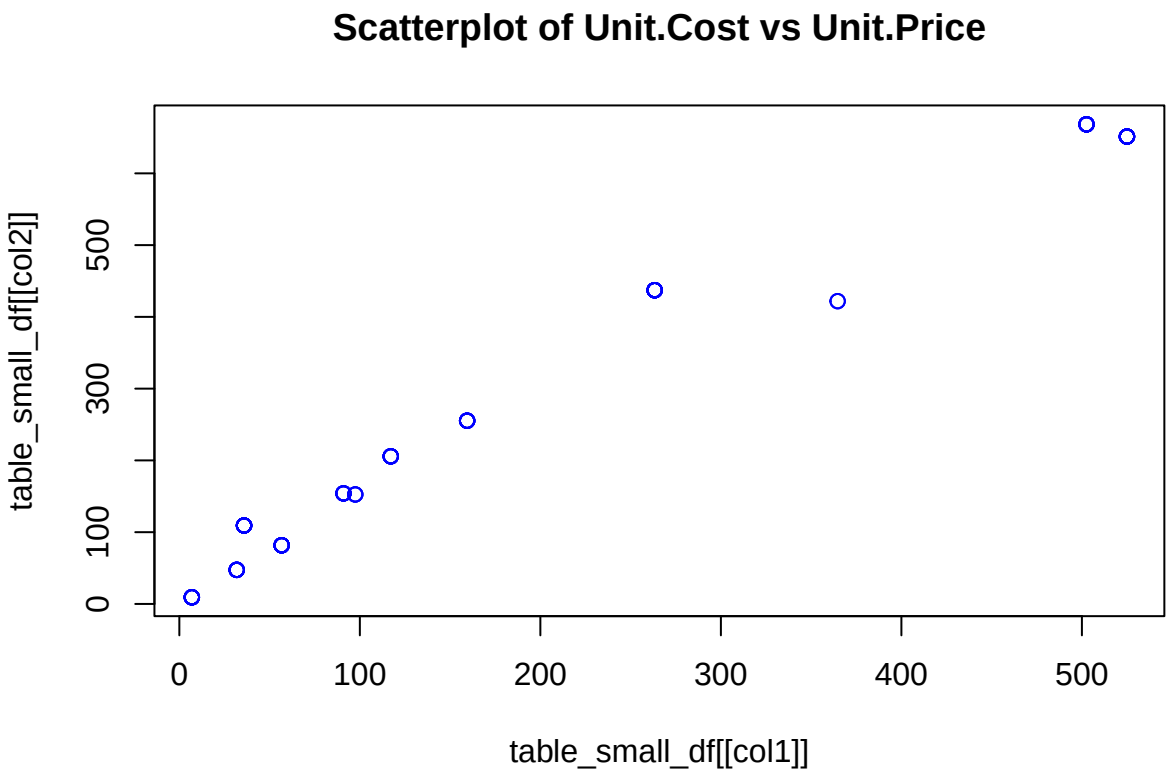

Scatterplot of Unit.Price vs Total.Revenue
table_small_df[[col2]]
0e+00
2e+06
4e+06
6e+06
0
100
200
300
400
500
600
table_small_df[[col1]]

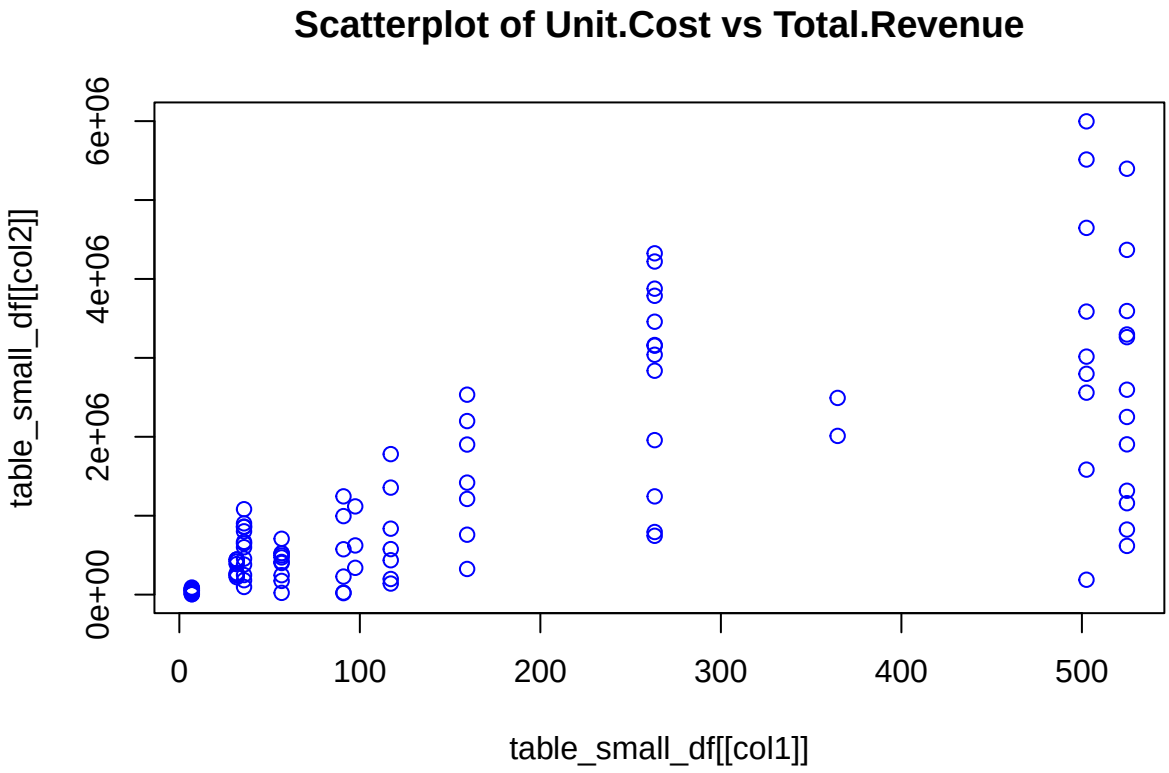

Scatterplot of Unit.Price vs Total.Cost
table_small_df[[col2]]
0e+00
2e+06
4e+06
0
100
200
300
400
500
600
table_small_df[[col1]]

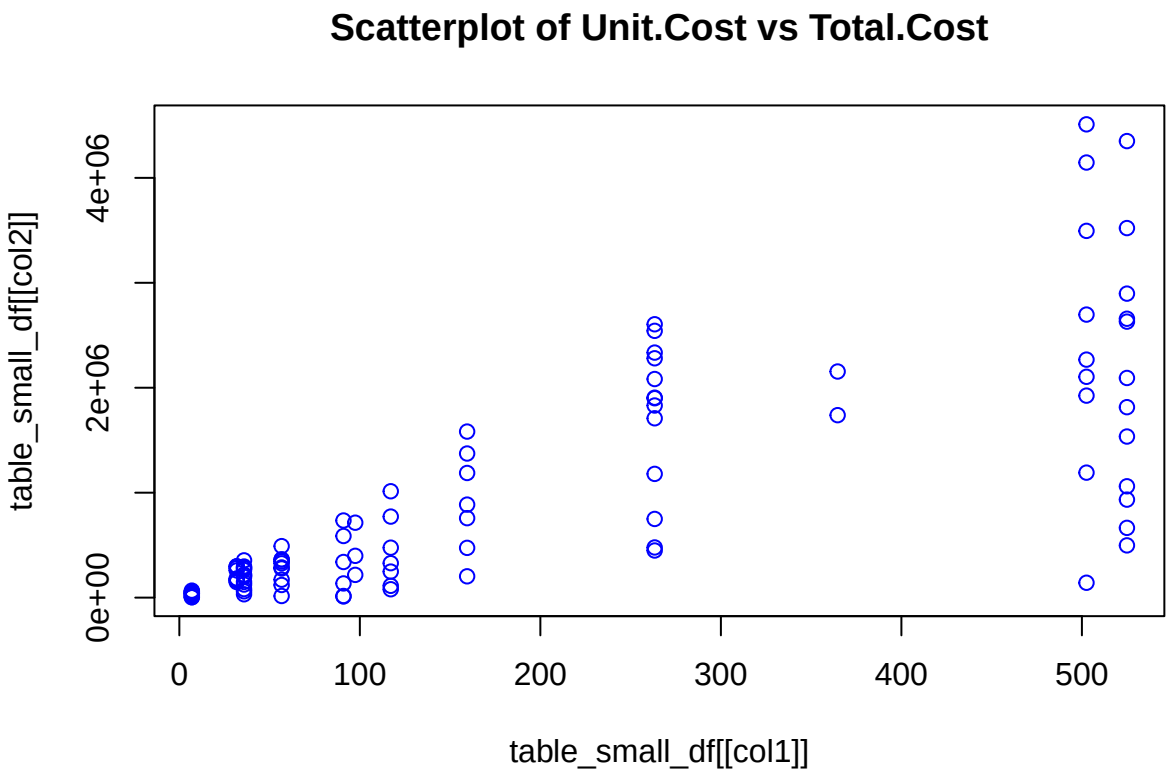

Scatterplot of Unit.Price vs Total.Profit
table_small_df[[col2]]
0
500000
1500000
0
100
200
300
400
500
600
table_small_df[[col1]]

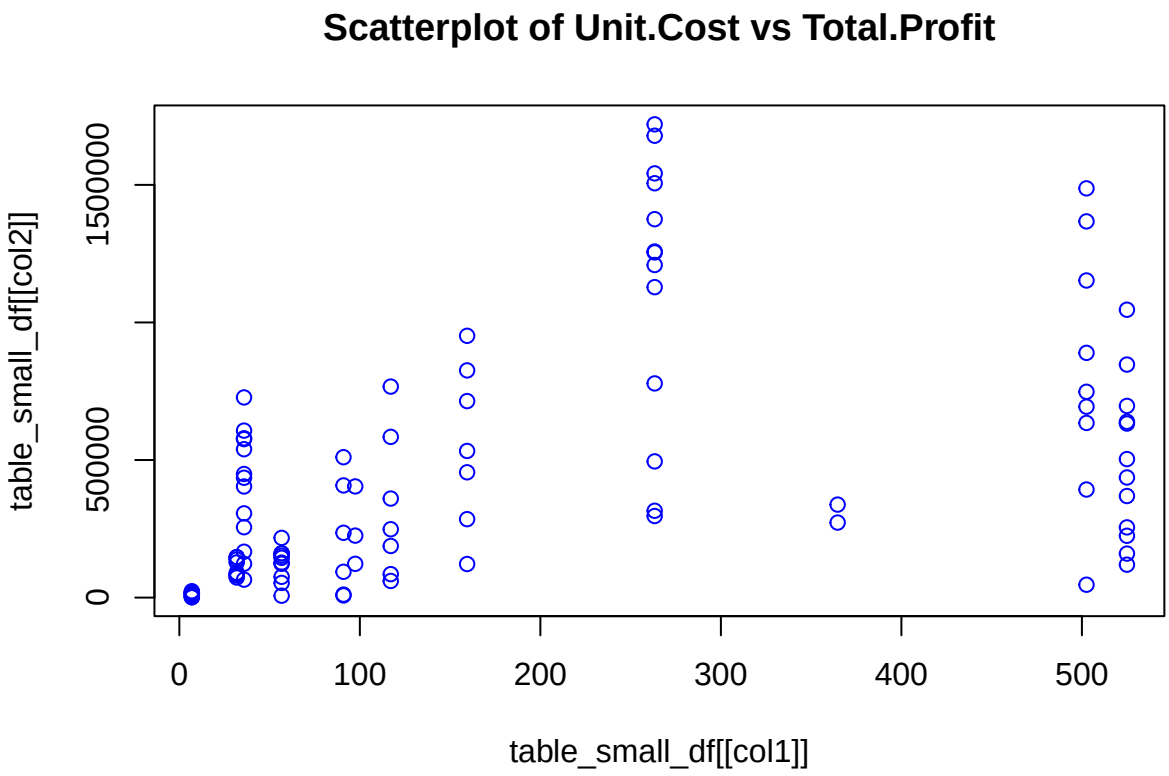

Scatterplot of Unit.Cost vs Order.ID
table_small_df[[col2]]
1e+09
6e+08
2e+08
0
100
200
300
400
500
table_small_df[[col1]]

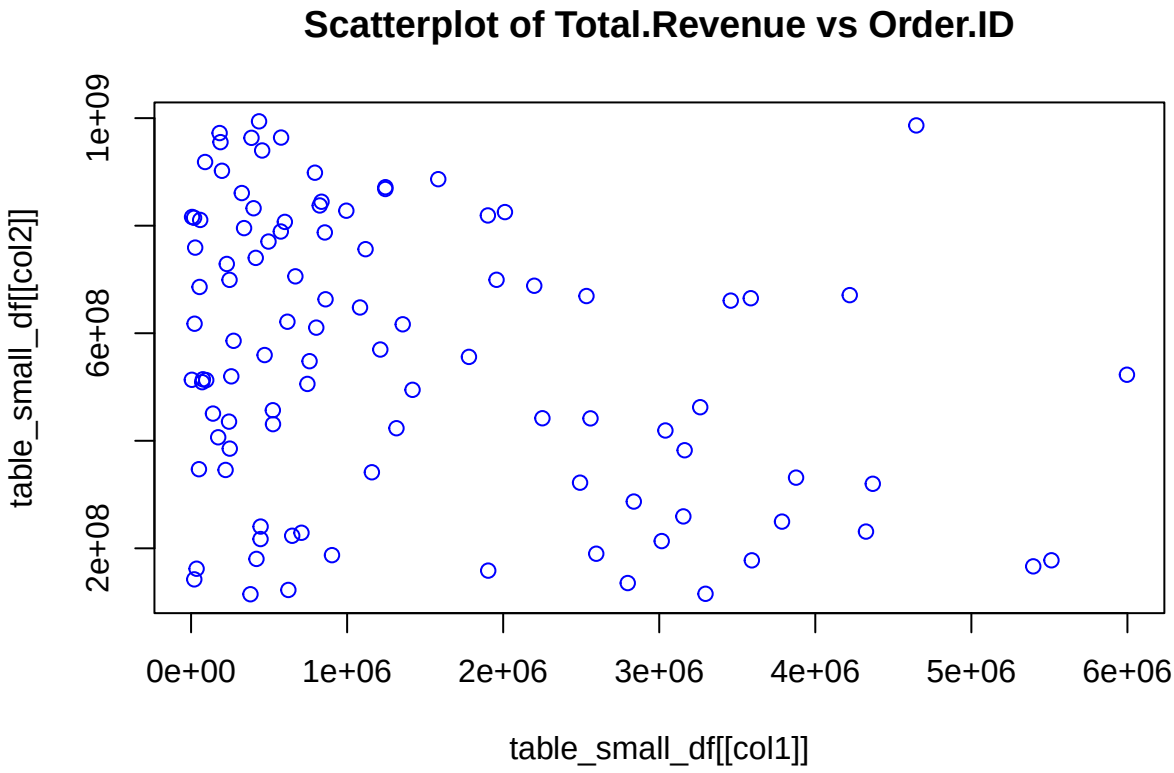

Scatterplot of Unit.Cost vs Units.Sold
table_small_df[[col2]]
0
2000
6000
10000
0
100
200
300
400
500
table_small_df[[col1]]

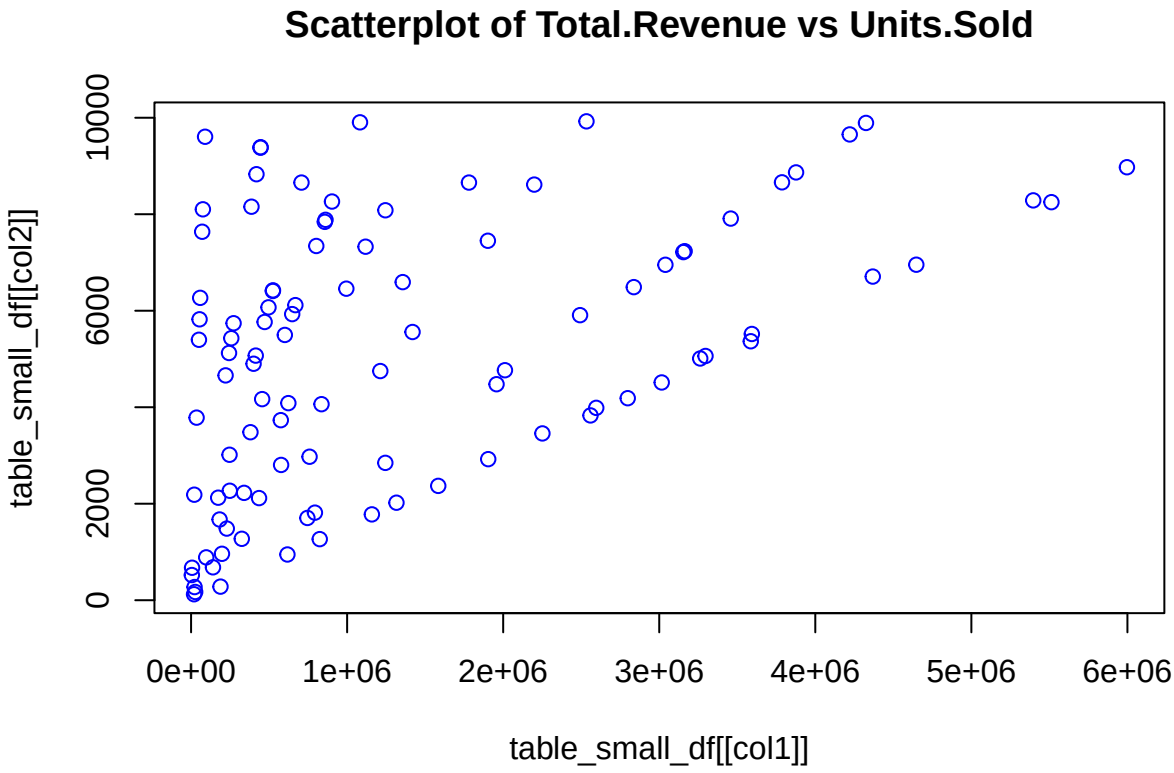

Scatterplot of Unit.Cost vs Unit.Price
table_small_df[[col2]]
0
100
300
500
0
100
200
300
400
500
table_small_df[[col1]]

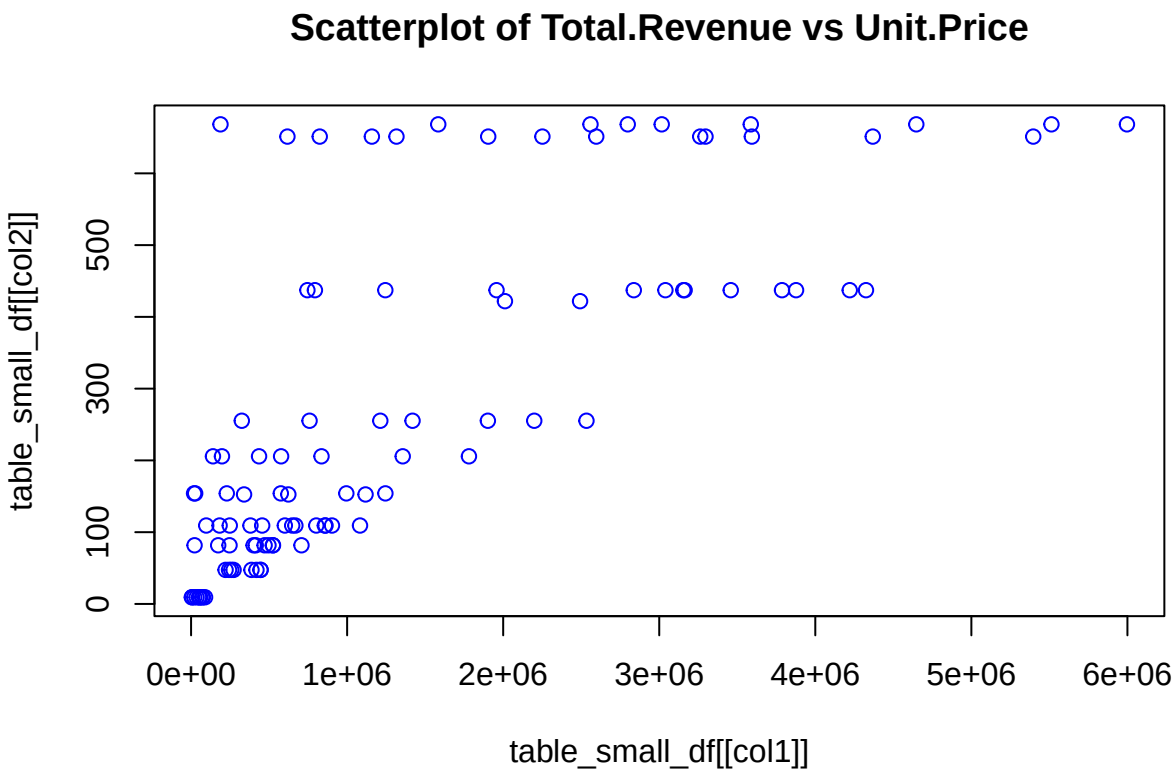

Scatterplot of Unit.Cost vs Total.Revenue
table_small_df[[col2]]
0e+00
2e+06
4e+06
6e+06
0
100
200
300
400
500
table_small_df[[col1]]

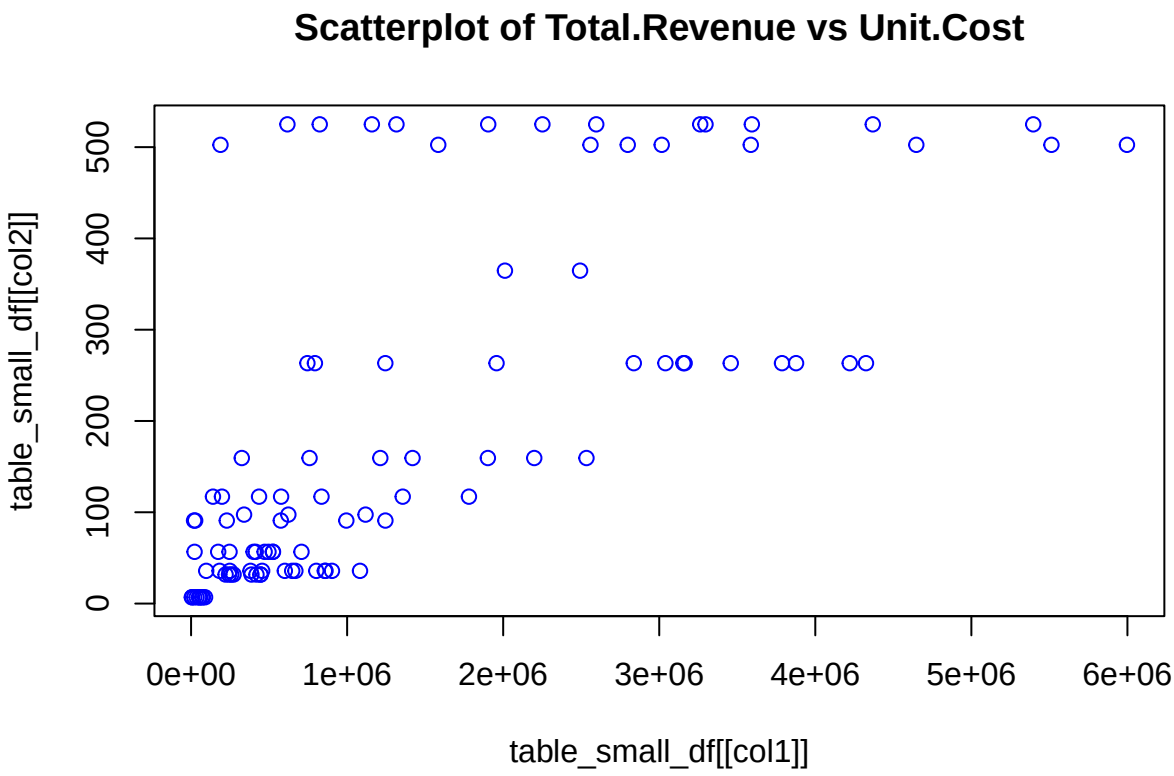

Scatterplot of Unit.Cost vs Total.Cost
table_small_df[[col2]]
0e+00
2e+06
4e+06
0
100
200
300
400
500
table_small_df[[col1]]

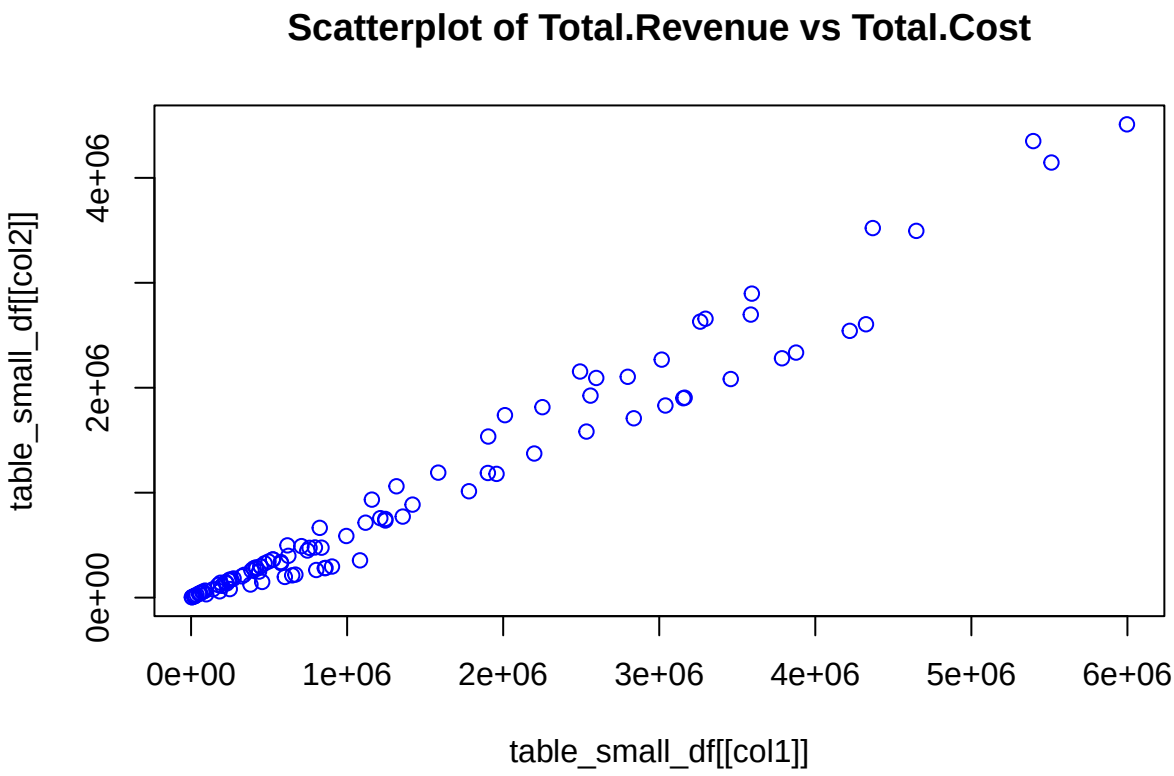

Scatterplot of Unit.Cost vs Total.Profit
table_small_df[[col2]]
0
500000
1500000
0
100
200
300
400
500
table_small_df[[col1]]

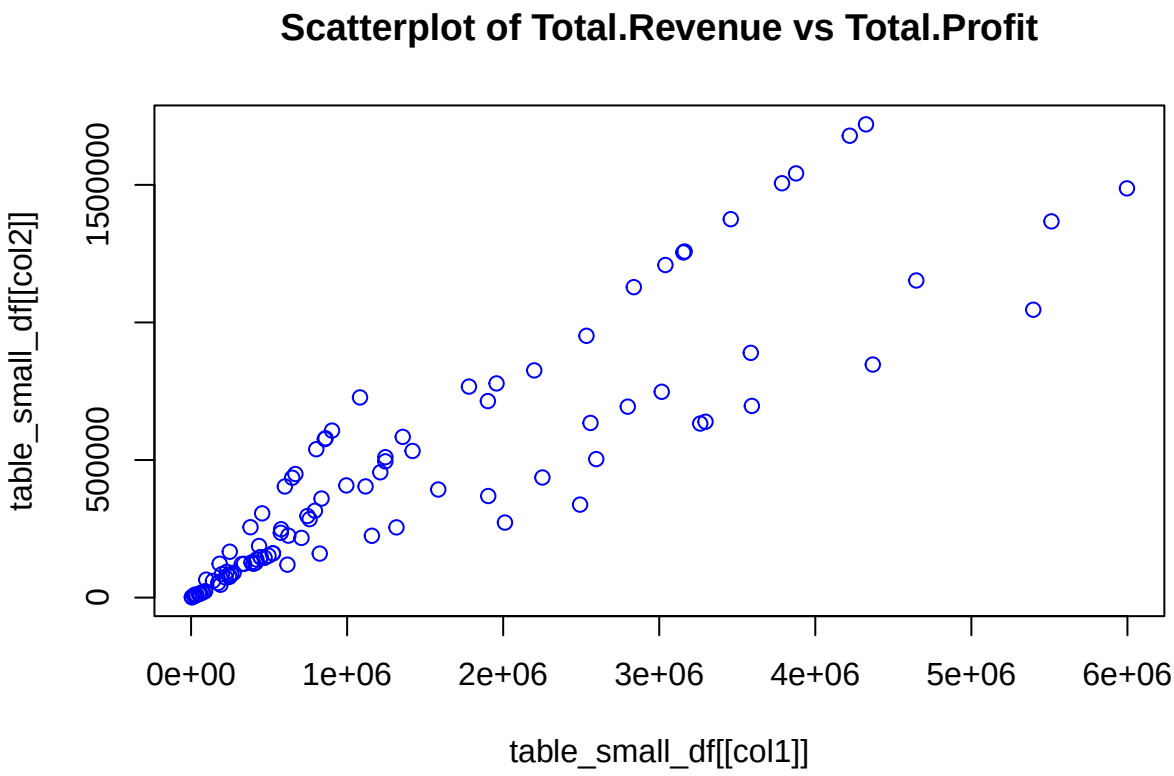

Scatterplot of Total.Revenue vs Order.ID
table_small_df[[col2]]
1e+09
6e+08
2e+08
0e+00
1e+06
2e+06
3e+06
4e+06
5e+06
6e+06
table_small_df[[col1]]

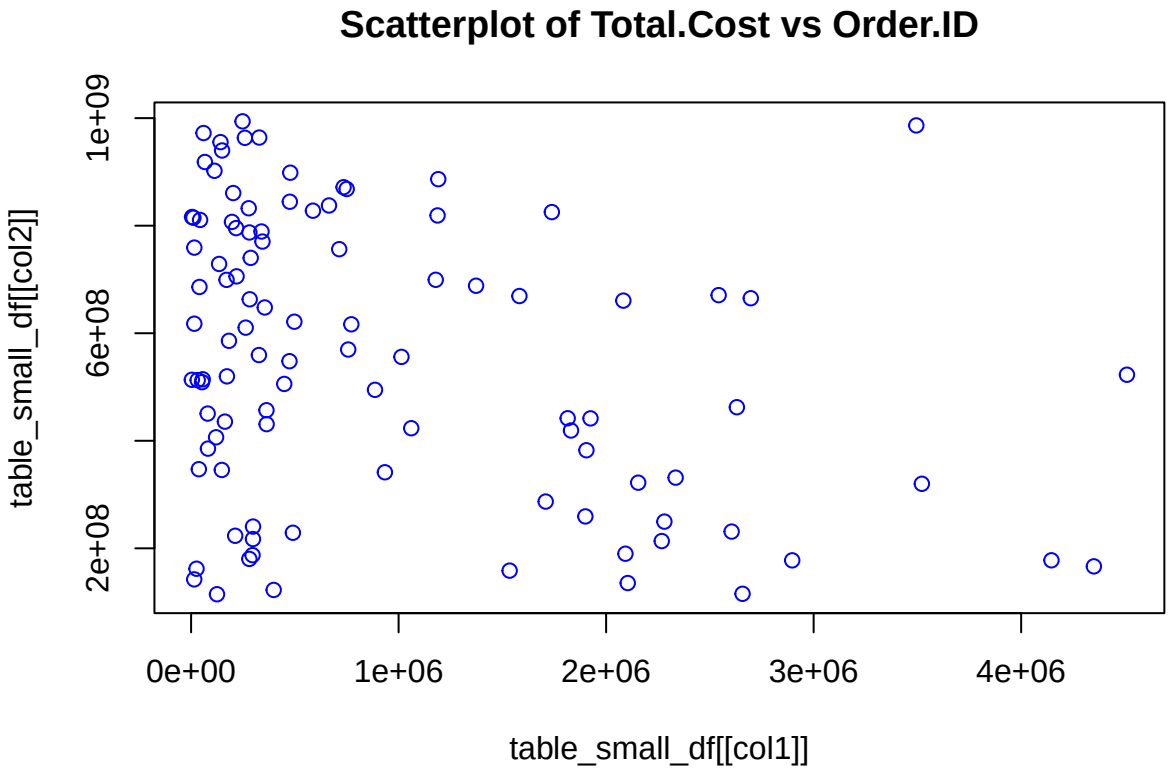

Scatterplot of Total.Revenue vs Units.Sold
table_small_df[[col2]]
0
2000
6000
10000
0e+00
1e+06
2e+06
3e+06
4e+06
5e+06
6e+06
table_small_df[[col1]]

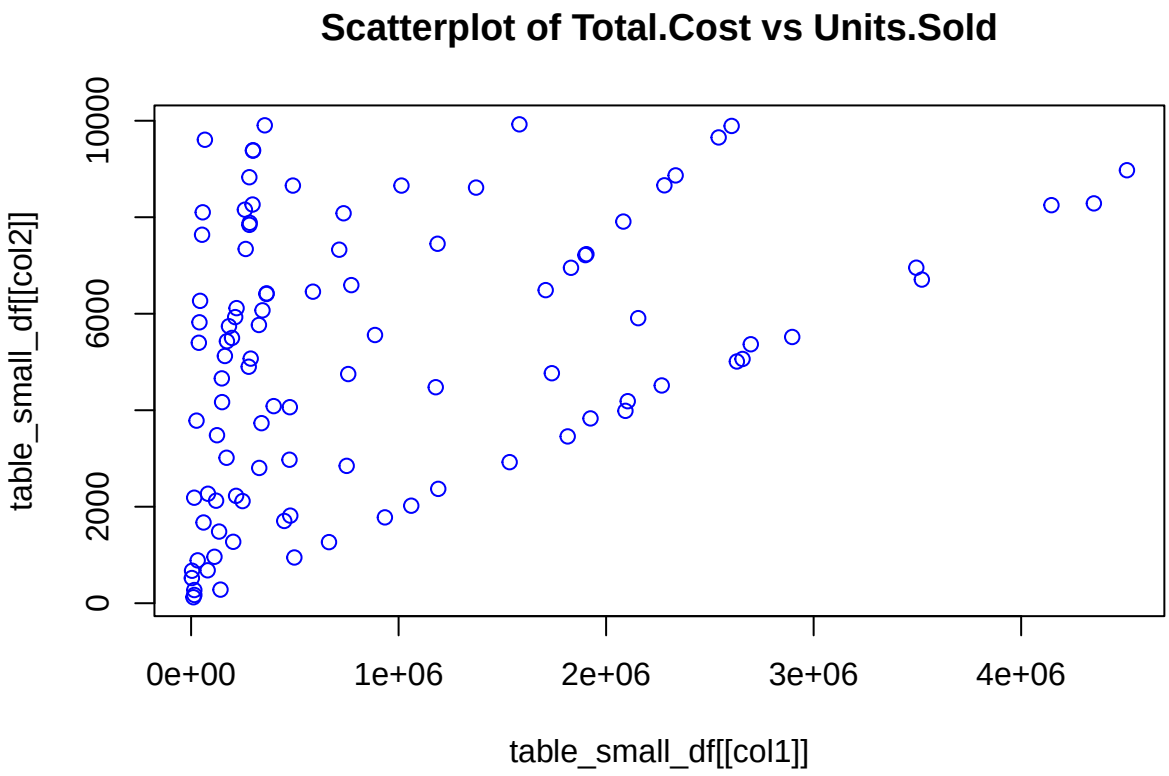

Scatterplot of Total.Revenue vs Unit.Price
table_small_df[[col2]]
0
100
300
500
0e+00
1e+06
2e+06
3e+06
4e+06
5e+06
6e+06
table_small_df[[col1]]

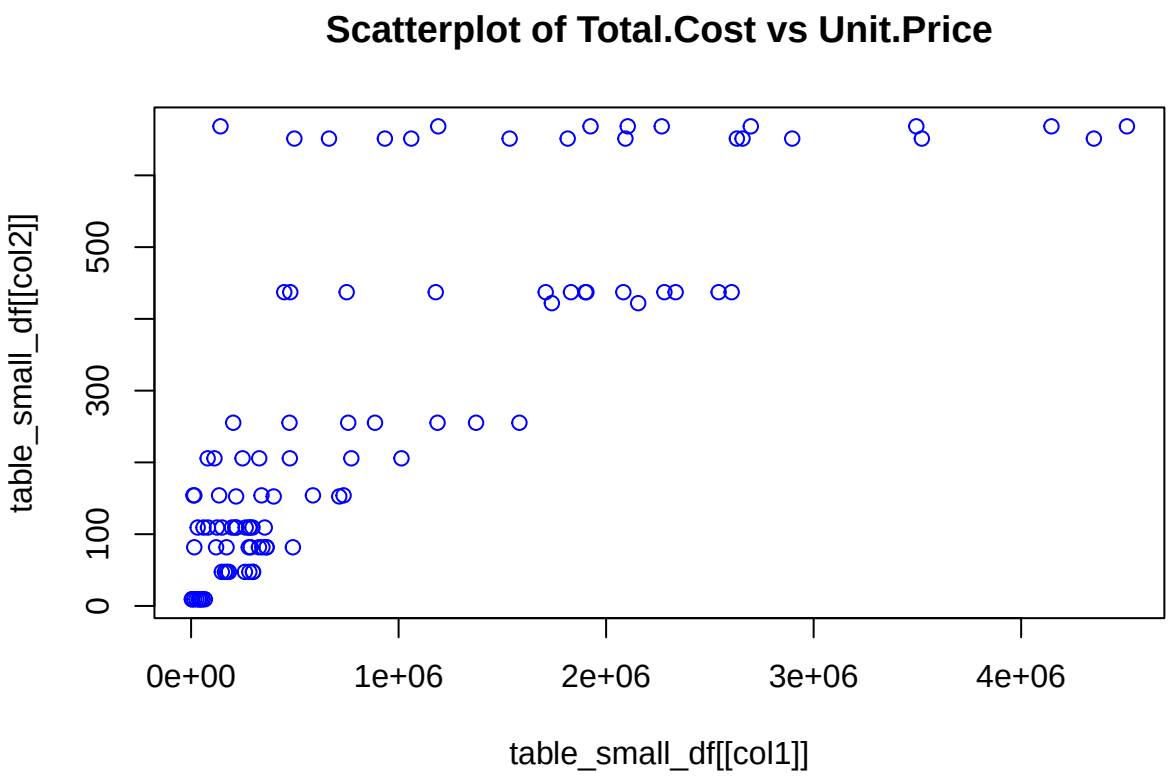

Scatterplot of Total.Revenue vs Unit.Cost
table_small_df[[col2]]
0
100
200
300
400
500
0e+00
1e+06
2e+06
3e+06
4e+06
5e+06
6e+06
table_small_df[[col1]]

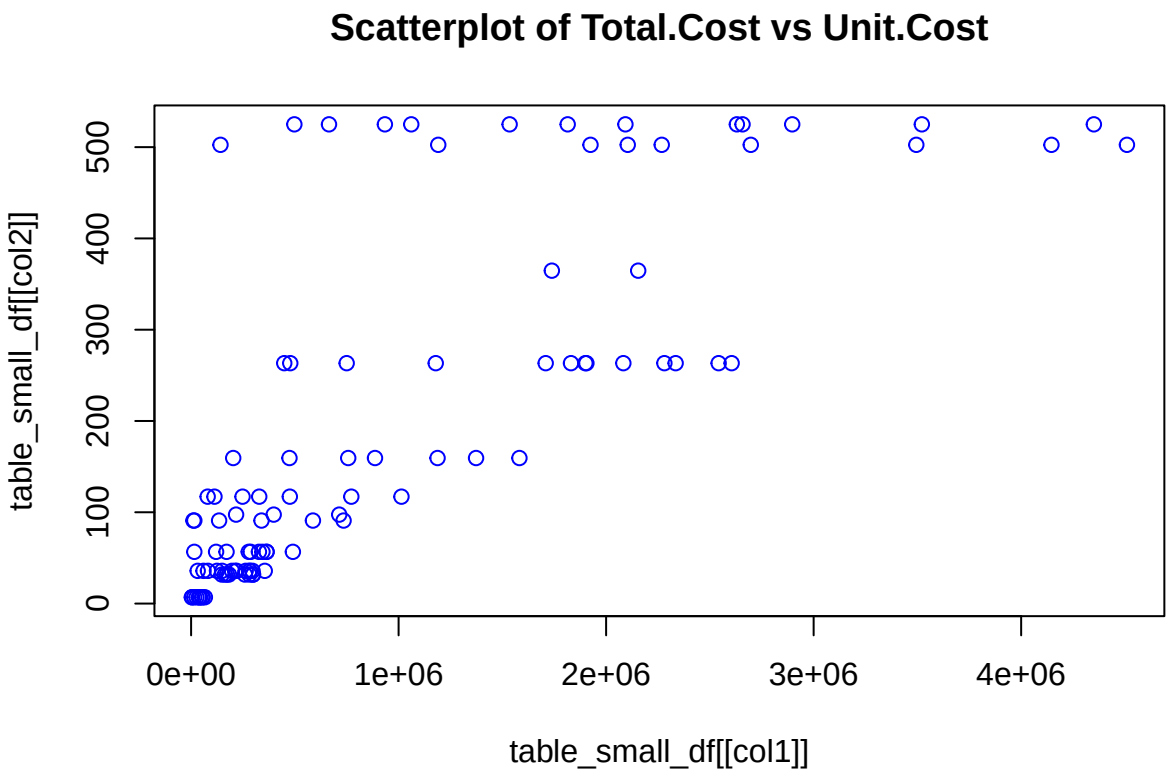

Scatterplot of Total.Revenue vs Total.Cost
table_small_df[[col2]]
0e+00
2e+06
4e+06
0e+00
1e+06
2e+06
3e+06
4e+06
5e+06
6e+06
table_small_df[[col1]]

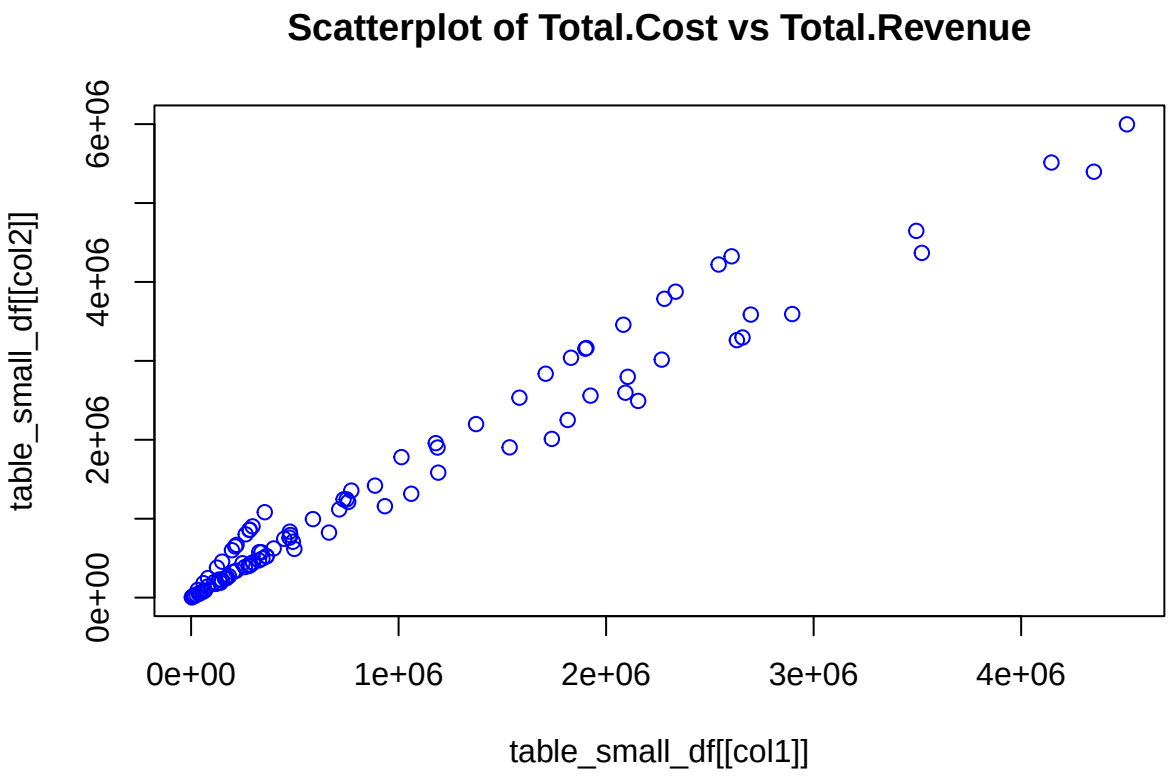

Scatterplot of Total.Revenue vs Total.Profit
table_small_df[[col2]]
0
500000
1500000
0e+00
1e+06
2e+06
3e+06
4e+06
5e+06
6e+06
table_small_df[[col1]]

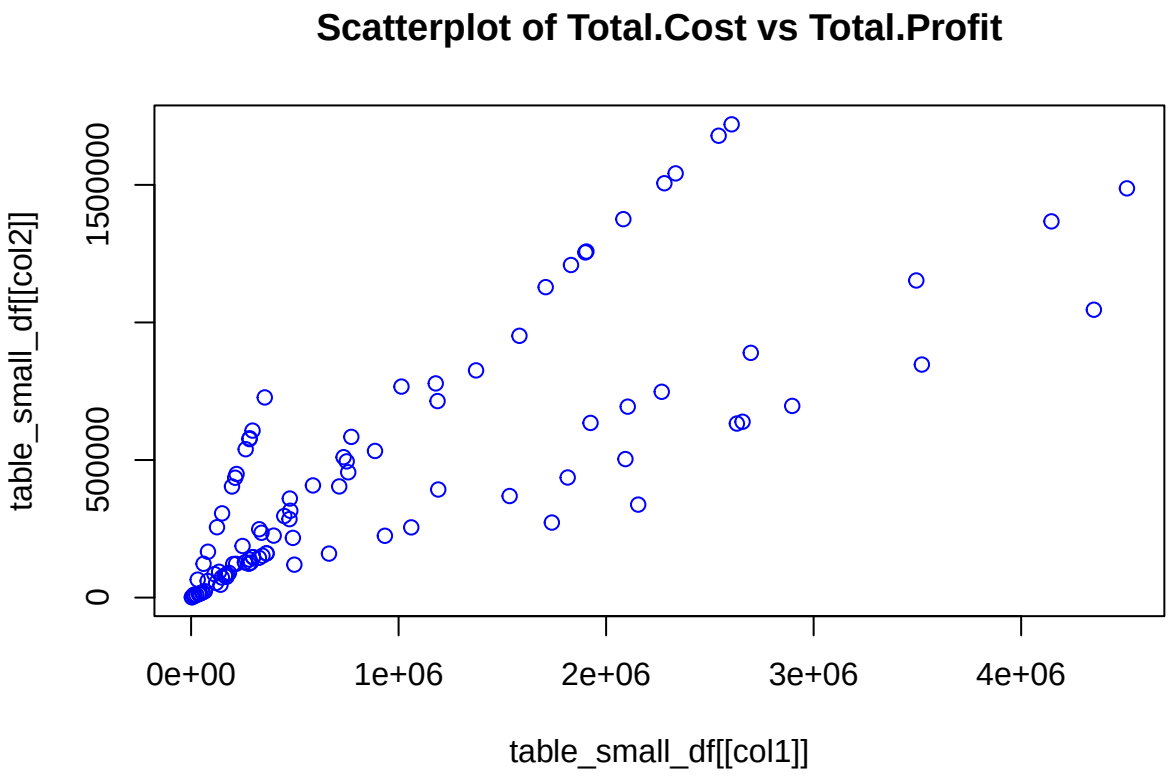

Scatterplot of Total.Cost vs Order.ID
table_small_df[[col2]]
1e+09
6e+08
2e+08
0e+00
1e+06
2e+06
3e+06
4e+06
table_small_df[[col1]]

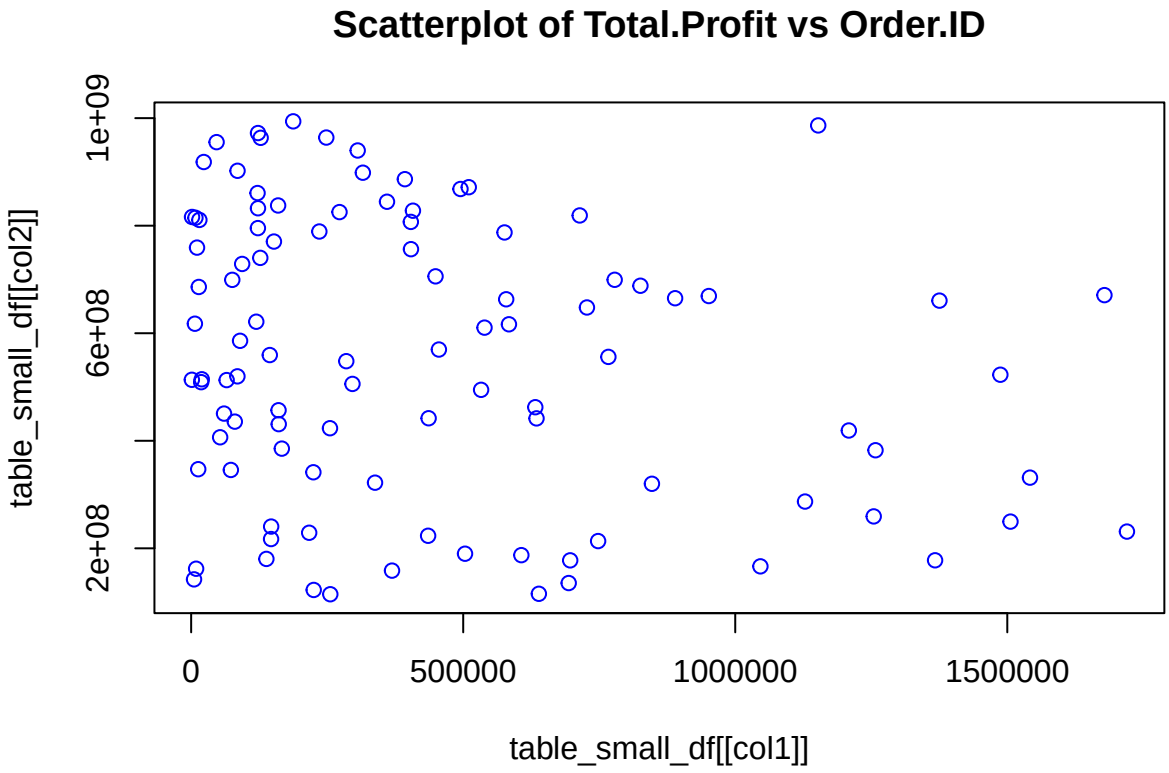

Scatterplot of Total.Cost vs Units.Sold
table_small_df[[col2]]
0
2000
6000
10000
0e+00
1e+06
2e+06
3e+06
4e+06
table_small_df[[col1]]

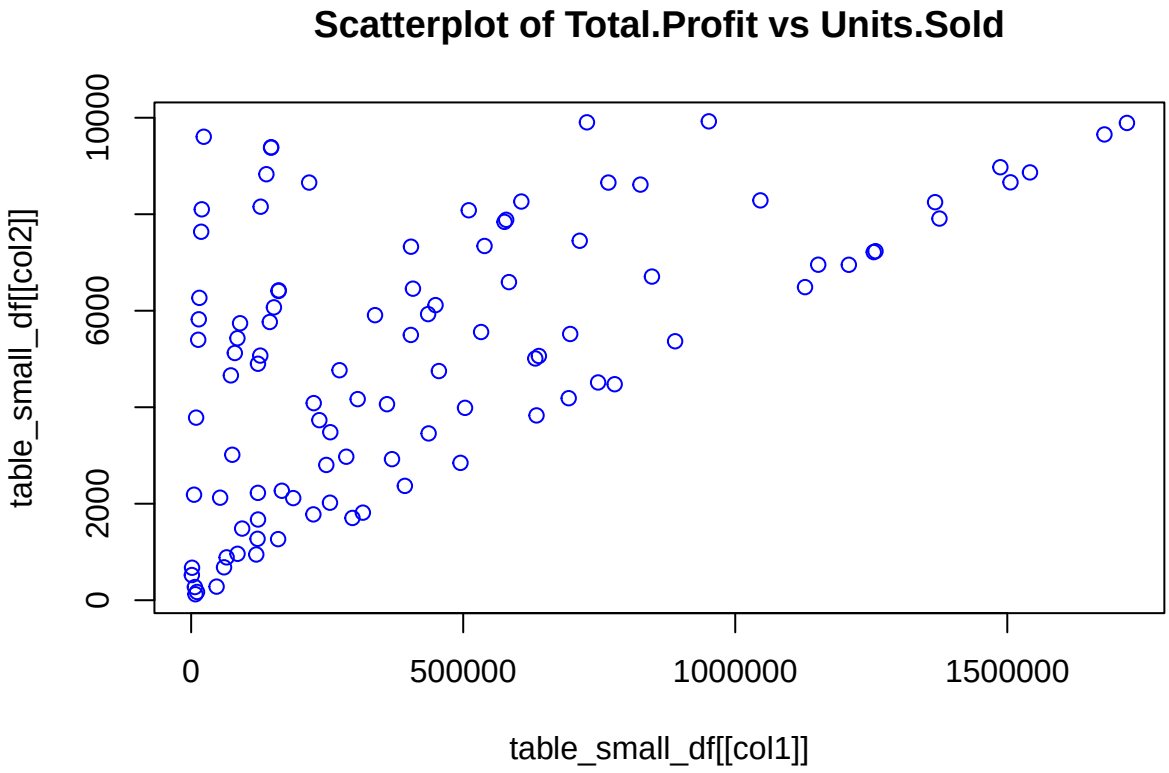

Scatterplot of Total.Cost vs Unit.Price
table_small_df[[col2]]
0
100
300
500
0e+00
1e+06
2e+06
3e+06
4e+06
table_small_df[[col1]]

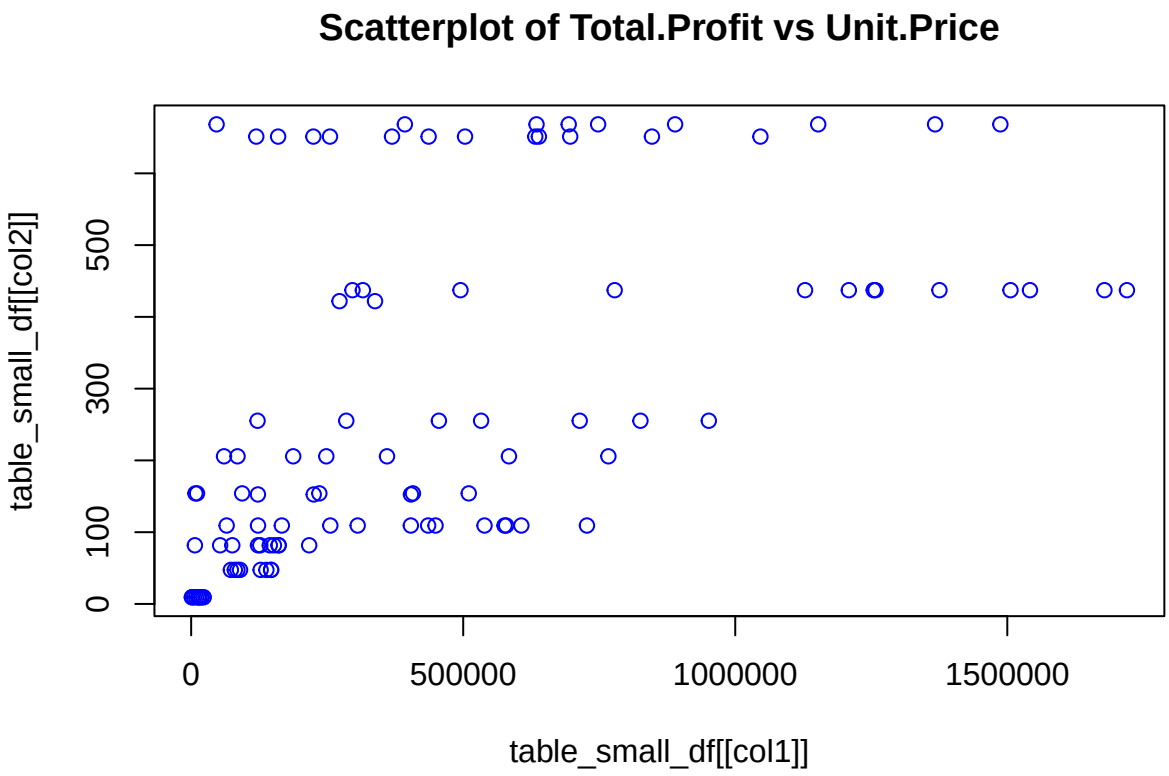

Scatterplot of Total.Cost vs Unit.Cost
table_small_df[[col2]]
0
100
200
300
400
500
0e+00
1e+06
2e+06
3e+06
4e+06
table_small_df[[col1]]

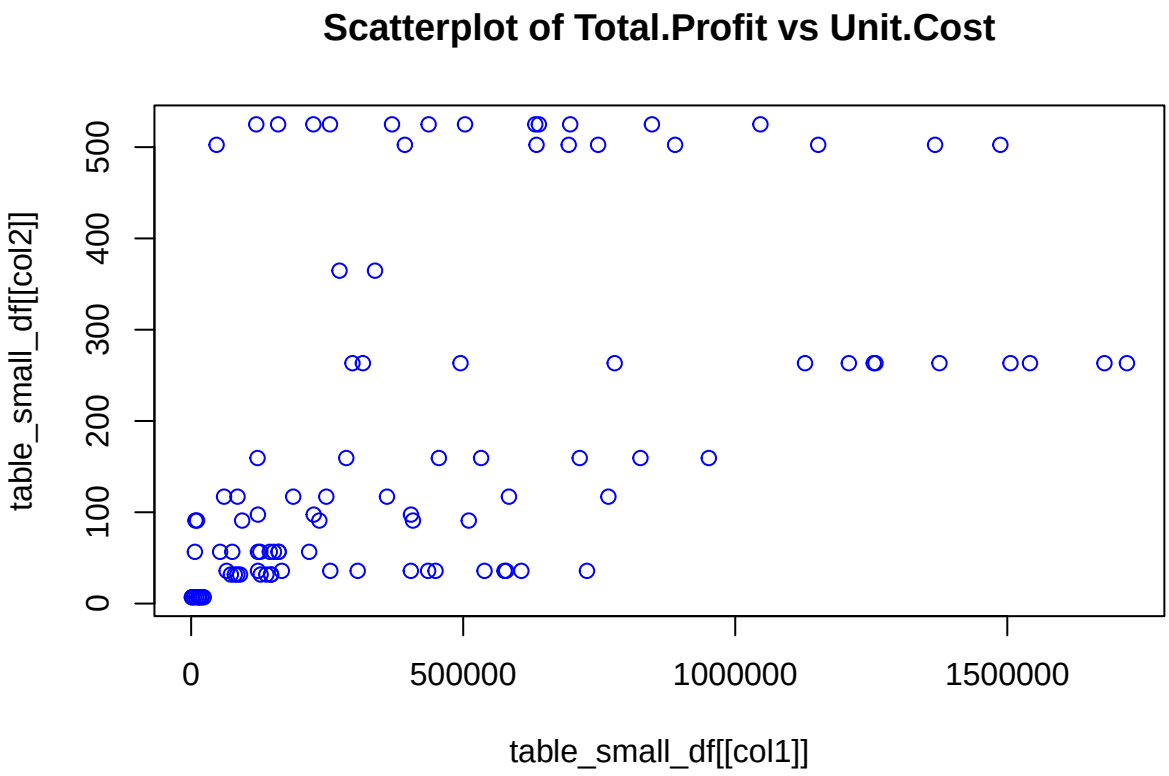

Scatterplot of Total.Cost vs Total.Revenue
table_small_df[[col2]]
0e+00
2e+06
4e+06
6e+06
0e+00
1e+06
2e+06
3e+06
4e+06
table_small_df[[col1]]

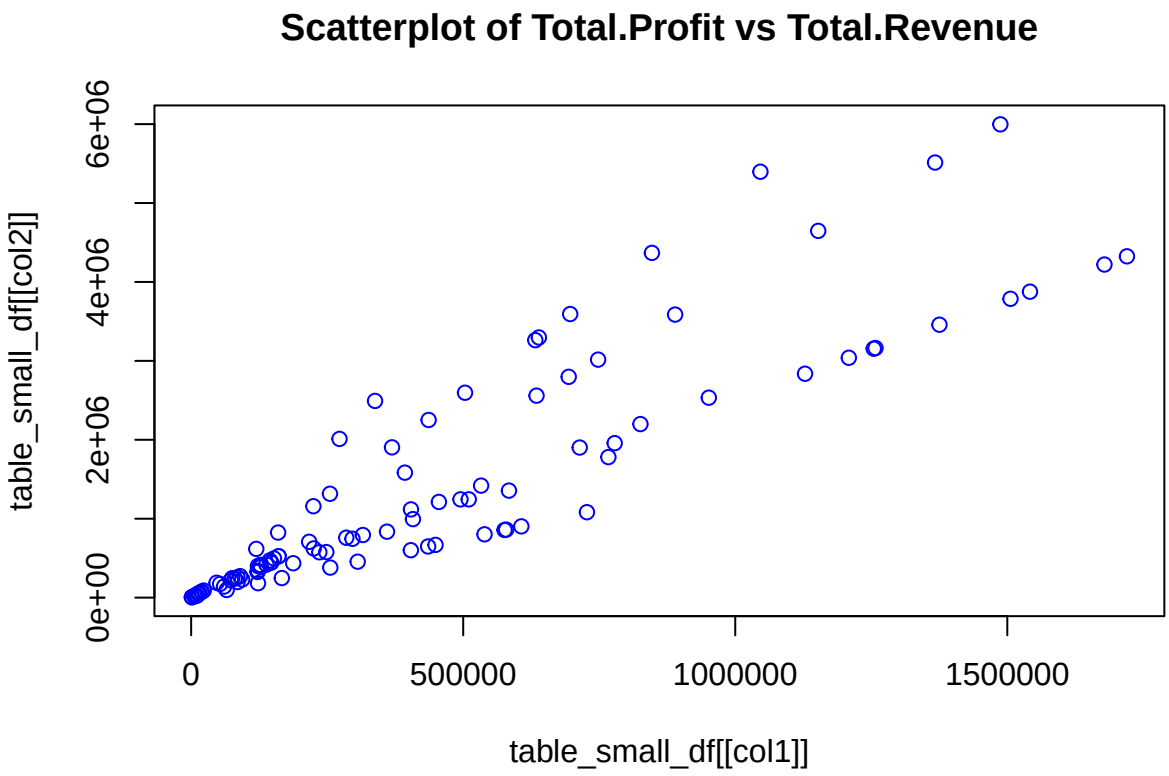

Scatterplot of Total.Cost vs Total.Profit
table_small_df[[col2]]
0
500000
1500000
0e+00
1e+06
2e+06
3e+06
4e+06
table_small_df[[col1]]

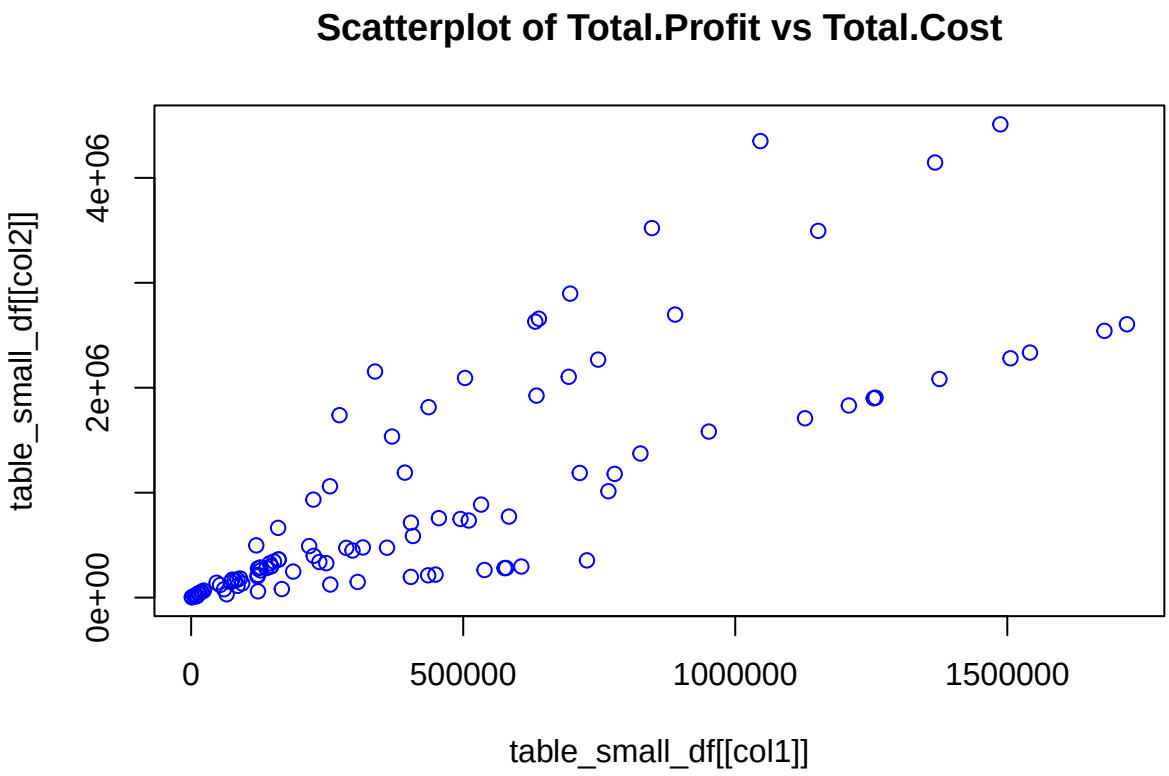

Scatterplot of Total.Profit vs Order.ID
table_small_df[[col2]]
1e+09
6e+08
2e+08
0
500000
1000000
1500000
table_small_df[[col1]]


Scatterplot of Total.Profit vs Units.Sold
table_small_df[[col2]]
0
2000
6000
10000
0
500000
1000000
1500000
table_small_df[[col1]]


Scatterplot of Total.Profit vs Unit.Price
table_small_df[[col2]]
0
100
300
500
0
500000
1000000
1500000
table_small_df[[col1]]


Scatterplot of Total.Profit vs Unit.Cost
table_small_df[[col2]]
0
100
200
300
400
500
0
500000
1000000
1500000
table_small_df[[col1]]


Scatterplot of Total.Profit vs Total.Revenue
table_small_df[[col2]]
0e+00
2e+06
4e+06
6e+06
0
500000
1000000
1500000
table_small_df[[col1]]



A scatter plot with scattered points suggests a lack of a strong linear relationship between variables, indicating their independence. The randomness in the distribution implies no structured or predictable connection. This phenomenon holds whether analyzing numeric or categorical variables. In statistical terms, this scattered appearance may signify homoscedasticity, indicating consistent variability across levels. While the lack of a clear pattern hints at independence or absence of correlation, further analysis is essential for a comprehensive understanding of the data dynamics.

A scatter plot showing points clustered at the top and bottom with a straight line in the middle suggests a potential linear relationship between the variables. This pattern indicates a positive correlation, implying that as one variable increases, the other tends to increase, and vice versa. Further statistical analysis, such as calculating correlation coefficients, is needed to quantify and confirm the strength of this relationship.

When scatter plots display a pattern with outliers concentrated at the top and the majority of data points clustered towards the bottom, it implies potential non-linear relationships or the presence of influential data points. These outliers can strongly affect correlation or regression analyses, necessitating careful investigation. Understanding the nature of these outliers is crucial, as they might indicate unique characteristics or anomalies in the dataset. Further analysis, such as examining residuals and exploring alternative modeling techniques, may be needed to accurately capture underlying patterns in the data.

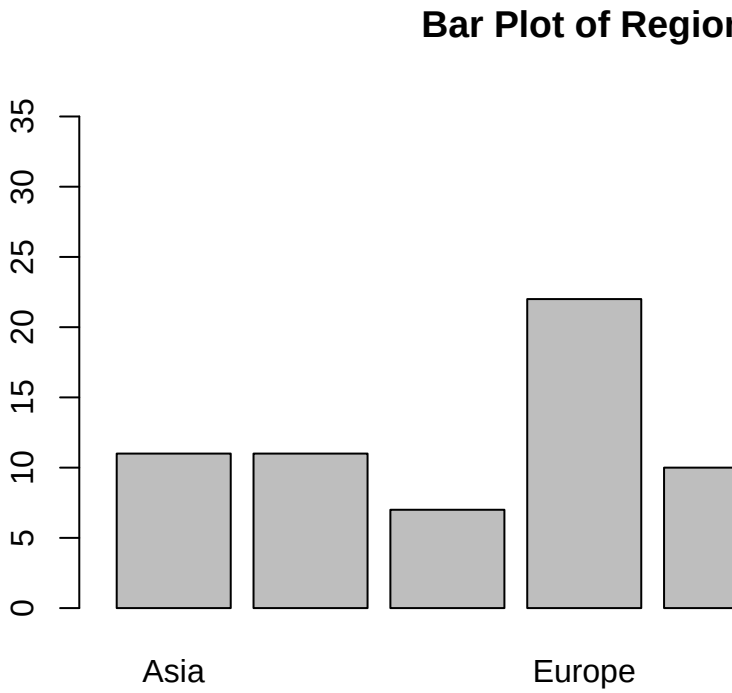
A diagonal line in a scatter plot from the bottom-left to the top-right indicates a positive linear relationship between the variables being plotted. This suggests that as one variable increases, the other also tends to increase. The steeper the slope, the stronger the positive correlation. Further analysis, such as calculating correlation coefficients and conducting regression analysis, can provide a more quantitative understanding of the relationship.

A scatter plot displaying a straight vertical line from bottom to top suggests that the two variables being compared have a perfect linear relationship. This means that as one variable increases, the other also increases proportionally. The correlation coefficient would be +1, indicating a strong positive correlation.

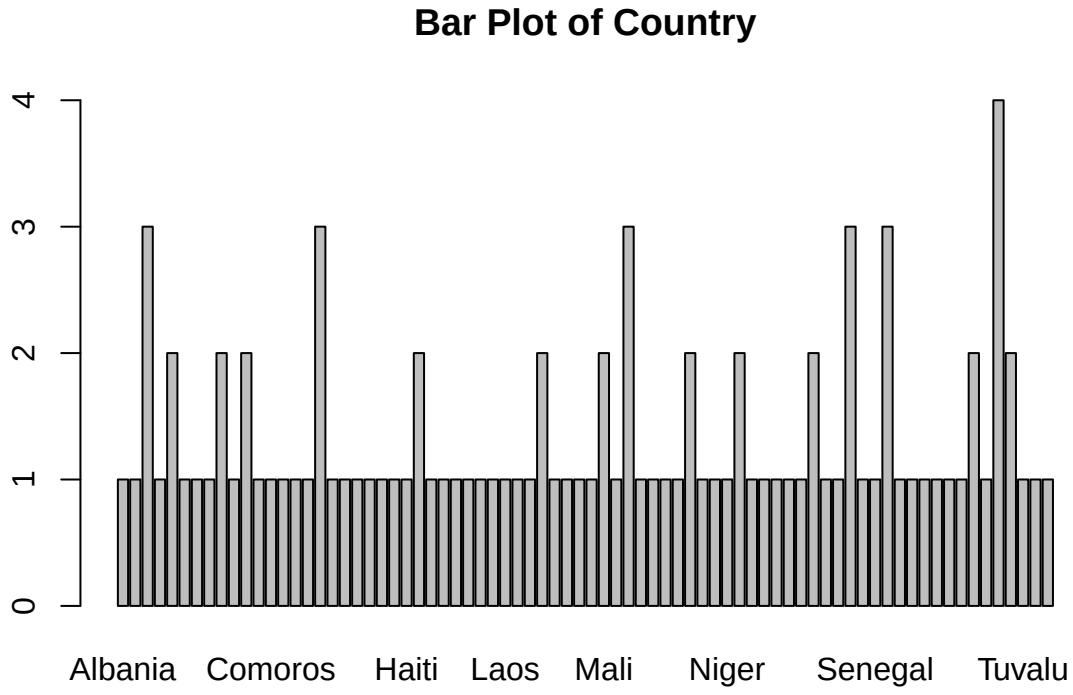
However, it's important to note that this ideal scenario is less common in real-world data, and some variations or deviations may be present due to other factors or measurement errors.

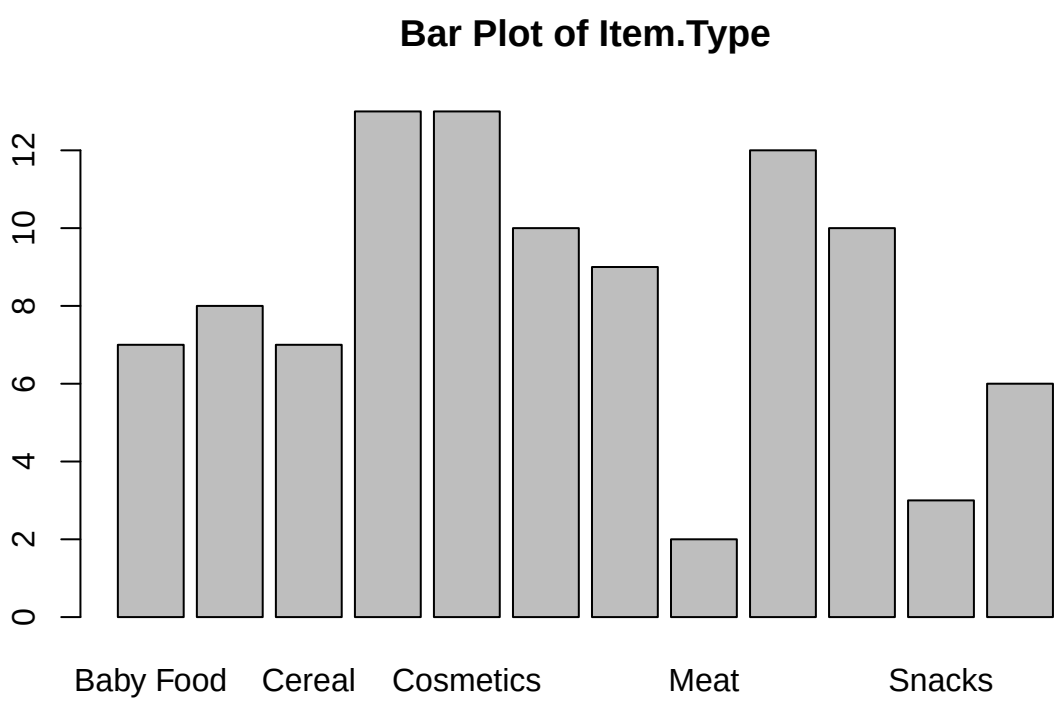
A scatter plot with points aligned horizontally indicates a perfect linear relationship where the two variables being compared have a constant value for one of them, regardless of the changes in the other variable. This implies a correlation coefficient of -1, representing a strong negative correlation. In simpler terms, as one variable increases, the other decreases proportionally. As with other ideal scenarios, variations and deviations may occur in real-world data due to external factors or measurement errors.

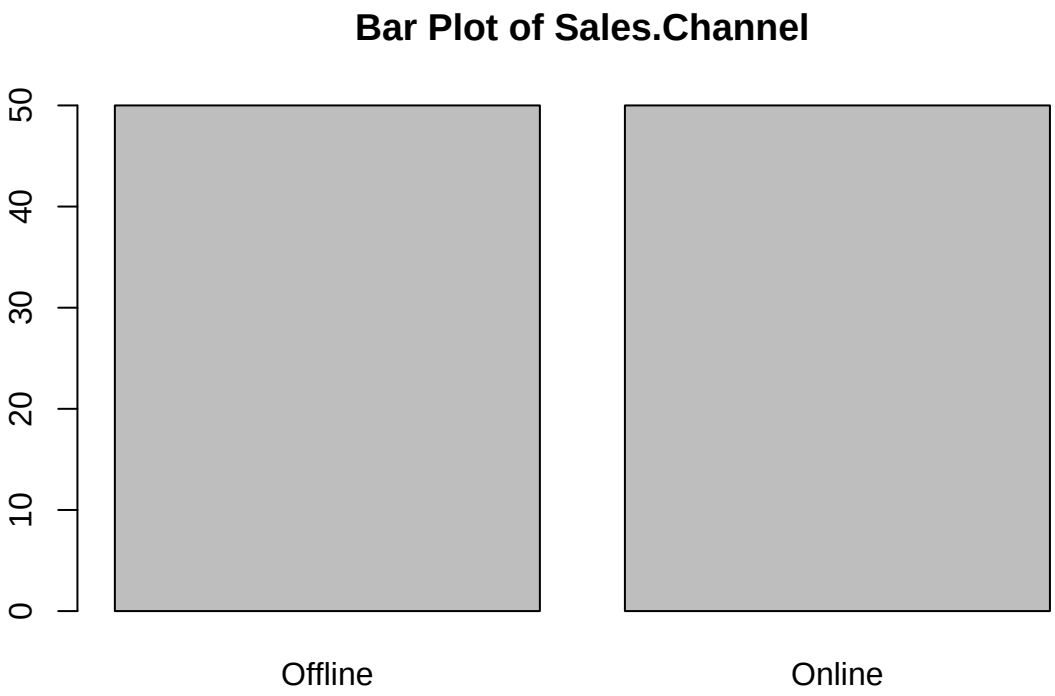
```
for (column_name in names(table_small_df)) {
  if (!is.numeric(table_small_df[[column_name]])) {
    barplot(table(table_small_df[[column_name]]), main = paste("Bar Plot of", column_name))
  }
}
```



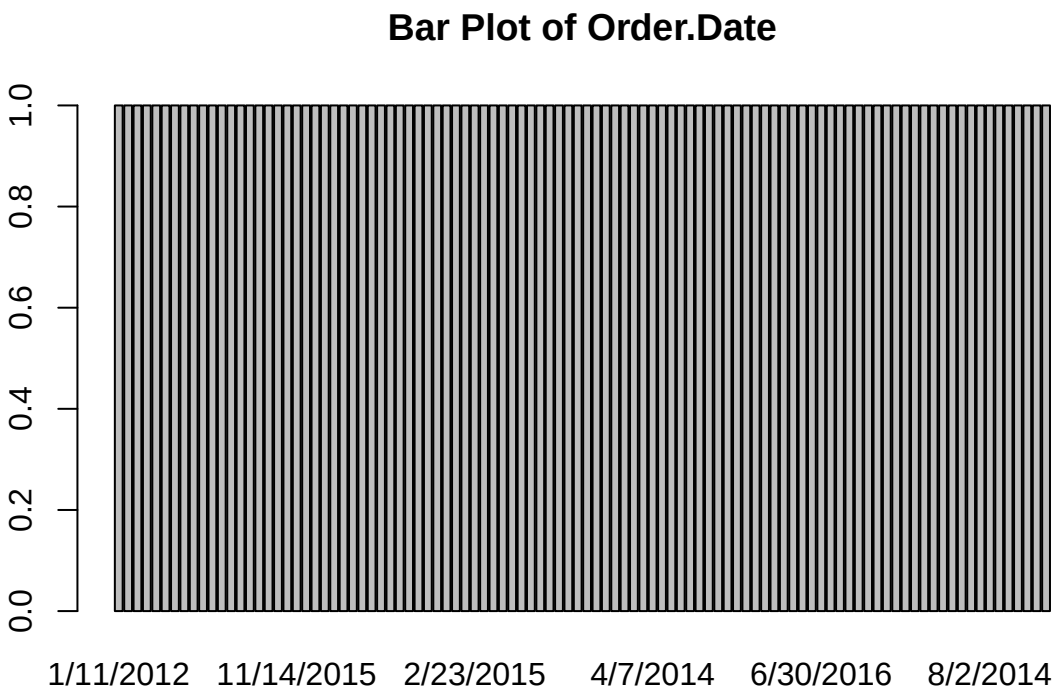
Frequency for each variables for each columns:

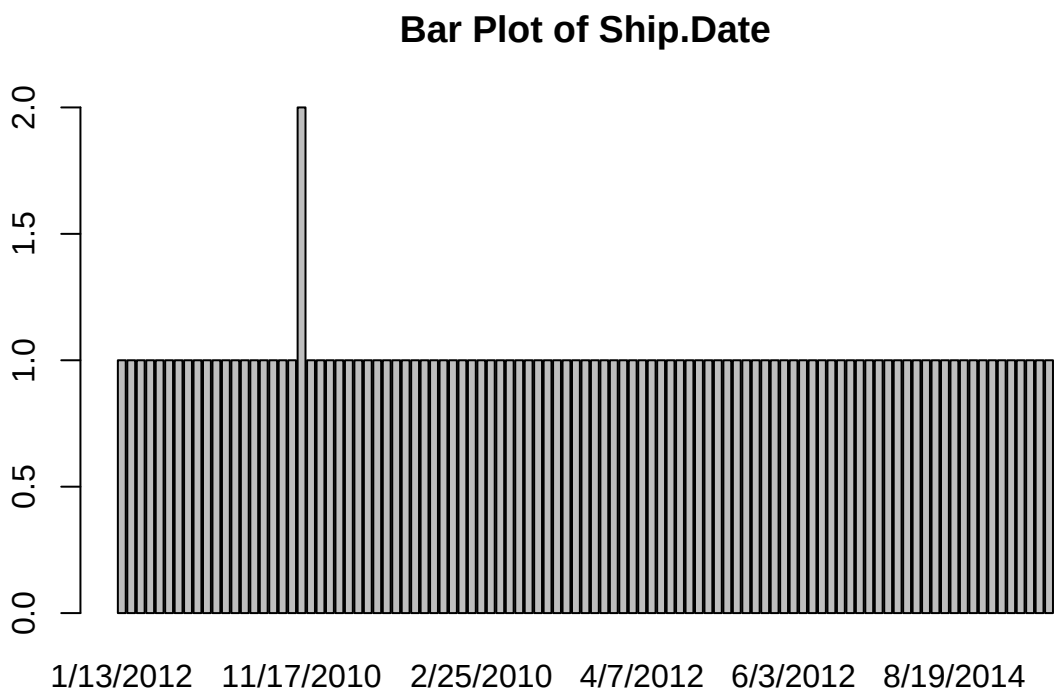



Bar Plot of Item.Type
0
2
4
6
8
10
12
Baby Food
Cereal
Cosmetics
Meat
Snacks


Bar Plot of Sales.Channel
0
10
20
30
40
50
Offline
Online


Bar Plot of Order.Priority
0
5
10
15
20
25
30
C
H
L
M

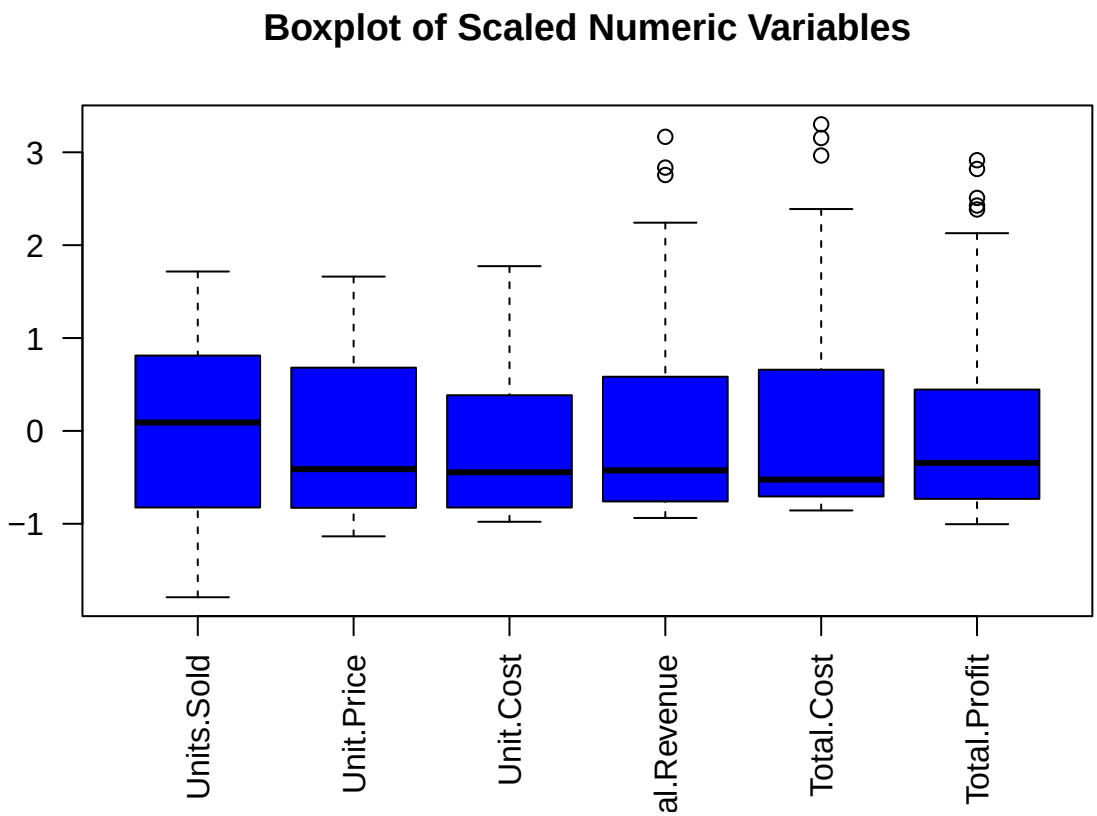

Bar Plot of Order.Date
0.0
0.2
0.4
0.6
0.8
1.0
1/11/2012
11/14/2015
2/23/2015
4/7/2014
6/30/2016
8/2/2014



Boxplots for all numeric columns. From the above summary of your dataset, it appears that the numeric variables (Units.Sold, Unit.Price, Unit.Cost, Total.Revenue, Total.Cost, Total.Profit) have a wide range of values with varying scales. The Units.Sold variable, for instance, has a minimum value of 124 and a maximum of 9,925, while the Total.Revenue variable ranges from 4,870 to 5,997,055.

Given the diverse scales of these variables, when I create a boxplot for all of them at once, some may dominate the plot, making it challenging to see the details of others. I considered normalizing or scaling these variables to bring them to a similar scale for better visualization.

```
# Assuming 'df' is your dataframe
scaled_df <- table_small_df[, c("Units.Sold", "Unit.Price", "Unit.Cost", "Total.Revenue", "Total.Cost",

# Scale the numeric variables
scaled_df <- scale(scaled_df)

# Create a boxplot
boxplot(scaled_df, col = "blue", main = "Boxplot of Scaled Numeric Variables", las = 2)
```



The boxplot analysis reveals a positive distribution, indicating a concentration of data towards higher values. The right-skewed pattern suggests a majority of observations falling on the higher end of the axis. Additionally, the presence of outliers highlights extreme values that significantly deviate from the general trend, adding complexity to the dataset.

Large Dataset

Content of the Tables:

```
# head() displays the first few rows of a dataframe, giving you a quick look at the data.
head(table_large_df)
```

Head of the Dataframe:

```
##                          Region                          Country       Item.Type
## 1           Sub-Saharan Africa                             Chad Office Supplies
## 2                        Europe                           Latvia       Beverages
## 3 Middle East and North Africa                         Pakistan      Vegetables
## 4           Sub-Saharan Africa Democratic Republic of the Congo       Household
## 5                        Europe                   Czech Republic       Beverages
## 6           Sub-Saharan Africa                     South Africa       Beverages
##   Sales.Channel Order.Priority Order.Date  Order.ID Ship.Date Units.Sold
## 1        Online              L  1/27/2011 292494523 2/12/2011       4484
```

```
## 2        Online              C 12/28/2015 361825549 1/23/2016       1075
## 3       Offline              C  1/13/2011 141515767  2/1/2011       6515
## 4        Online              C  9/11/2012 500364005 10/6/2012       7683
## 5        Online              C 10/27/2015 127481591 12/5/2015       3491
## 6       Offline              H  7/10/2012 482292354 8/21/2012       9880
##   Unit.Price Unit.Cost Total.Revenue Total.Cost Total.Profit
## 1     651.21    524.96    2920025.64 2353920.64    566105.00
## 2      47.45     31.79      51008.75   34174.25     16834.50
## 3     154.06     90.93    1003700.90  592408.95    411291.95
## 4     668.27    502.54    5134318.41 3861014.82   1273303.59
## 5      47.45     31.79     165647.95  110978.89     54669.06
## 6      47.45     31.79     468806.00  314085.20    154720.80
```

```
# summary() provides a summary of the central tendency, dispersion, and distribution of the data, inclu

summary(table_large_df)
```

Summary Statistic:

```
##     Region             Country           Item.Type         Sales.Channel
##  Length:10000       Length:10000       Length:10000       Length:10000
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##  Order.Priority      Order.Date           Order.ID          Ship.Date
##  Length:10000       Length:10000       Min.   :100089156   Length:10000
##  Class :character   Class :character   1st Qu.:321806669   Class :character
##  Mode  :character   Mode  :character   Median :548566305   Mode  :character
##                                        Mean   :549871874
##                                        3rd Qu.:775998104
##                                        Max.   :999934232
##    Units.Sold      Unit.Price       Unit.Cost      Total.Revenue
##  Min.   :    2   Min.   :  9.33   Min.   :  6.92   Min.   :    168
##  1st Qu.: 2531   1st Qu.:109.28   1st Qu.: 56.67   1st Qu.: 288551
##  Median : 4962   Median :205.70   Median :117.11   Median : 800051
##  Mean   : 5003   Mean   :268.14   Mean   :188.81   Mean   :1333355
##  3rd Qu.: 7472   3rd Qu.:437.20   3rd Qu.:364.69   3rd Qu.:1819143
##  Max.   :10000   Max.   :668.27   Max.   :524.96   Max.   :6680027
##    Total.Cost       Total.Profit
##  Min.   :    125   Min.   :     43.4
##  1st Qu.: 164786   1st Qu.:  98329.1
##  Median : 481606   Median : 289099.0
##  Mean   : 938266   Mean   : 395089.3
##  3rd Qu.:1183822   3rd Qu.: 566422.7
##  Max.   :5241726   Max.   :1738178.4
```

The dataset contains 10,000 observations for each variable, and the variables have different types (character, integer, numeric). The descriptive statistics give insights into the distribution and central tendency of the numeric variables.

```r
# The str() function provides an overview of the structure of an R object, showing the data type and st
str(table_large_df)
```

Structure of Tables:

```
## 'data.frame':    10000 obs. of  14 variables:
##  $ Region        : chr  "Sub-Saharan Africa" "Europe" "Middle East and North Africa" "Sub-Saharan Af
##  $ Country       : chr  "Chad" "Latvia" "Pakistan" "Democratic Republic of the Congo" ...
##  $ Item.Type     : chr  "Office Supplies" "Beverages" "Vegetables" "Household" ...
##  $ Sales.Channel : chr  "Online" "Online" "Offline" "Online" ...
##  $ Order.Priority: chr  "L" "C" "C" "C" ...
##  $ Order.Date    : chr  "1/27/2011" "12/28/2015" "1/13/2011" "9/11/2012" ...
##  $ Order.ID      : int  292494523 361825549 141515767 500364005 127481591 482292354 844532620 564251
##  $ Ship.Date     : chr  "2/12/2011" "1/23/2016" "2/1/2011" "10/6/2012" ...
##  $ Units.Sold    : int  4484 1075 6515 7683 3491 9880 4825 3330 2431 6197 ...
##  $ Unit.Price    : num  651.2 47.5 154.1 668.3 47.5 ...
##  $ Unit.Cost     : num  525 31.8 90.9 502.5 31.8 ...
##  $ Total.Revenue : num  2920026 51009 1003701 5134318 165648 ...
##  $ Total.Cost    : num  2353921 34174 592409 3861015 110979 ...
##  $ Total.Profit  : num  566105 16835 411292 1273304 54669 ...
```

Again, the dataset consists of 10,000 observations and 14 variables. The variables include categorical features such as Region, Country, Item Type, Sales Channel, Order Priority, Order Date, and Ship Date. Additionally, there are numerical features, including Order ID, Units Sold, Unit Price, Unit Cost, Total Revenue, Total Cost, and Total Profit.

The categorical features provide information about the geographic and logistical aspects of each transaction, while the numerical features quantify the financial details, such as sales quantities, pricing, and associated costs. Notably, the dataset covers a diverse range of regions, countries, and product types, making it suitable for exploring sales trends and financial performance.

Dataset Characteristics (Structure, Size, Dependencies, Labels, etc.):

```r
# nrow() and ncol() give the number of rows and columns, respectively.
number_of_rows <- nrow(table_large_df)
print(paste("Number of rows:", number_of_rows))
```

Number of Rows and Columns:

```
## [1] "Number of rows: 10000"
```

```r
number_of_columns <- ncol(table_large_df)
print(paste("Number of rows:", number_of_columns))
```

```
## [1] "Number of rows: 14"
```

```r
# names() shows the variable (column) names.
names(table_large_df)
```

```
##  [1] "Region"         "Country"        "Item.Type"      "Sales.Channel"
##  [5] "Order.Priority" "Order.Date"     "Order.ID"       "Ship.Date"
##  [9] "Units.Sold"     "Unit.Price"     "Unit.Cost"      "Total.Revenue"
## [13] "Total.Cost"     "Total.Profit"
```

```r
# Get the total number of NA's in the entire dataframe
total_na_count <- sum(is.na(table_large_df))
print("Total number of NA's in the entire dataframe:")
```

Number of NA's:

```
## [1] "Total number of NA's in the entire dataframe:"
```

```r
print(total_na_count)
```

```
## [1] 0
```

```r
# Get the number of NA's in each column
na_per_column <- colSums(is.na(table_large_df))
print("Number of NA's in each column:")
```

```
## [1] "Number of NA's in each column:"
```

```r
print(na_per_column)
```

```
##         Region        Country      Item.Type  Sales.Channel Order.Priority
##              0              0              0              0              0
##     Order.Date       Order.ID      Ship.Date     Units.Sold     Unit.Price
##              0              0              0              0              0
##      Unit.Cost  Total.Revenue     Total.Cost   Total.Profit
##              0              0              0              0
```

There are no NA's in this dataset.

```r
# Exclude "Order.ID" and non-numeric variables
numeric_table_large_df <- table_large_df[sapply(table_large_df, is.numeric) & colnames(table_large_df)

# Check if there are any missing values
if (any(is.na(numeric_table_large_df))) {
  cat("Warning: There are missing values in the numeric columns. Consider handling missing values before
} else {
```

```r
  # Calculate standard deviation and variance for numeric variables
  std_dev <- apply(numeric_table_large_df, 2, sd)
  variance <- apply(numeric_table_large_df, 2, var)

  # Combine results into a data frame
  result_table_large_df <- data.frame(Variable = names(numeric_table_large_df), Standard_Deviation = st

 # Print the result as a nicely formatted table
  result_table <- kable(result_table_large_df, "html") %>%
    kable_styling()

  # Display the table
  print(result_table)
}
```

Standard Deviation and Variance:

```
## <table class="table" style="margin-left: auto; margin-right: auto;">
##  <thead>
##   <tr>
##    <th style="text-align:left;">   </th>
##    <th style="text-align:left;"> Variable </th>
##    <th style="text-align:right;"> Standard_Deviation </th>
##    <th style="text-align:right;"> Variance </th>
##   </tr>
##  </thead>
## <tbody>
##   <tr>
##    <td style="text-align:left;"> Units.Sold </td>
##    <td style="text-align:left;"> Units.Sold </td>
##    <td style="text-align:right;"> 2873.2465 </td>
##    <td style="text-align:right;"> 8.255545e+06 </td>
##   </tr>
##   <tr>
##    <td style="text-align:left;"> Unit.Price </td>
##    <td style="text-align:left;"> Unit.Price </td>
##    <td style="text-align:right;"> 217.9441 </td>
##    <td style="text-align:right;"> 4.749963e+04 </td>
##   </tr>
##   <tr>
##    <td style="text-align:left;"> Unit.Cost </td>
##    <td style="text-align:left;"> Unit.Cost </td>
##    <td style="text-align:right;"> 176.4459 </td>
##    <td style="text-align:right;"> 3.113316e+04 </td>
##   </tr>
##   <tr>
##    <td style="text-align:left;"> Total.Revenue </td>
##    <td style="text-align:left;"> Total.Revenue </td>
##    <td style="text-align:right;"> 1465026.1739 </td>
##    <td style="text-align:right;"> 2.146302e+12 </td>
##   </tr>
##   <tr>
##    <td style="text-align:left;"> Total.Cost </td>
##    <td style="text-align:left;"> Total.Cost </td>
```

```
##    <td style="text-align:right;"> 1145914.0694 </td>
##    <td style="text-align:right;"> 1.313119e+12 </td>
##   </tr>
##   <tr>
##    <td style="text-align:left;"> Total.Profit </td>
##    <td style="text-align:left;"> Total.Profit </td>
##    <td style="text-align:right;"> 377554.9607 </td>
##    <td style="text-align:right;"> 1.425477e+11 </td>
##   </tr>
## </tbody>
## </table>
```

The dataset reveals interesting insights into key variables, each with its unique characteristics. 'Units Sold' displays considerable variability, with a standard deviation of 2873.25 and a variance of 8.26 million, suggesting diverse sales patterns. 'Unit Price' exhibits lower variability, indicated by a standard deviation of 217.94 and a variance of 47,499.63, reflecting a more stable pricing strategy.

Similarly, 'Unit Cost' showcases moderate variability, with a standard deviation of 176.45 and a variance of 31,133.32, hinting at factors impacting production costs. The financial indicators 'Total Revenue' and 'Total Cost' demonstrate significant fluctuations, with standard deviations of 1,465,026.17 and 1,145,914.07, respectively, emphasizing the complexity of overall sales and operational expenses.

Lastly, 'Total Profit' displays notable variability, indicated by a standard deviation of 377,554.96 and a variance of 142,547.7, suggesting potential areas for further investigation to optimize overall financial outcomes. This comprehensive summary provides a snapshot of the dataset's numerical features, paving the way for deeper analysis and strategic decision-making.

Relationships between numeric variables using a correlation matrix.

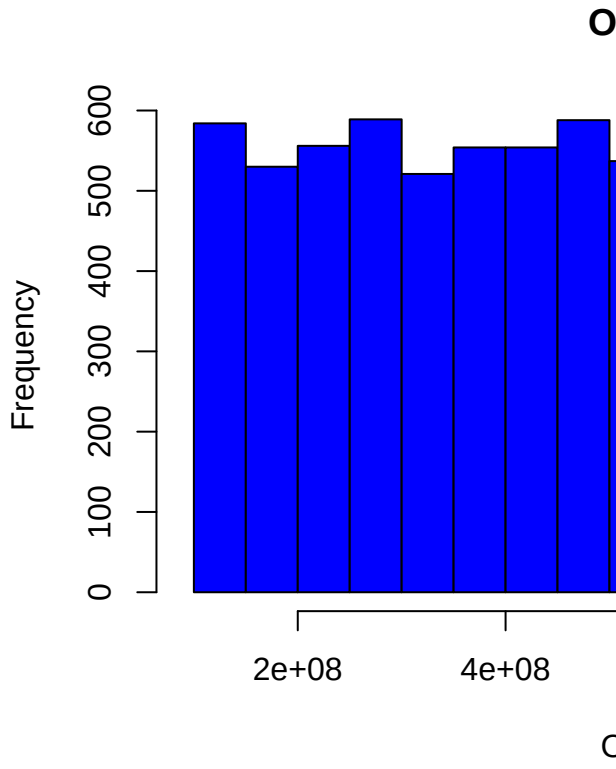
```
cor_matrix <- cor(table_large_df[, numeric_columns])
cor_matrix
```

See below visualization for Correlation.

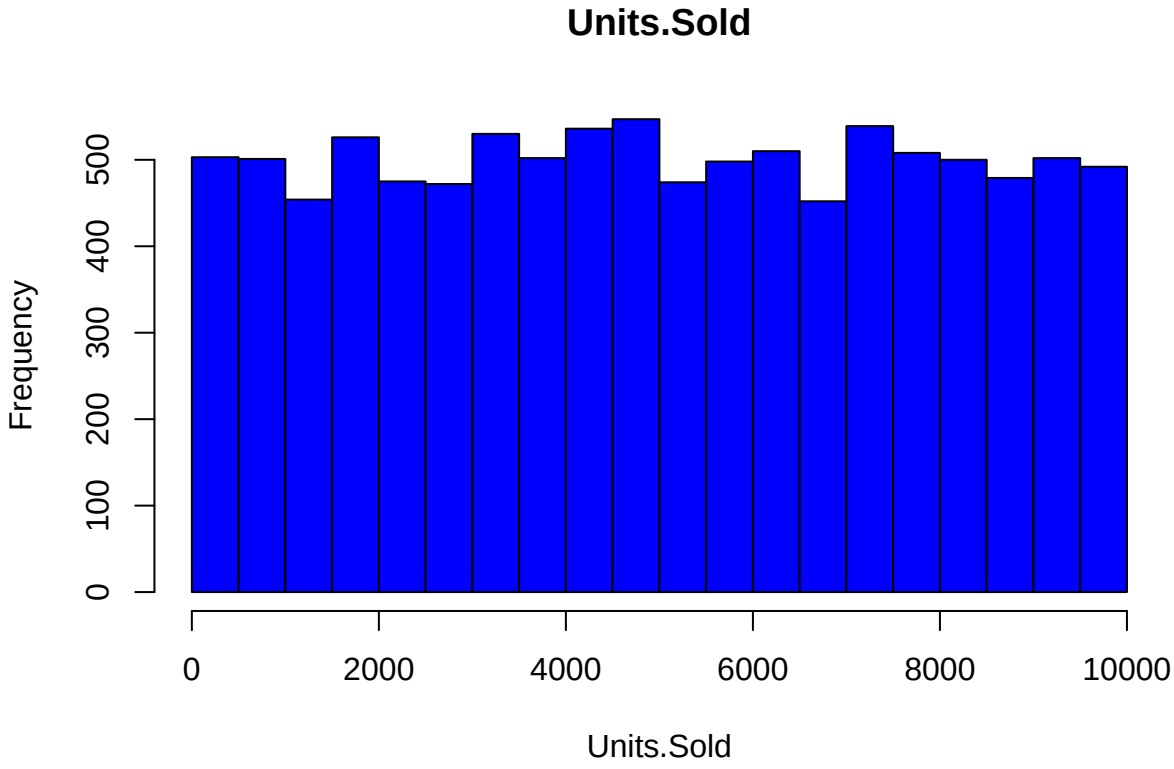
```
##                   Order.ID  Units.Sold  Unit.Price   Unit.Cost Total.Revenue
## Order.ID       1.000000000 -0.01973240  0.01217491  0.01044871  -0.009288123
## Units.Sold    -0.019732401  1.00000000 -0.01297840 -0.01244102   0.518615102
## Unit.Price     0.012174909 -0.01297840  1.00000000  0.98632386   0.733225977
## Unit.Cost      0.010448711 -0.01244102  0.98632386  1.00000000   0.723267379
## Total.Revenue -0.009288123  0.51861510  0.73322598  0.72326738   1.000000000
## Total.Cost    -0.009530296  0.46617752  0.74924307  0.75983568   0.987873673
## Total.Profit  -0.007115369  0.59749001  0.57111439  0.50032251   0.882011543
##                 Total.Cost Total.Profit
## Order.ID      -0.009530296 -0.007115369
## Units.Sold     0.466177522  0.597490009
## Unit.Price     0.749243073  0.571114386
## Unit.Cost      0.759835678  0.500322512
## Total.Revenue  0.987873673  0.882011543
## Total.Cost     1.000000000  0.798153247
## Total.Profit   0.798153247  1.000000000
```

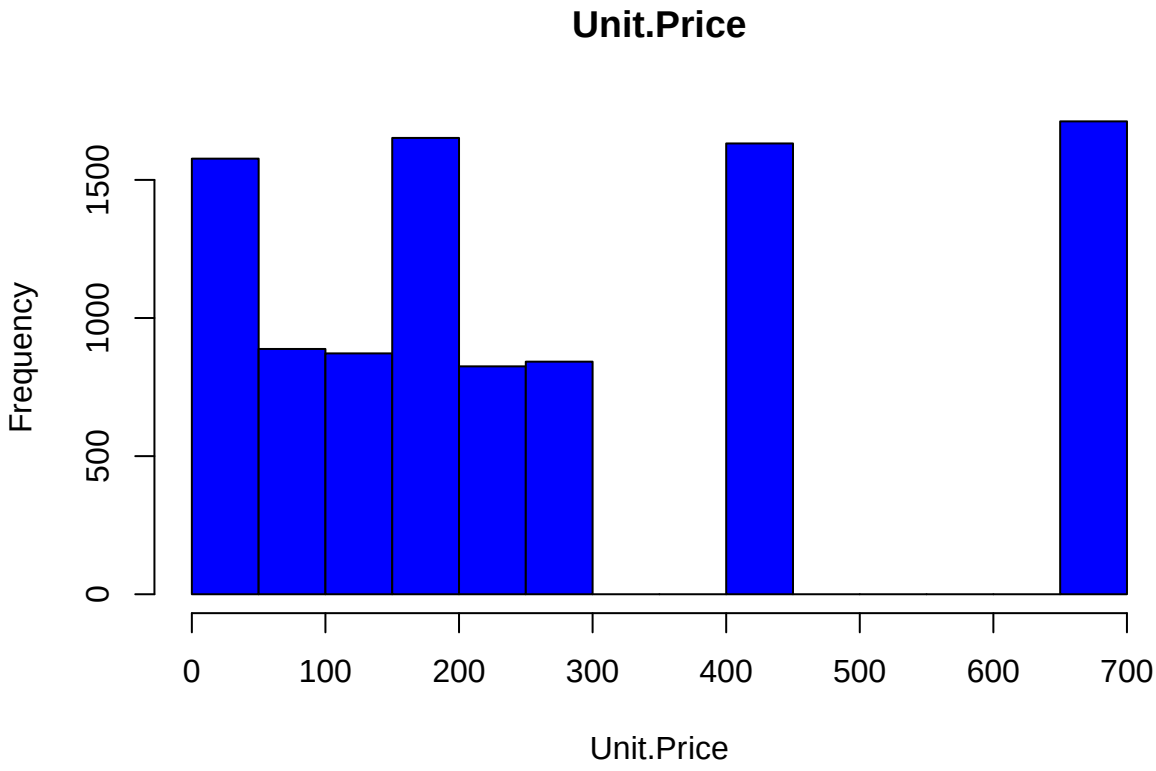
Visualization

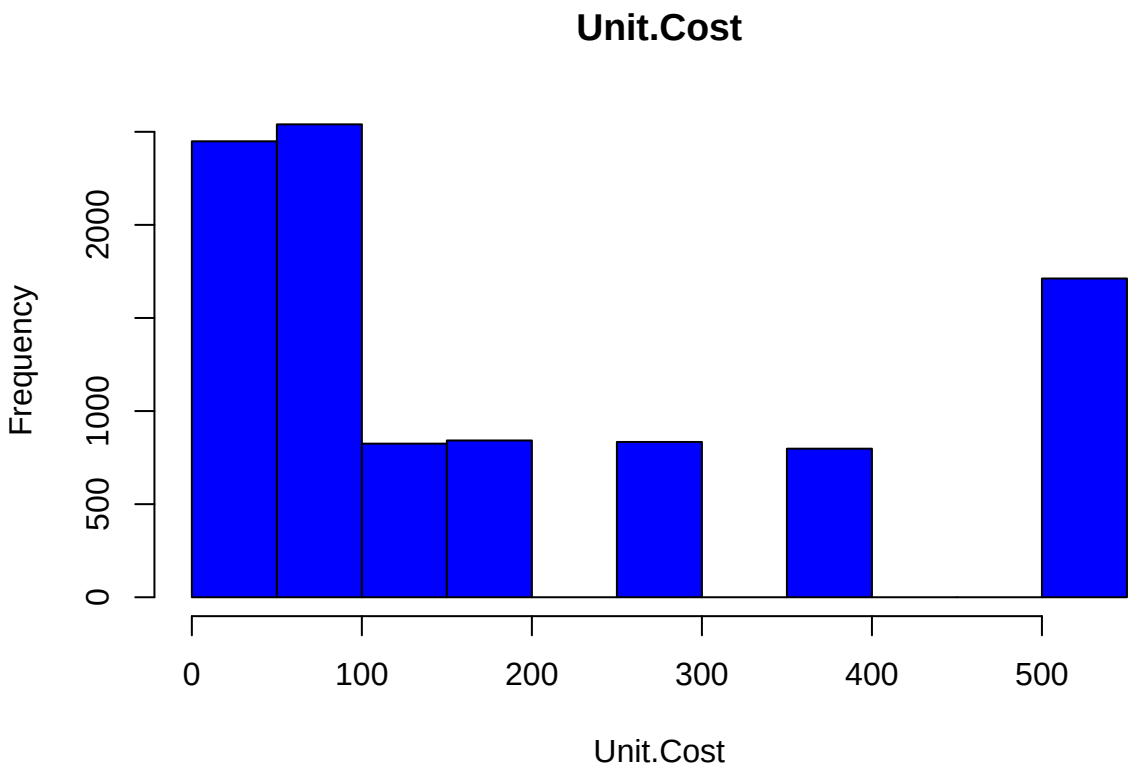
```r
# Identify numeric columns
numeric_columns <- sapply(table_large_df, is.numeric)

# Create histograms for numeric columns with blue color
for (col in names(table_large_df)[numeric_columns]) {
  hist(table_large_df[[col]], main = col, xlab = col, col = "blue")
}
```

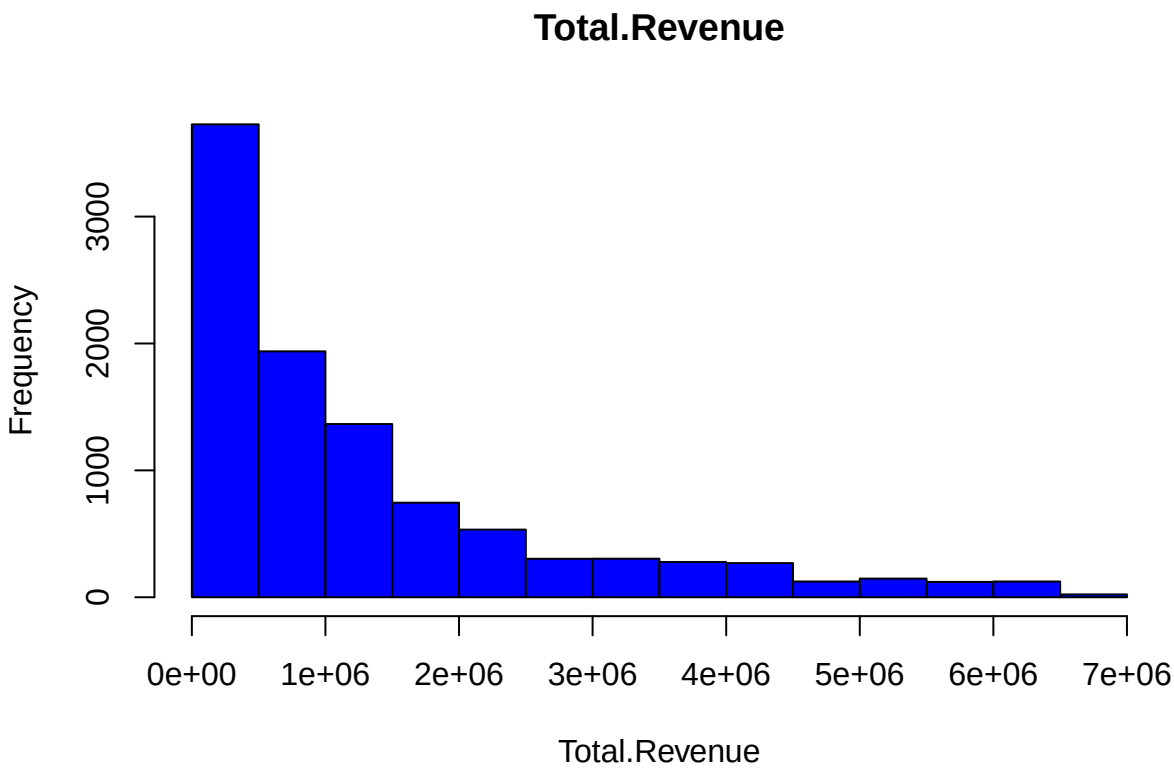


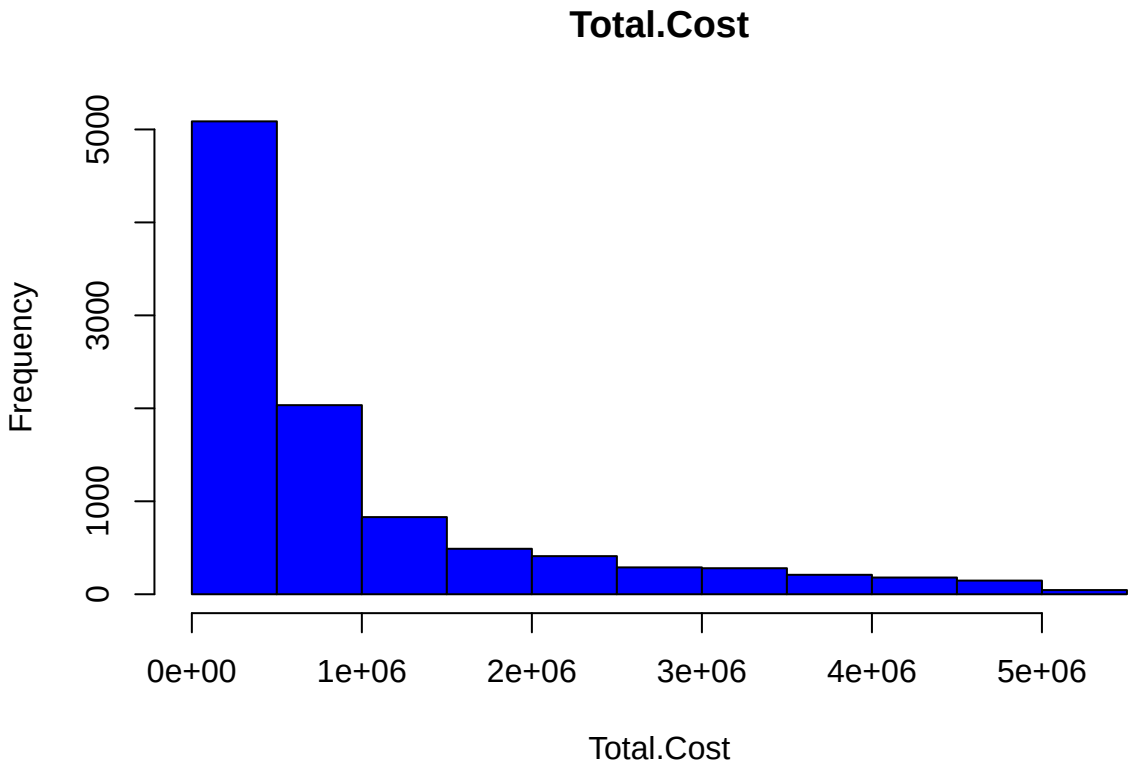
Histograms to visualize the distribution of numeric variables.

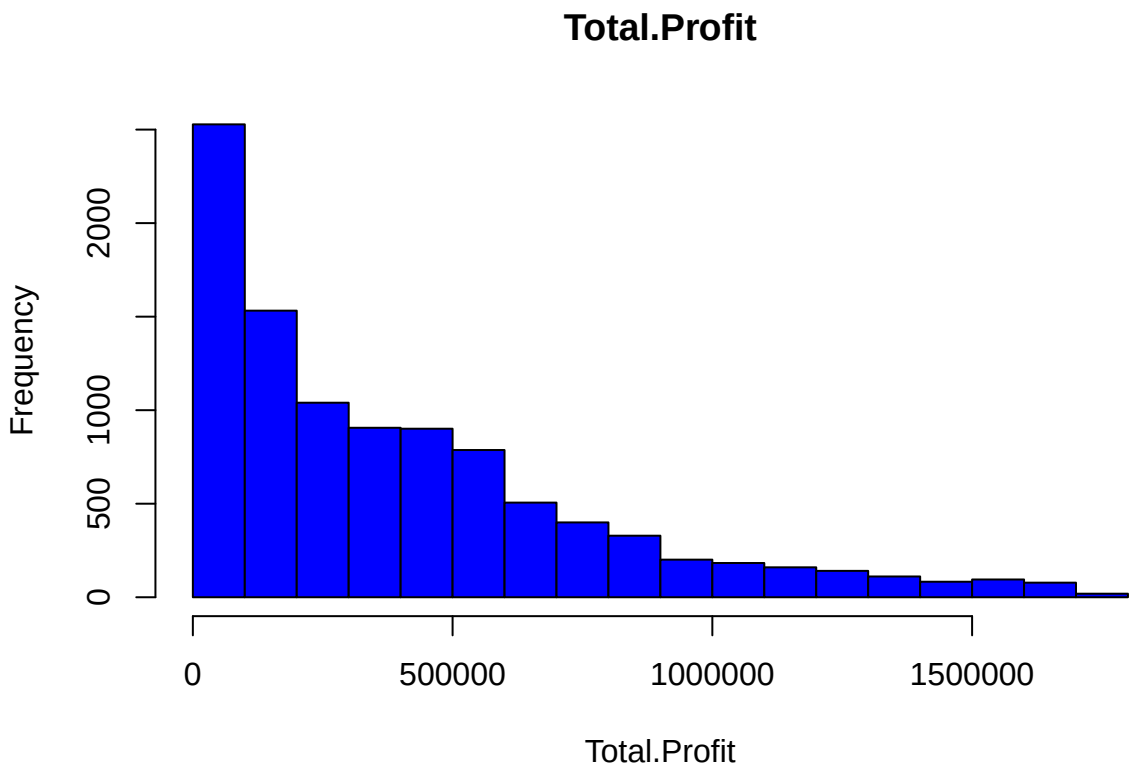



Unit.Price
Frequency
0
500
1000
1500
0
100
200
300
400
500
600
700
Unit.Price


Unit.Cost
Frequency
0
500
1000
2000
0
100
200
300
400
500
Unit.Cost

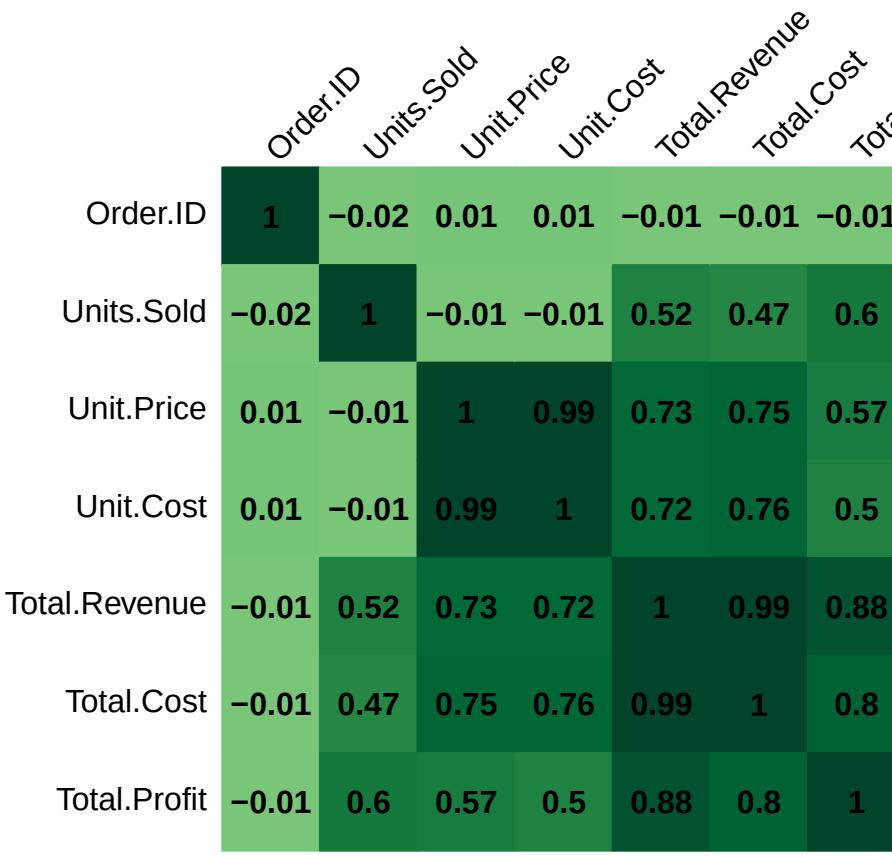

Total.Revenue
Frequency
0
1000
2000
3000
0e+00
1e+06
2e+06
3e+06
4e+06
5e+06
6e+06
7e+06
Total.Revenue


Total.Cost
Frequency
0
1000
3000
5000
0e+00
1e+06
2e+06
3e+06
4e+06
5e+06
Total.Cost



The result of the of Histograms to visualize the distribution of numeric variables is similar as the small dataset.

```
# Create a full correlation plot with a lighter color palette
corrplot(cor_matrix, method = "color", addCoef.col = "black", tl.col = "black", tl.srt = 45, col = colo
```



Correlation between variables.

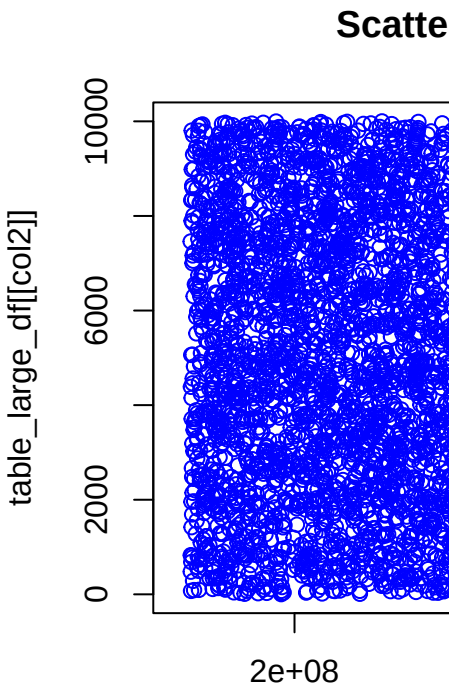
Similar to the small data set the correlation coefficients in the provided dataset offer valuable insights into the relationships between various key variables. Notably, a correlation coefficient of 0.99 between Unit Cost and Unit Price indicates an exceptionally strong positive correlation. This implies that as the cost of producing a unit increases, there is a proportional increase in its selling price. This tight relationship suggests a direct link between production costs and pricing.

Moving on to Total Cost and Total Revenue, a correlation coefficient slightly higher of .01 the the small dataset of 0.99 suggests a robust positive correlation. This implies that an increase in the total costs incurred in production corresponds to a proportional increase in the total revenue generated. This strong positive correlation underscores the interconnectedness of production costs and overall revenue.

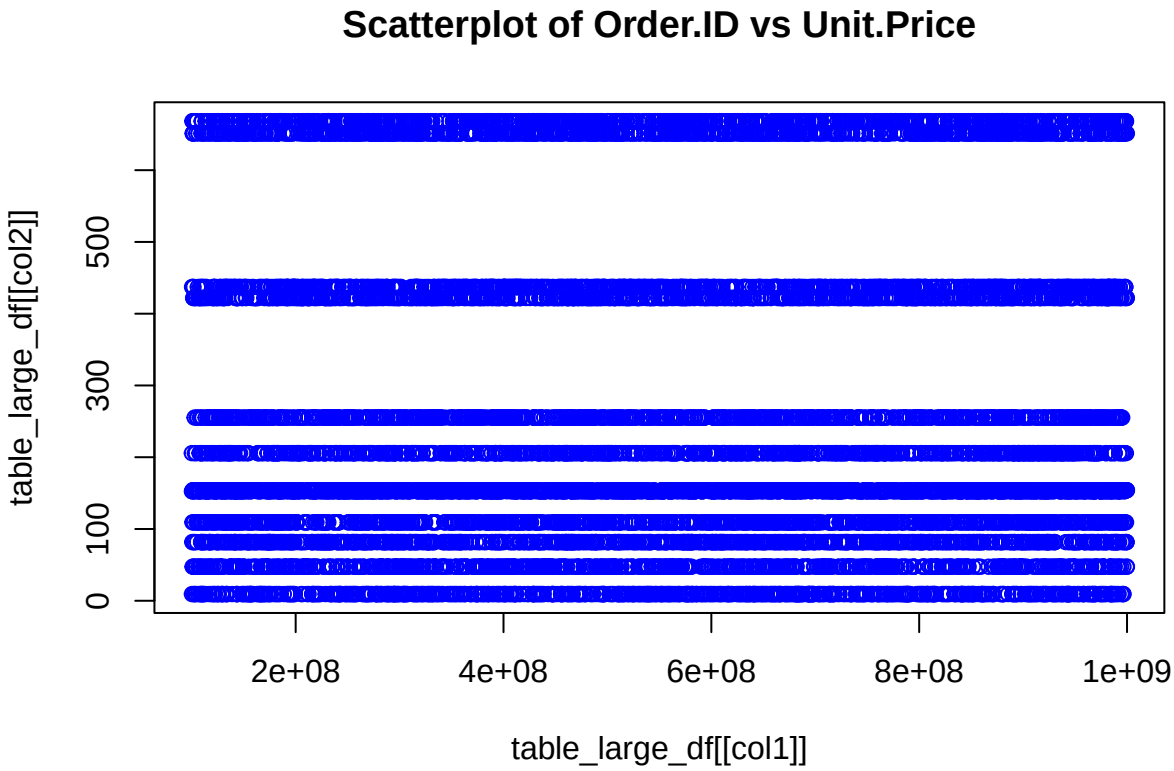
The correlation coefficient of 0.88 between Total Profit and Total Revenue indicates a little less strong positive correlation then the small dataset, albeit slightly less intense than the previous examples. This suggests that as total profits increase, there is a tendency for total revenue to increase as well. While not as extreme, this correlation emphasizes a meaningful relationship between overall profitability and revenue.

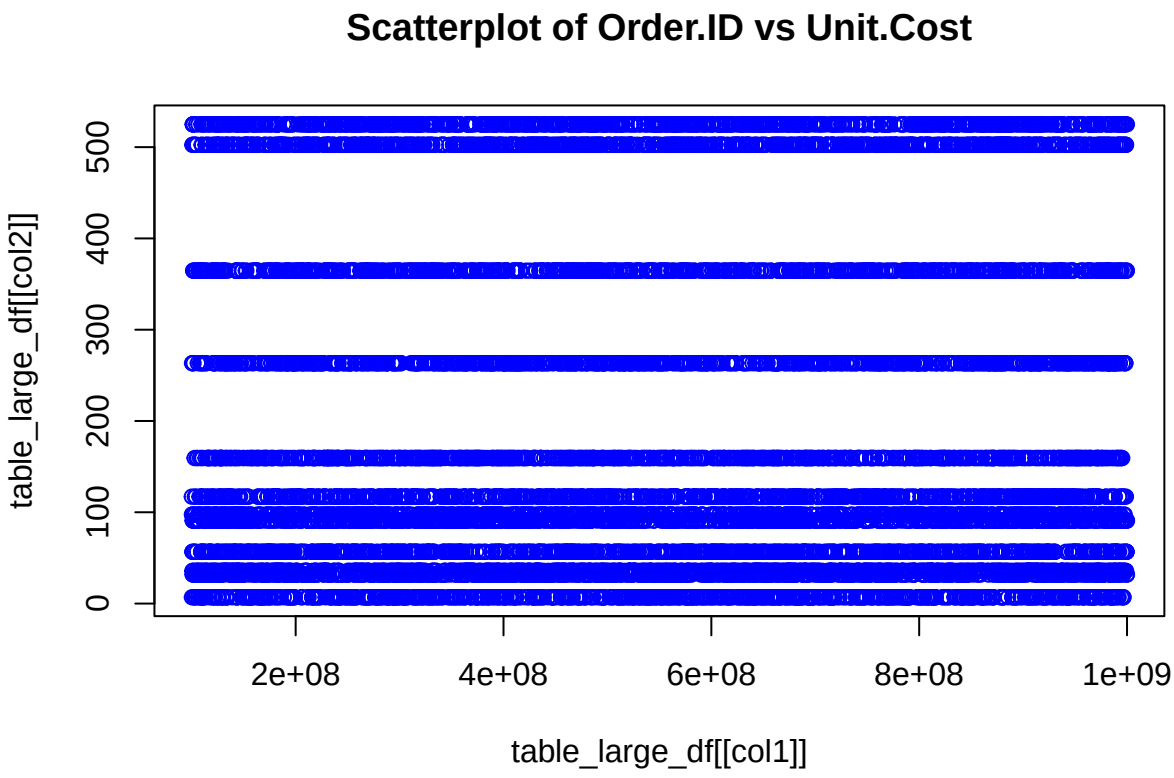
```r
# Assuming your dataframe is named 'table_small_df'
# Identify numeric columns
numeric_columns <- sapply(table_large_df, is.numeric)

# Create individual scatterplots for pairs of numeric columns using plot
for (col1 in names(table_large_df)[numeric_columns]) {
  for (col2 in names(table_large_df)[numeric_columns]) {
    if (col1 != col2) {
      plot(table_large_df[[col1]], table_large_df[[col2]], main = paste("Scatterplot of", col1, "vs", co
```

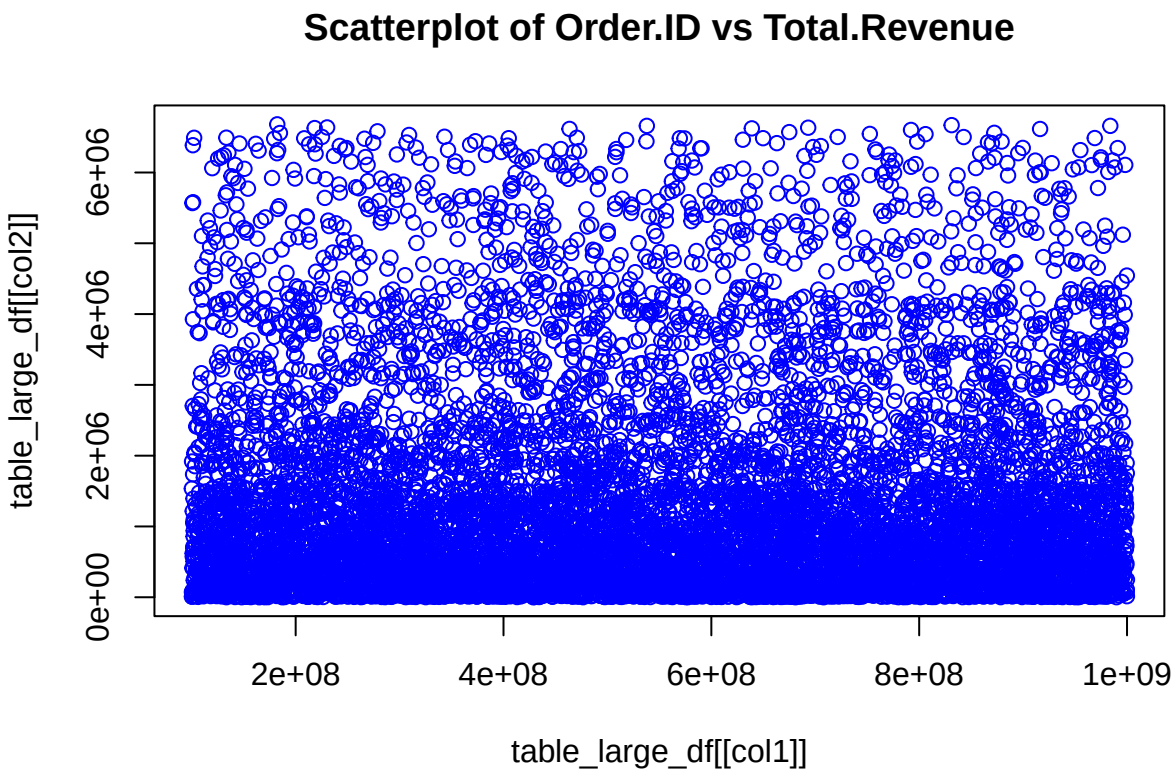
```
    }
  }
}
```

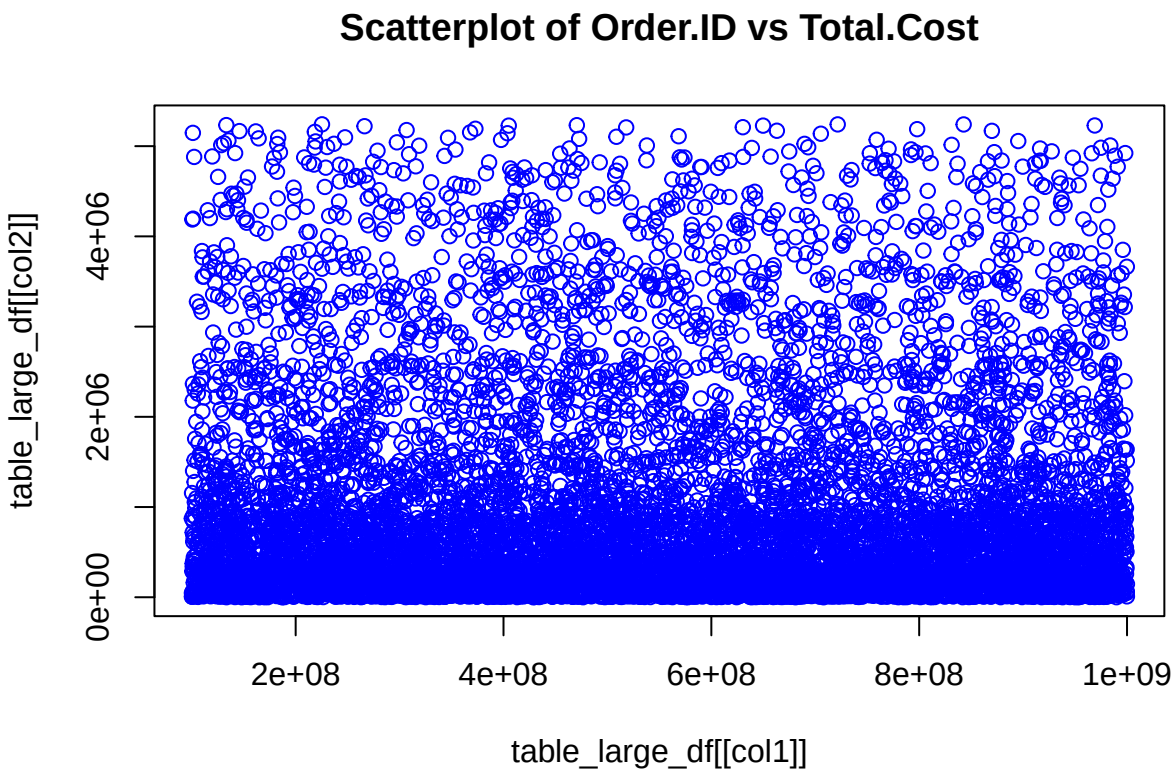


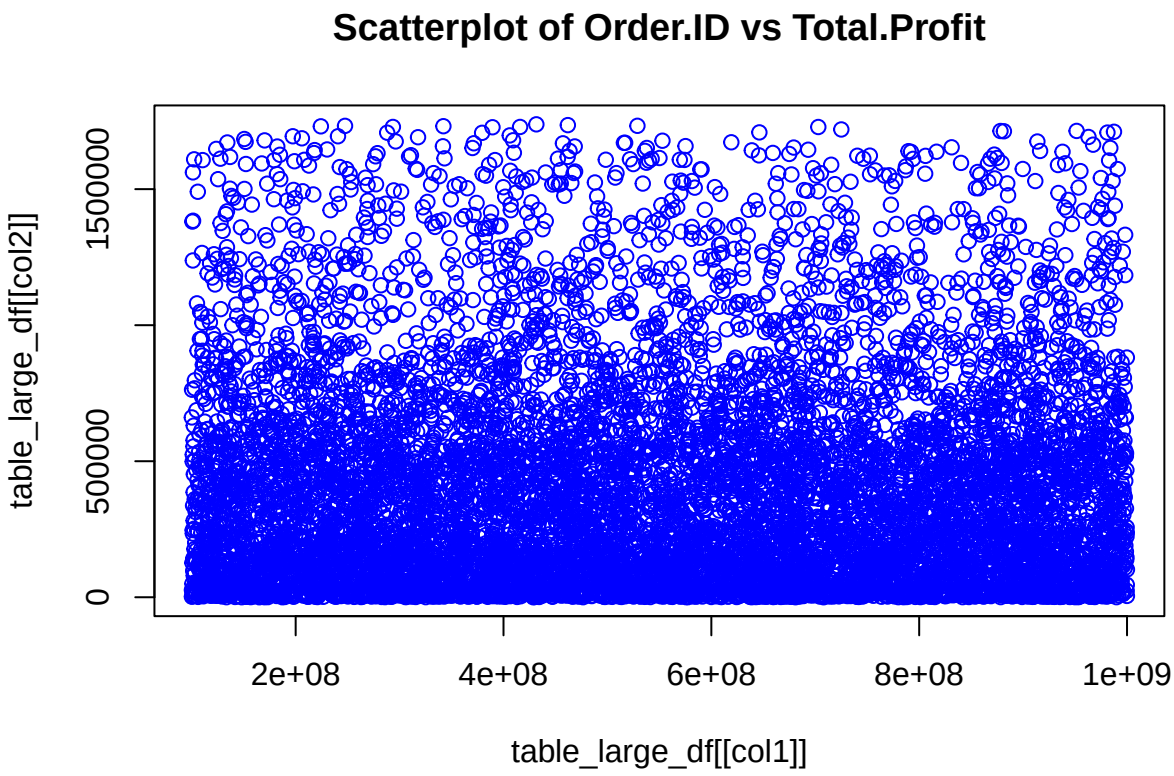
Individual Scatter Plots to explore relationships for each pair of variables.

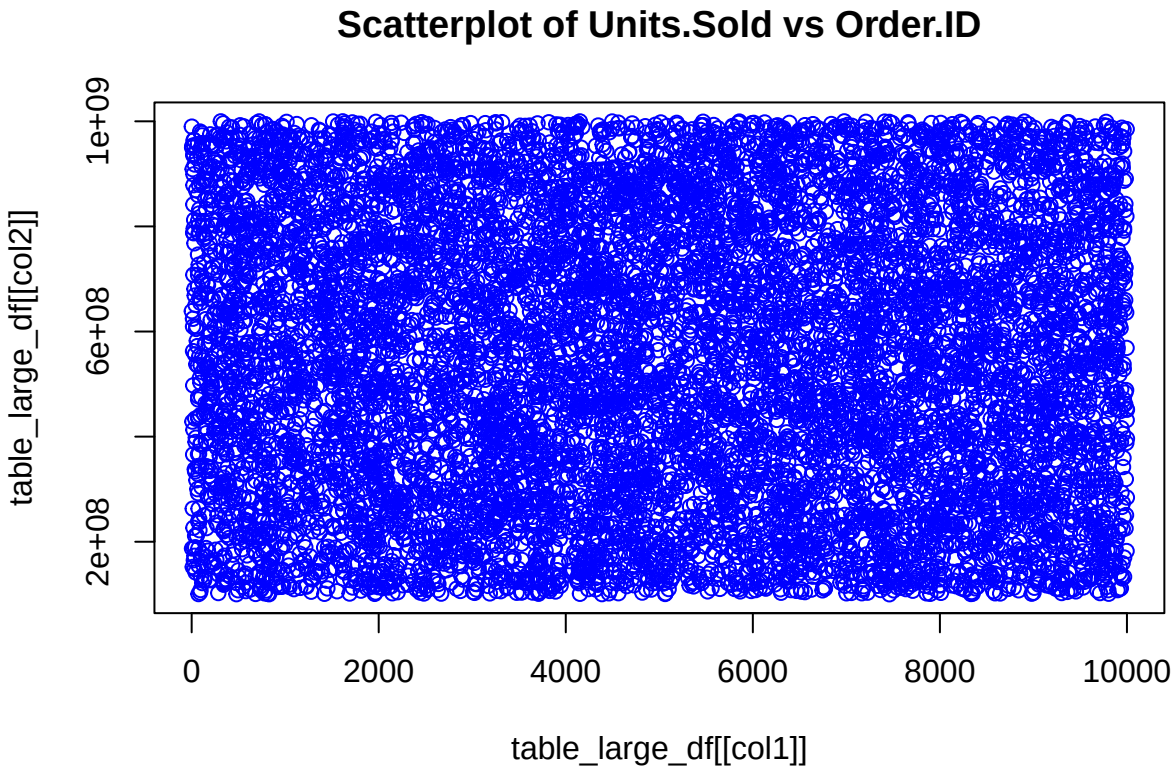


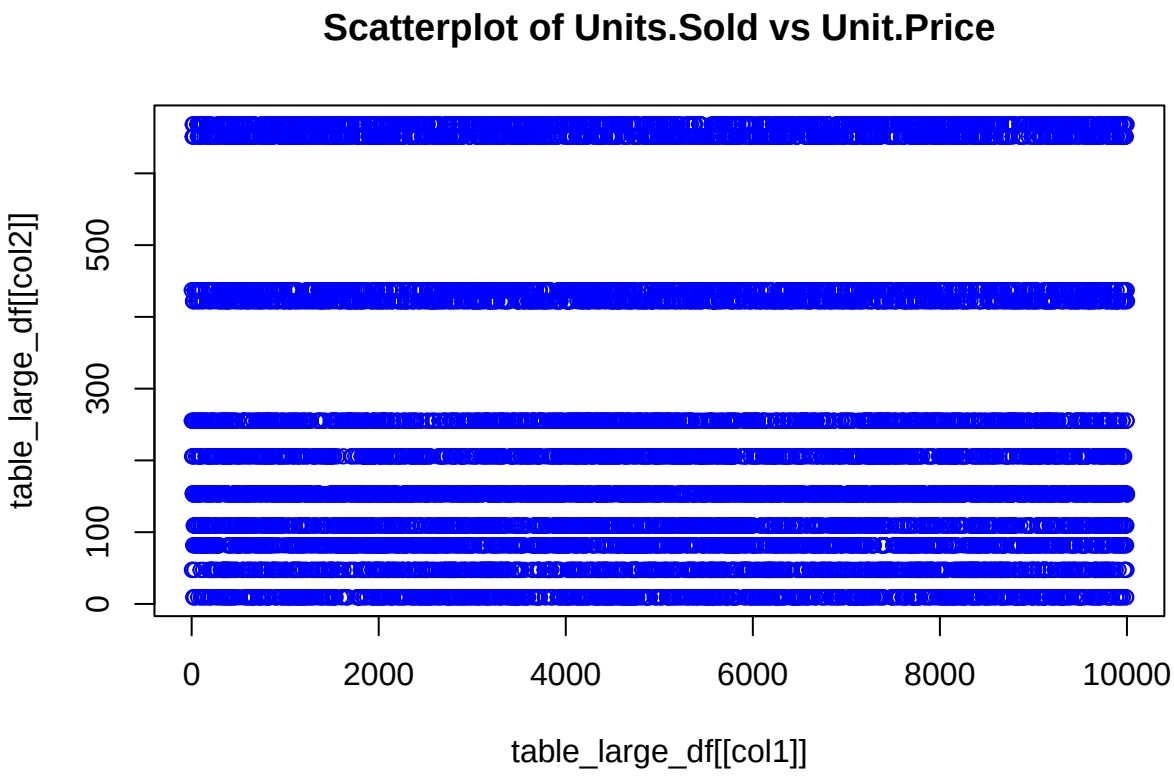

Scatterplot of Order.ID vs Unit.Cost
table_large_df[[col2]]
0
100
200
300
400
500
2e+08
4e+08
6e+08
8e+08
1e+09
table_large_df[[col1]]

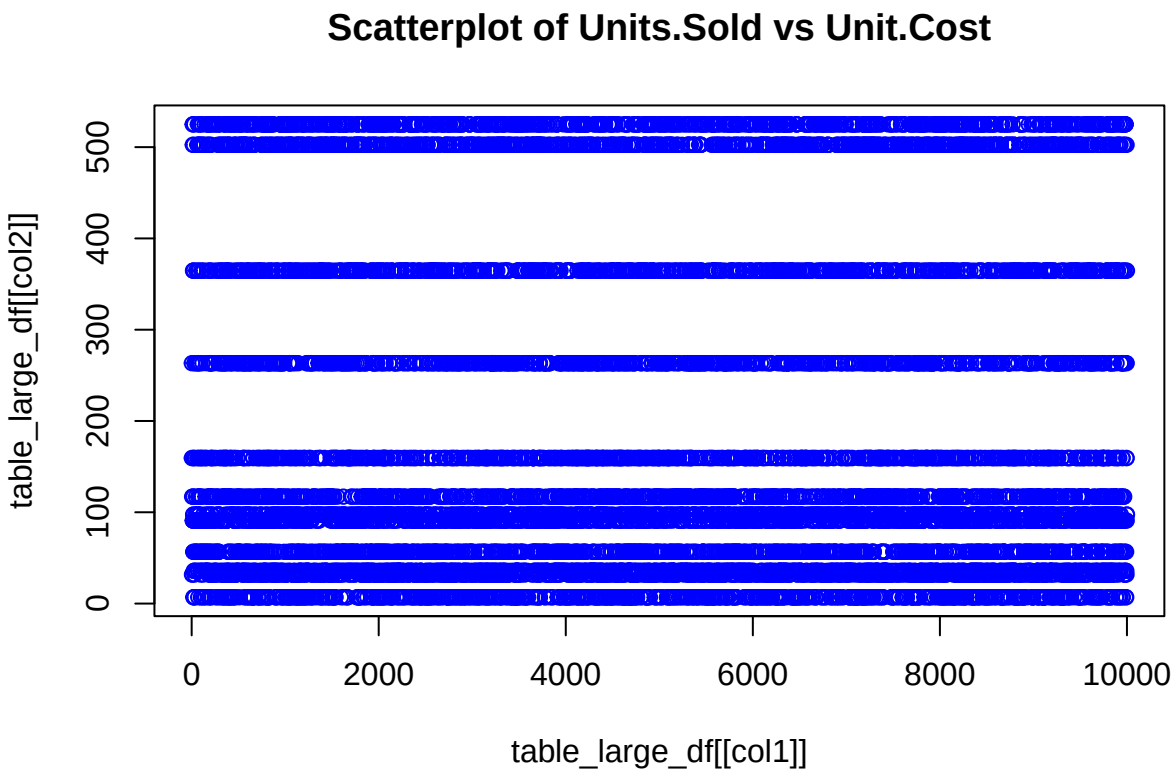

Scatterplot of Order.ID vs Total.Revenue
table_large_df[[col2]]
0e+00
2e+06
4e+06
6e+06
2e+08
4e+08
6e+08
8e+08
1e+09
table_large_df[[col1]]

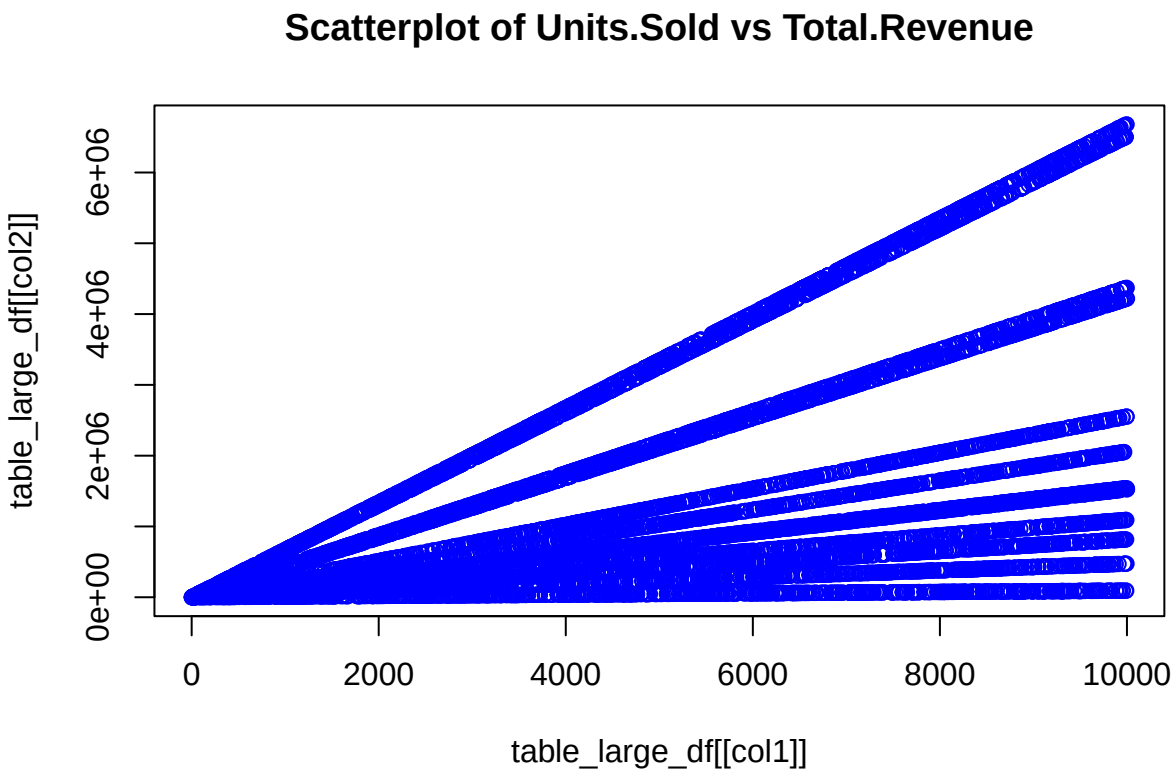

Scatterplot of Order.ID vs Total.Cost
table_large_df[[col2]]
0e+00
2e+06
4e+06
2e+08
4e+08
6e+08
8e+08
1e+09
table_large_df[[col1]]

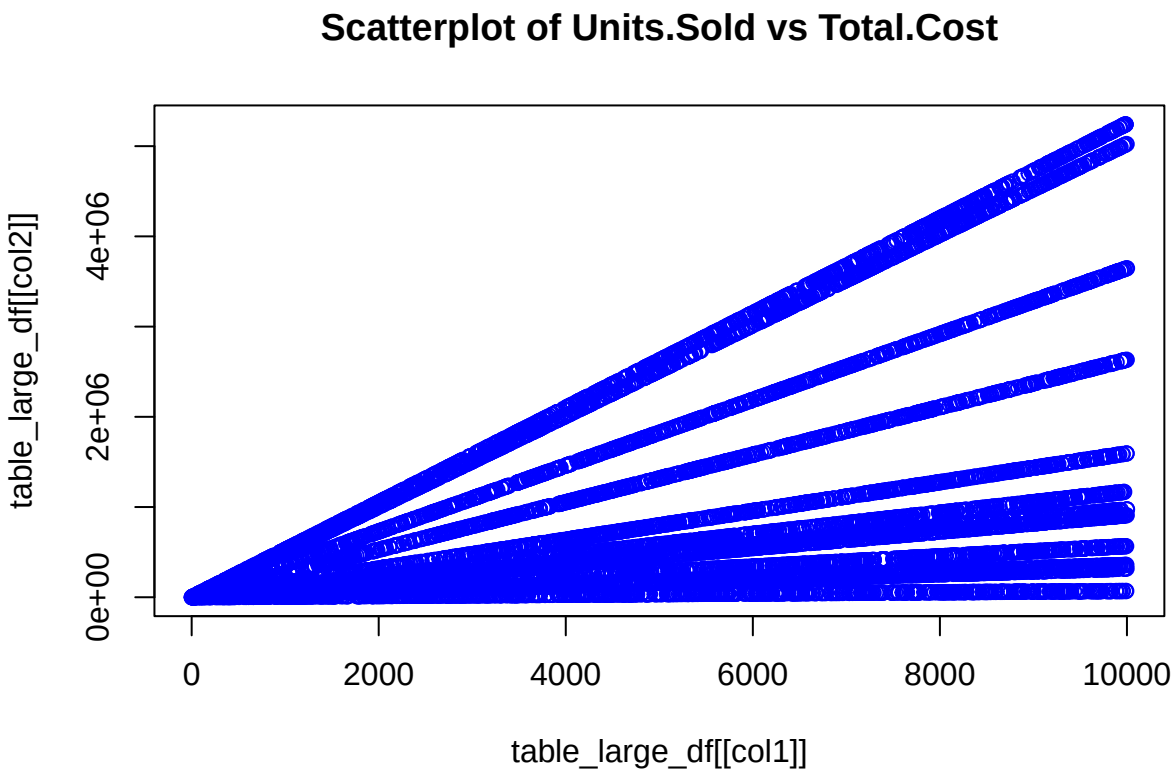

Scatterplot of Order.ID vs Total.Profit
table_large_df[[col2]]
0
500000
1500000
2e+08
4e+08
6e+08
8e+08
1e+09
table_large_df[[col1]]

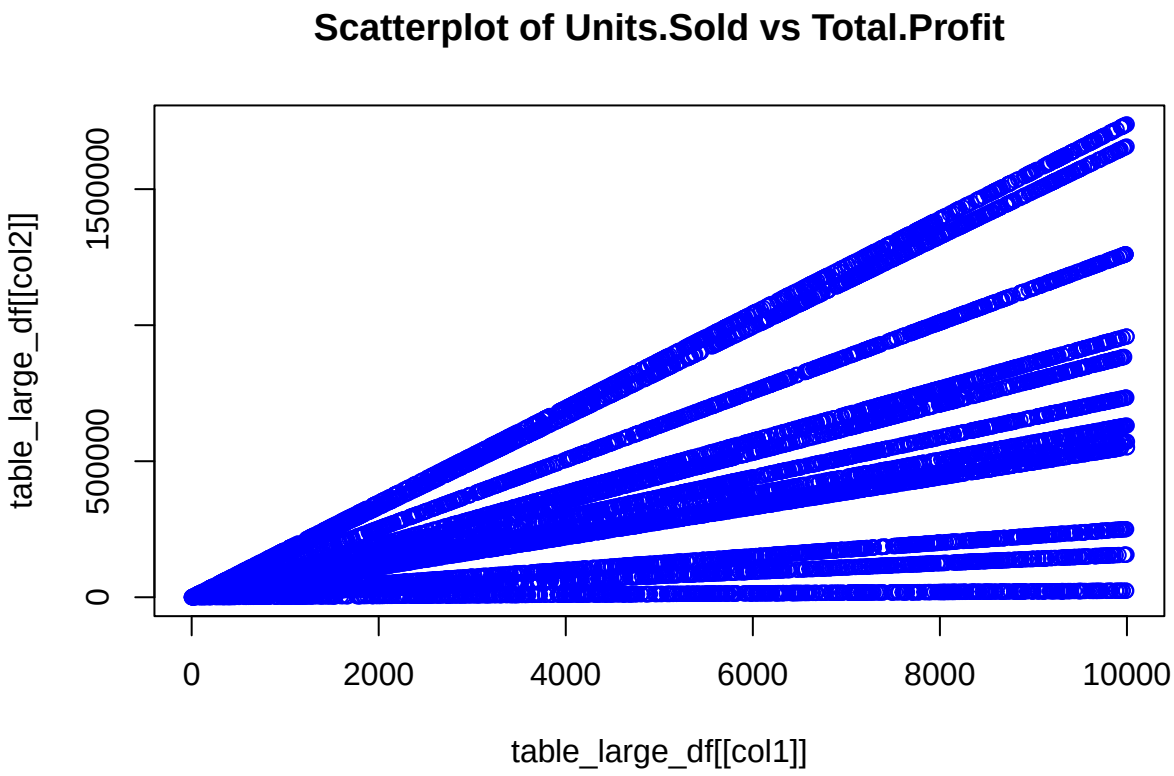

Scatterplot of Units.Sold vs Order.ID
table_large_df[[col2]]
2e+08
6e+08
1e+09
0
2000
4000
6000
8000
10000
table_large_df[[col1]]

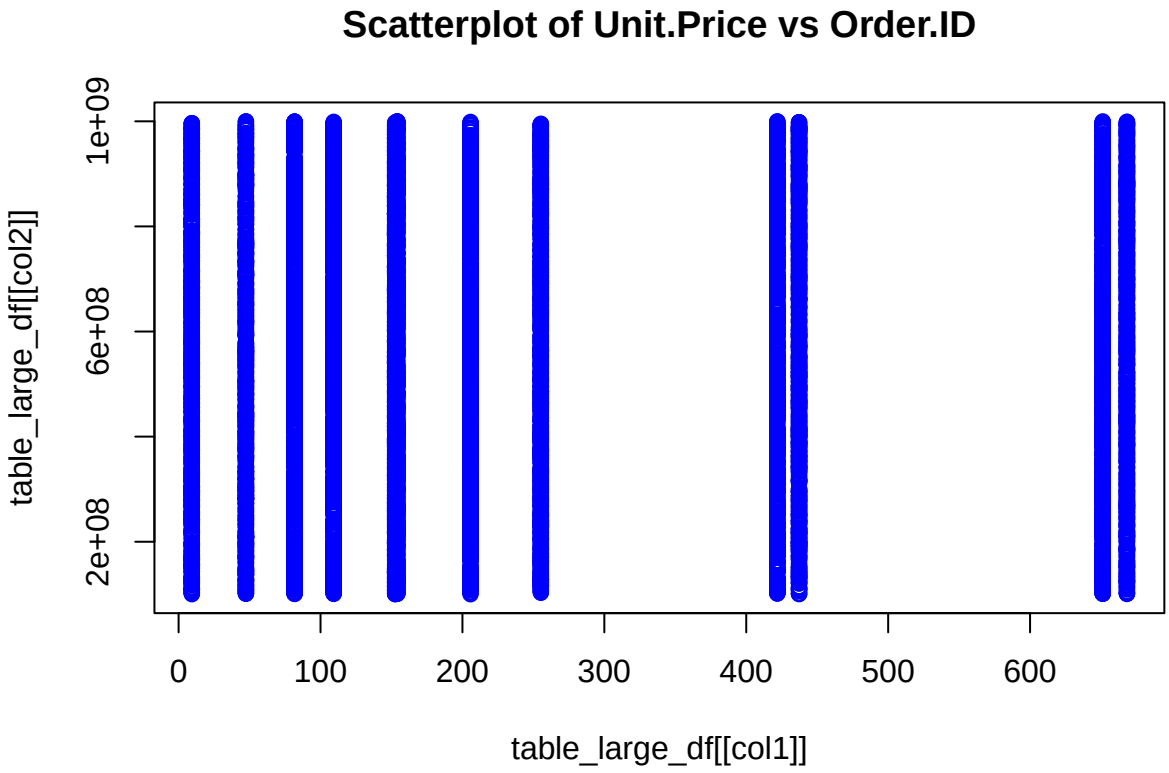

Scatterplot of Units.Sold vs Unit.Price
table_large_df[[col2]]
0
100
300
500
0
2000
4000
6000
8000
10000
table_large_df[[col1]]

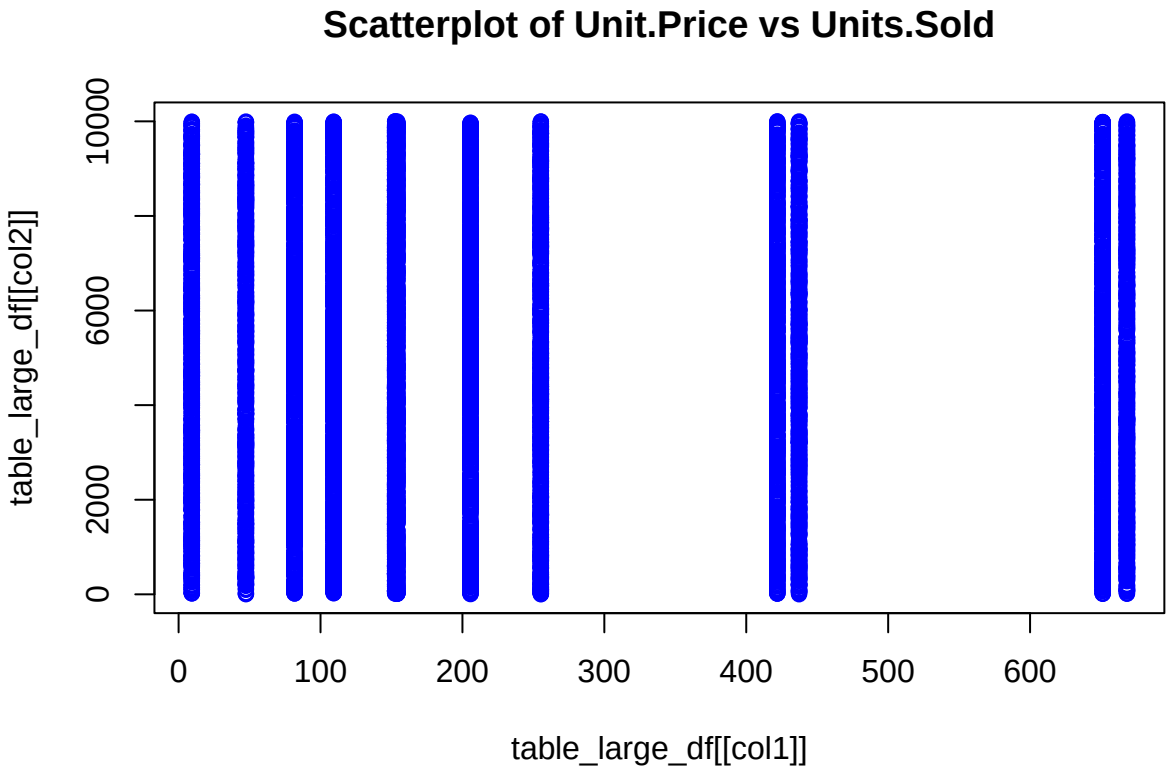

Scatterplot of Units.Sold vs Unit.Cost
table_large_df[[col2]]
0
100
200
300
400
500
0
2000
4000
6000
8000
10000
table_large_df[[col1]]

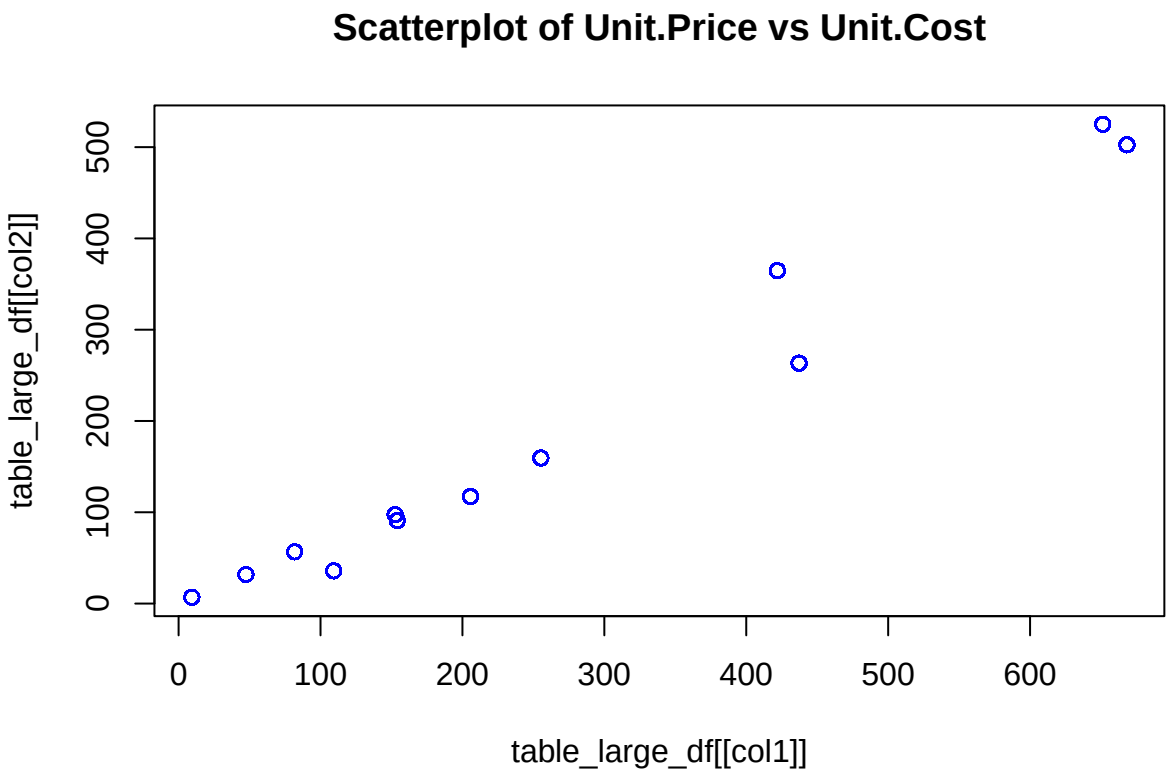

Scatterplot of Units.Sold vs Total.Revenue
table_large_df[[col2]]
0e+00
2e+06
4e+06
6e+06
0
2000
4000
6000
8000
10000
table_large_df[[col1]]

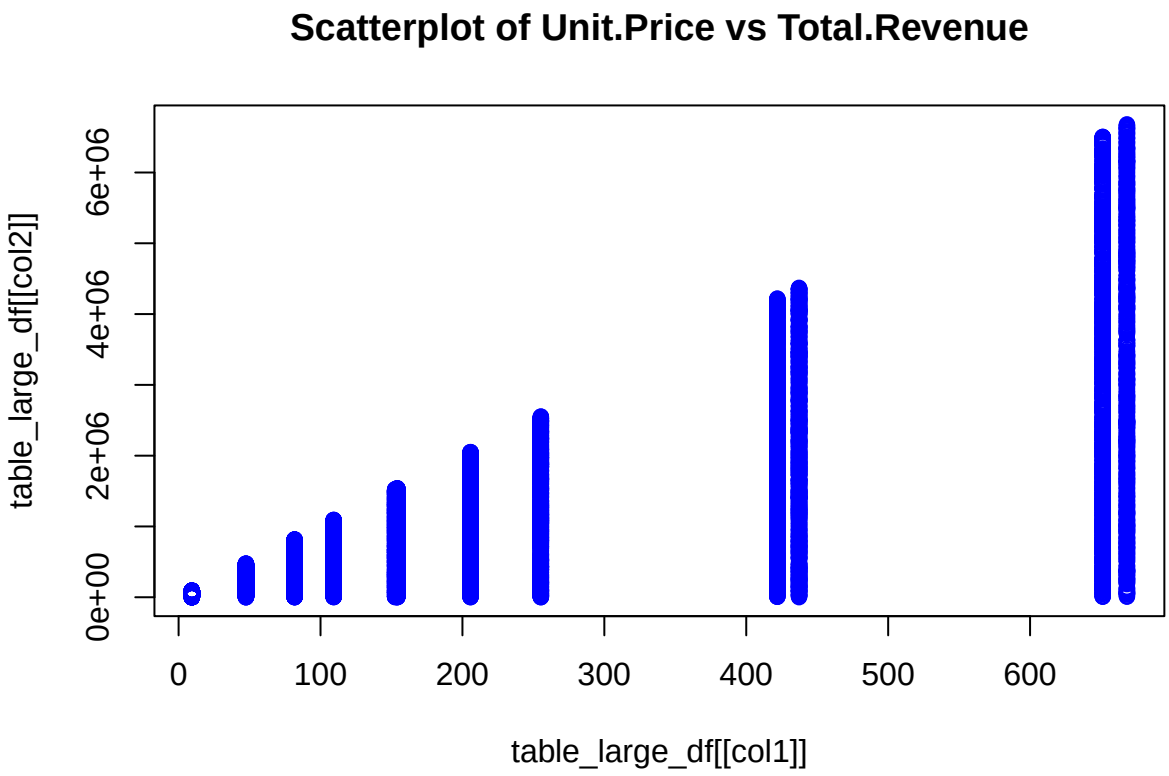

Scatterplot of Units.Sold vs Total.Cost
table_large_df[[col2]]
0e+00
2e+06
4e+06
0
2000
4000
6000
8000
10000
table_large_df[[col1]]

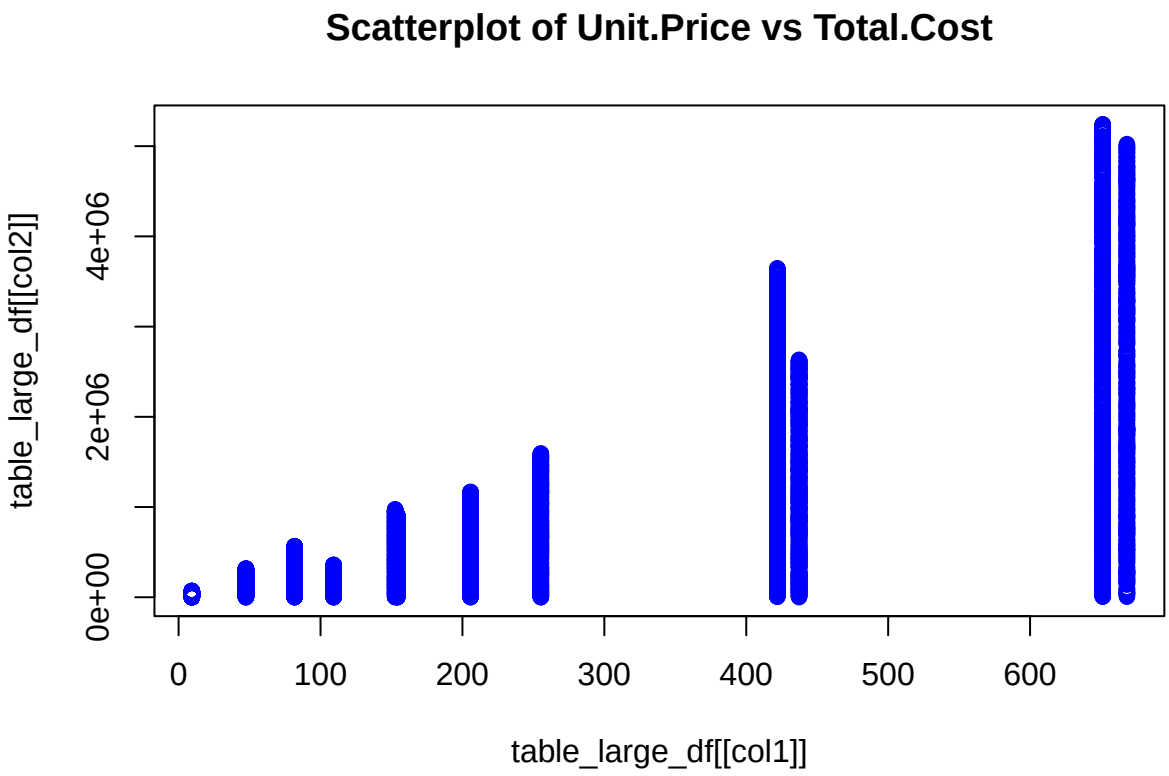

Scatterplot of Units.Sold vs Total.Profit
table_large_df[[col2]]
0
500000
1500000
0
2000
4000
6000
8000
10000
table_large_df[[col1]]

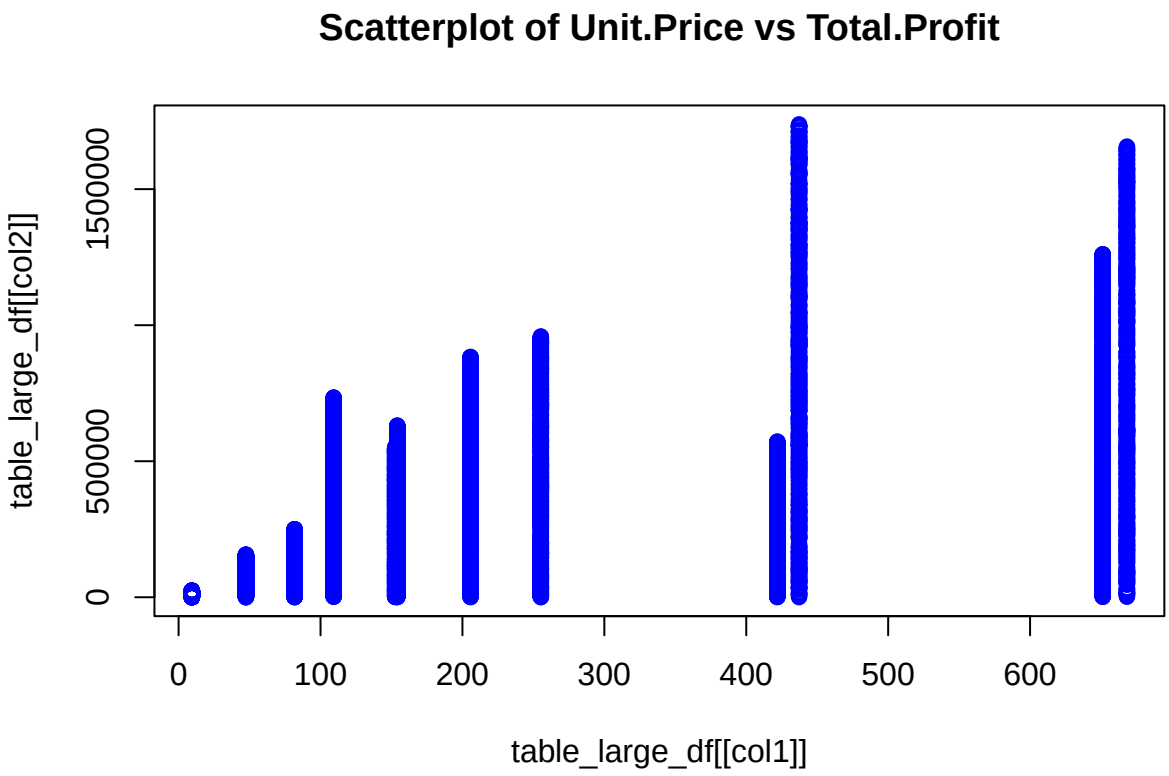

Scatterplot of Unit.Price vs Order.ID
table_large_df[[col2]]
1e+09
6e+08
2e+08
0
100
200
300
400
500
600
table_large_df[[col1]]

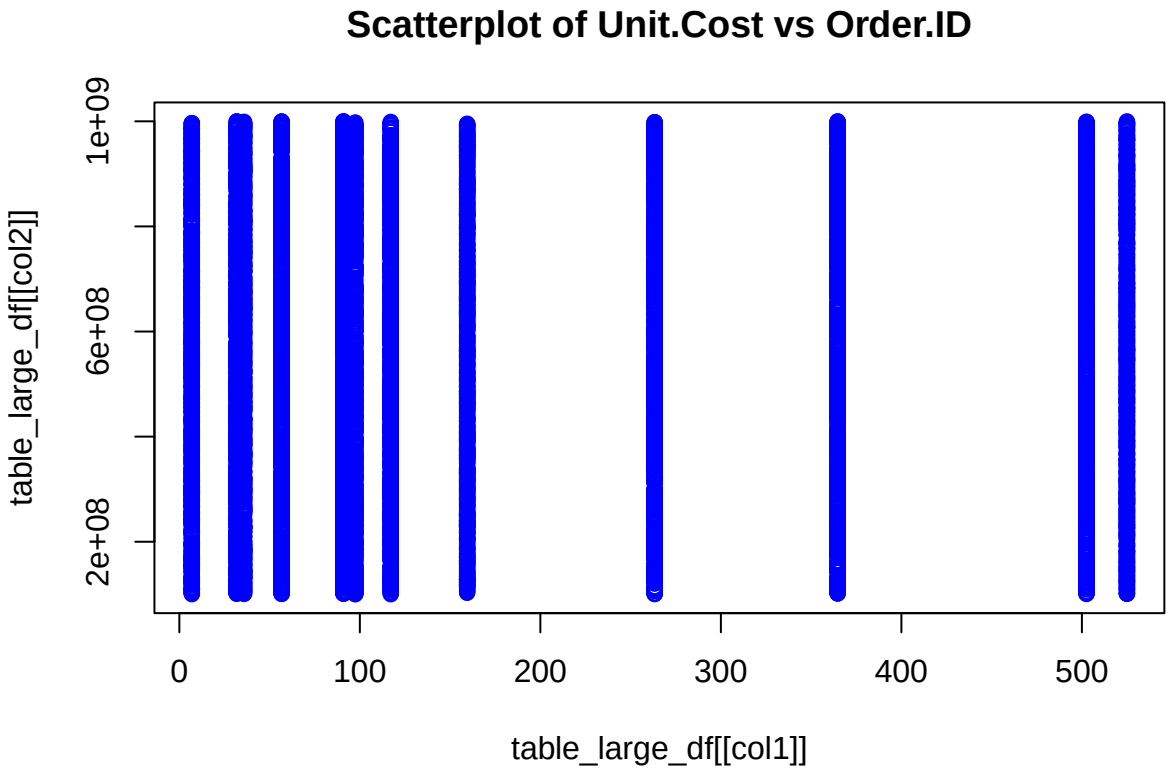

Scatterplot of Unit.Price vs Units.Sold
table_large_df[[col2]]
0
2000
6000
10000
0
100
200
300
400
500
600
table_large_df[[col1]]

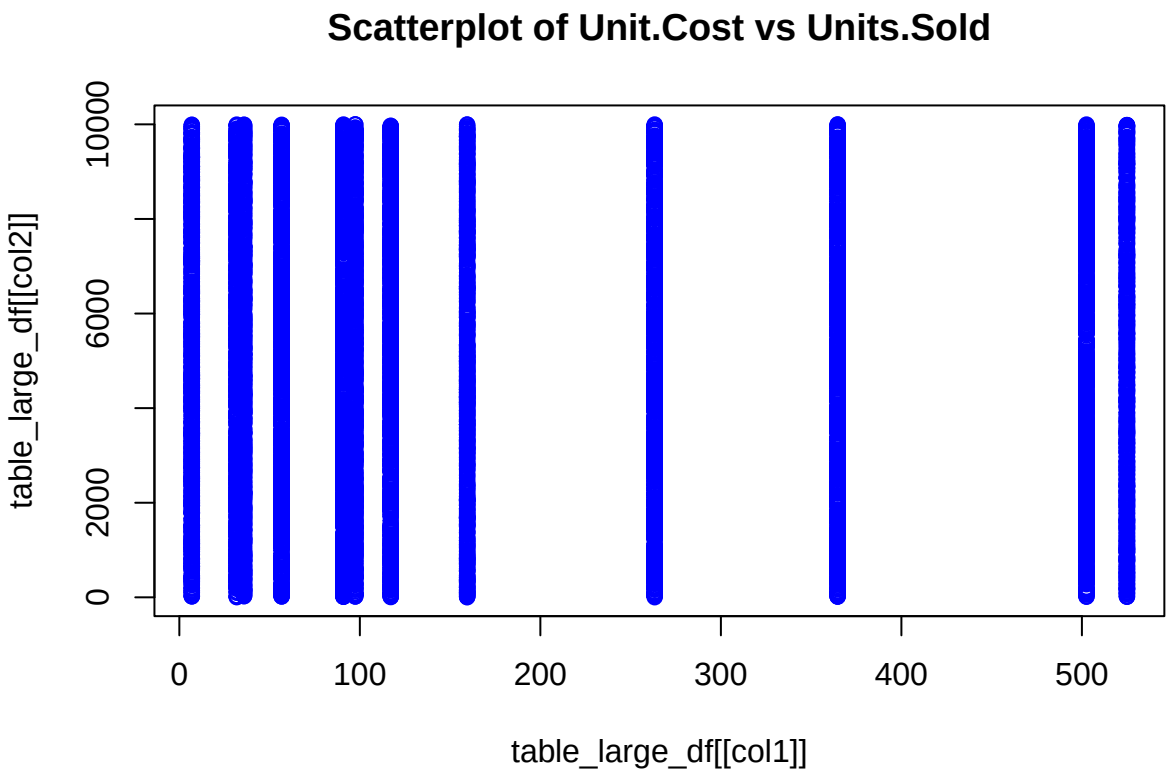

Scatterplot of Unit.Price vs Unit.Cost
table_large_df[[col2]]
0
100
200
300
400
500
table_large_df[[col1]]
0
100
200
300
400
500
600

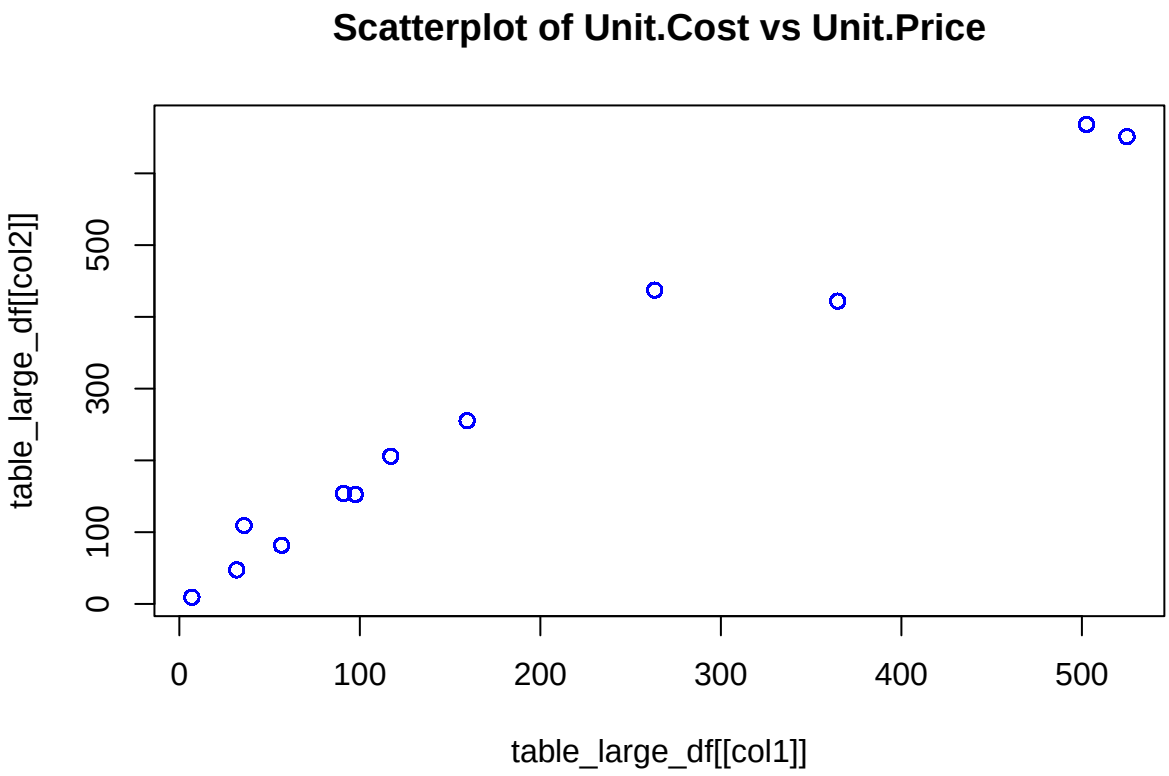

Scatterplot of Unit.Price vs Total.Revenue
table_large_df[[col2]]
0e+00
2e+06
4e+06
6e+06
0
100
200
300
400
500
600
table_large_df[[col1]]

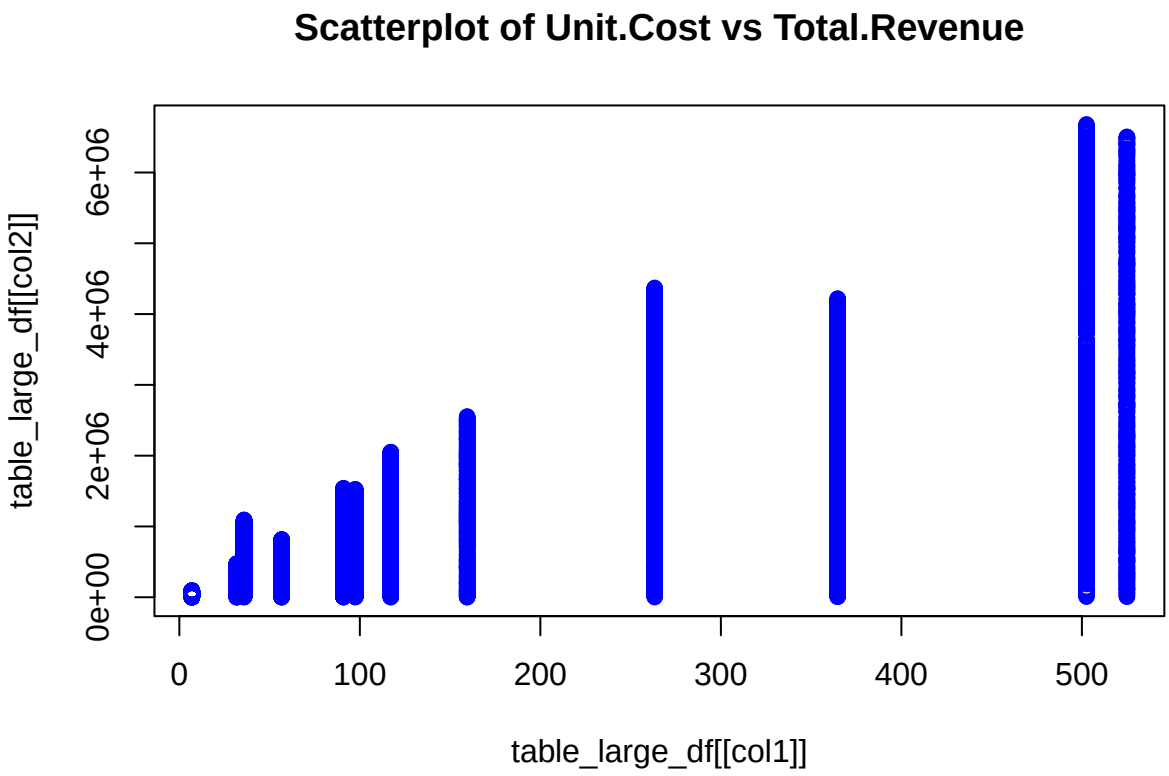

Scatterplot of Unit.Price vs Total.Cost
table_large_df[[col2]]
0e+00
2e+06
4e+06
0
100
200
300
400
500
600
table_large_df[[col1]]

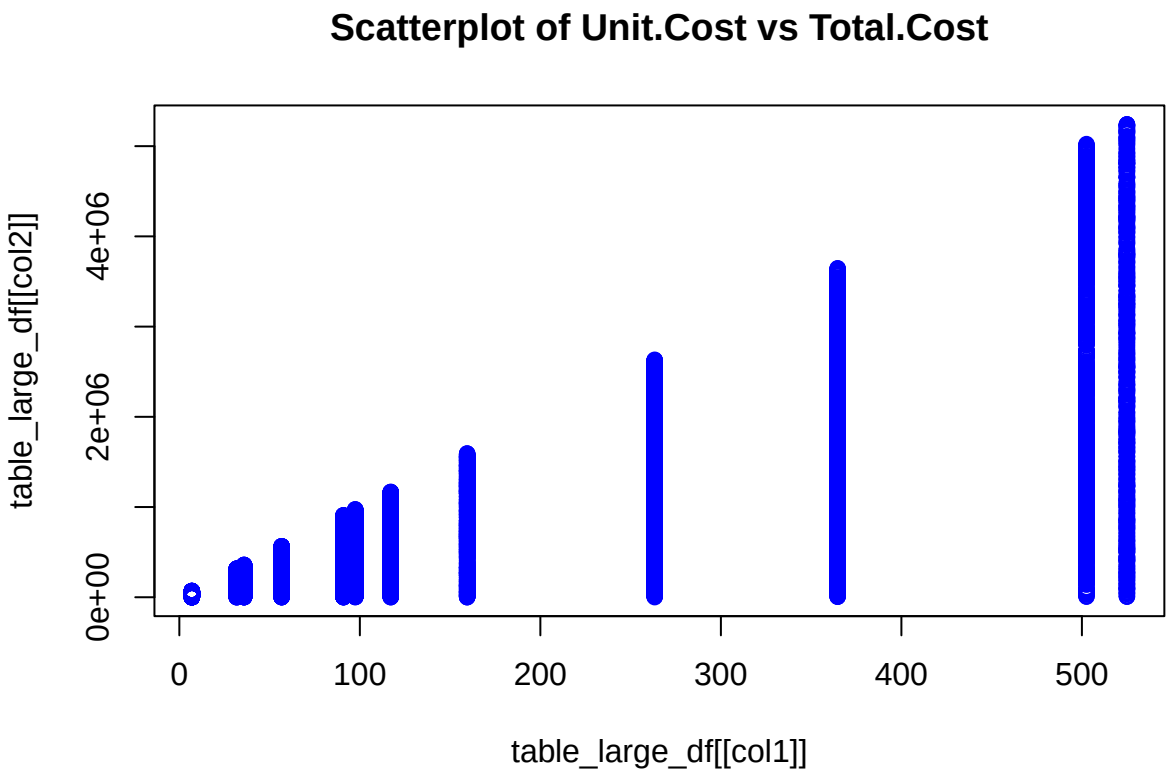

Scatterplot of Unit.Price vs Total.Profit
table_large_df[[col2]]
0
500000
1500000
0
100
200
300
400
500
600
table_large_df[[col1]]

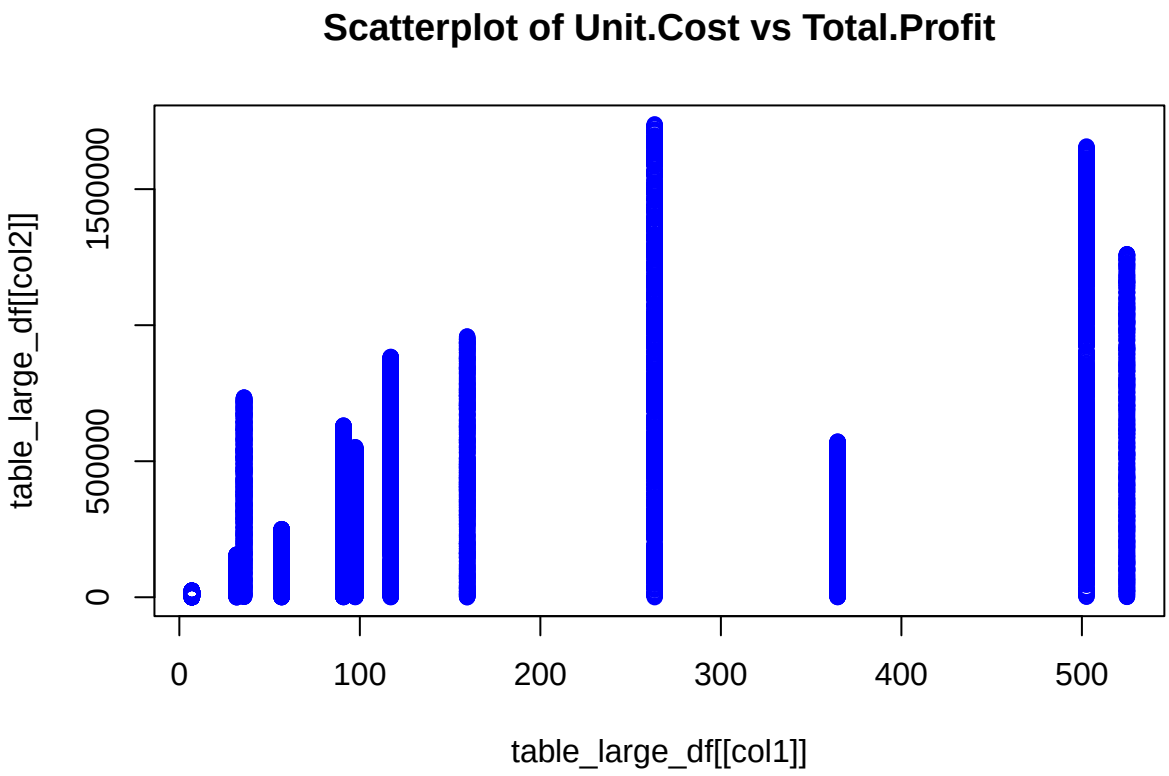

Scatterplot of Unit.Cost vs Order.ID
table_large_df[[col2]]
2e+08
6e+08
1e+09
0
100
200
300
400
500
table_large_df[[col1]]

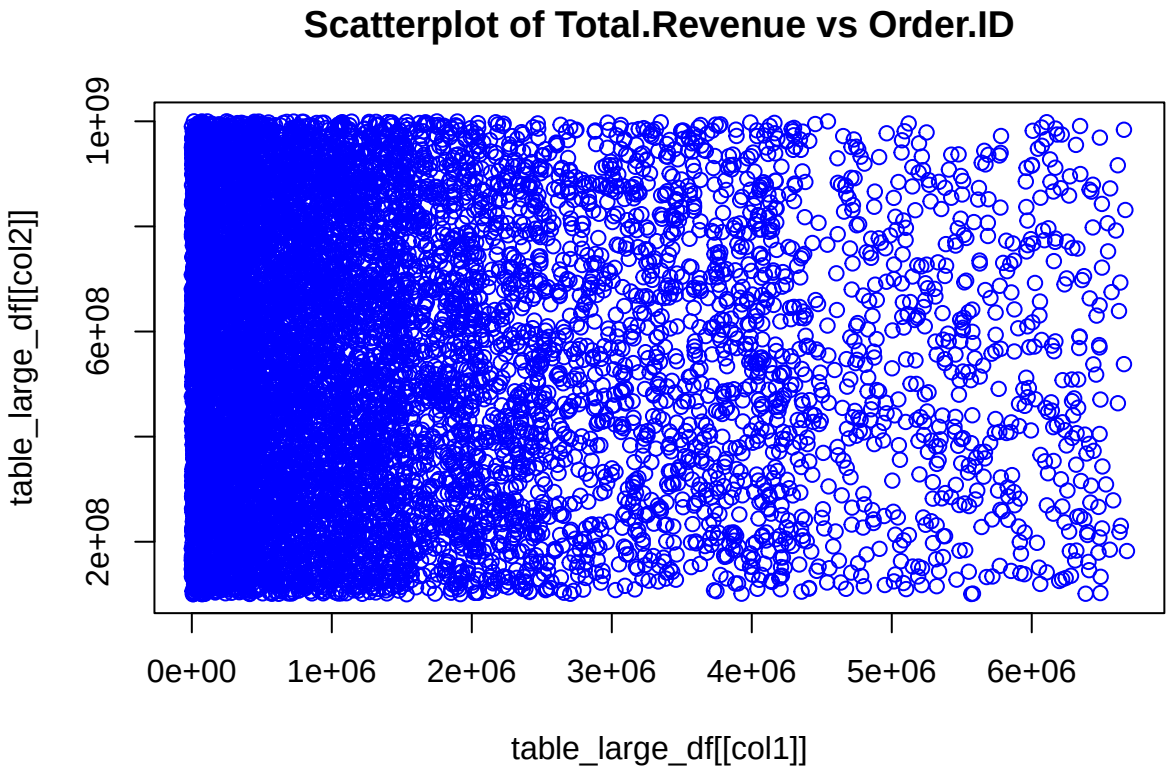

Scatterplot of Unit.Cost vs Units.Sold
table_large_df[[col2]]
0
2000
6000
10000
0
100
200
300
400
500
table_large_df[[col1]]

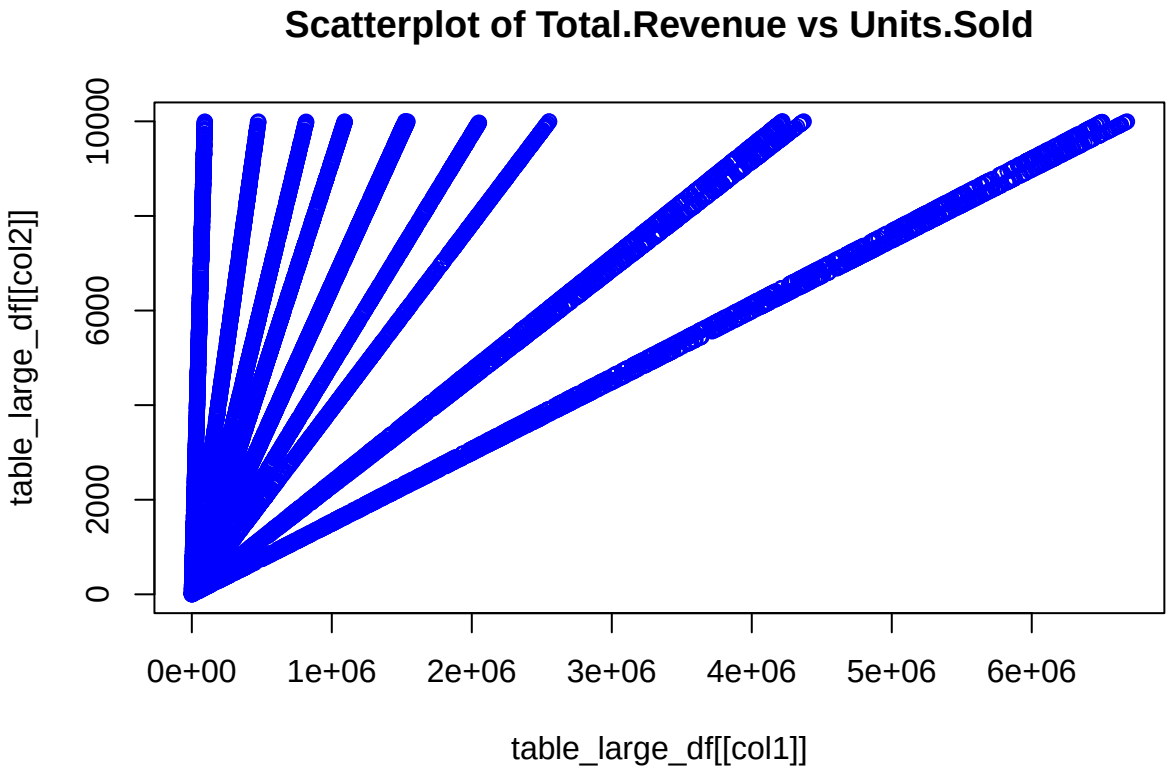

Scatterplot of Unit.Cost vs Unit.Price
table_large_df[[col2]]
0
100
300
500
0
100
200
300
400
500
table_large_df[[col1]]

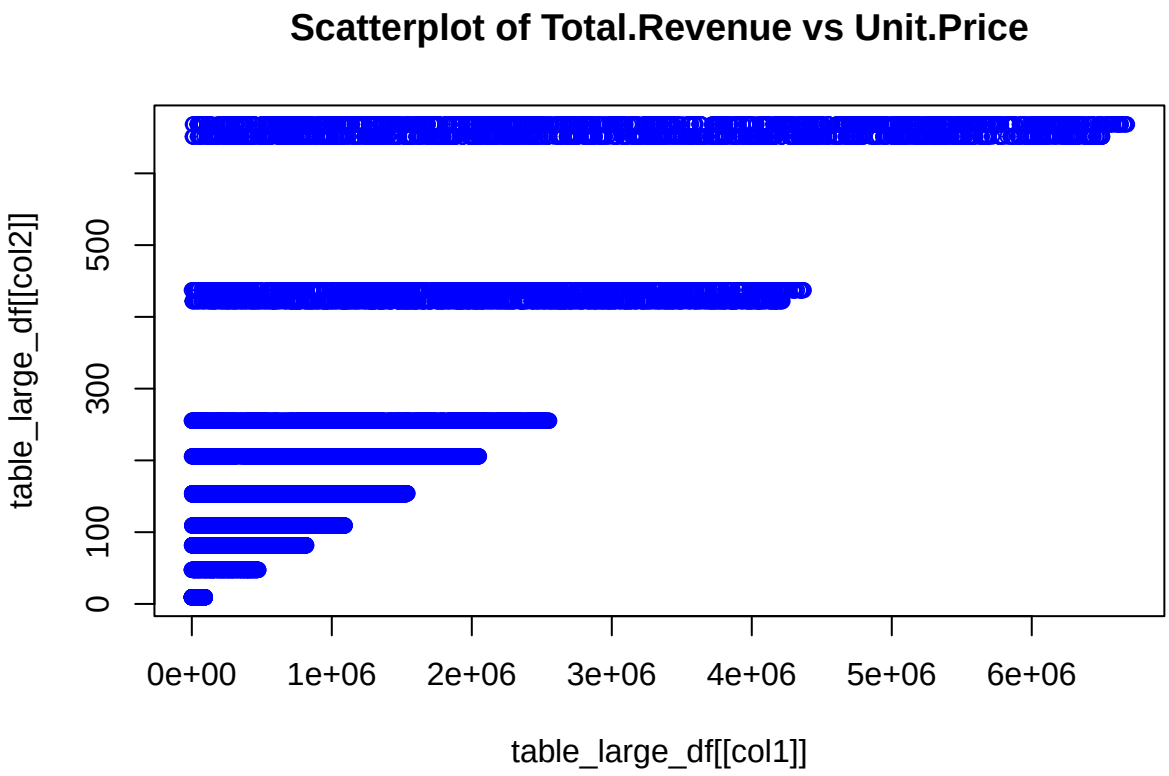

Scatterplot of Unit.Cost vs Total.Revenue
table_large_df[[col2]]
0e+00
2e+06
4e+06
6e+06
0
100
200
300
400
500
table_large_df[[col1]]

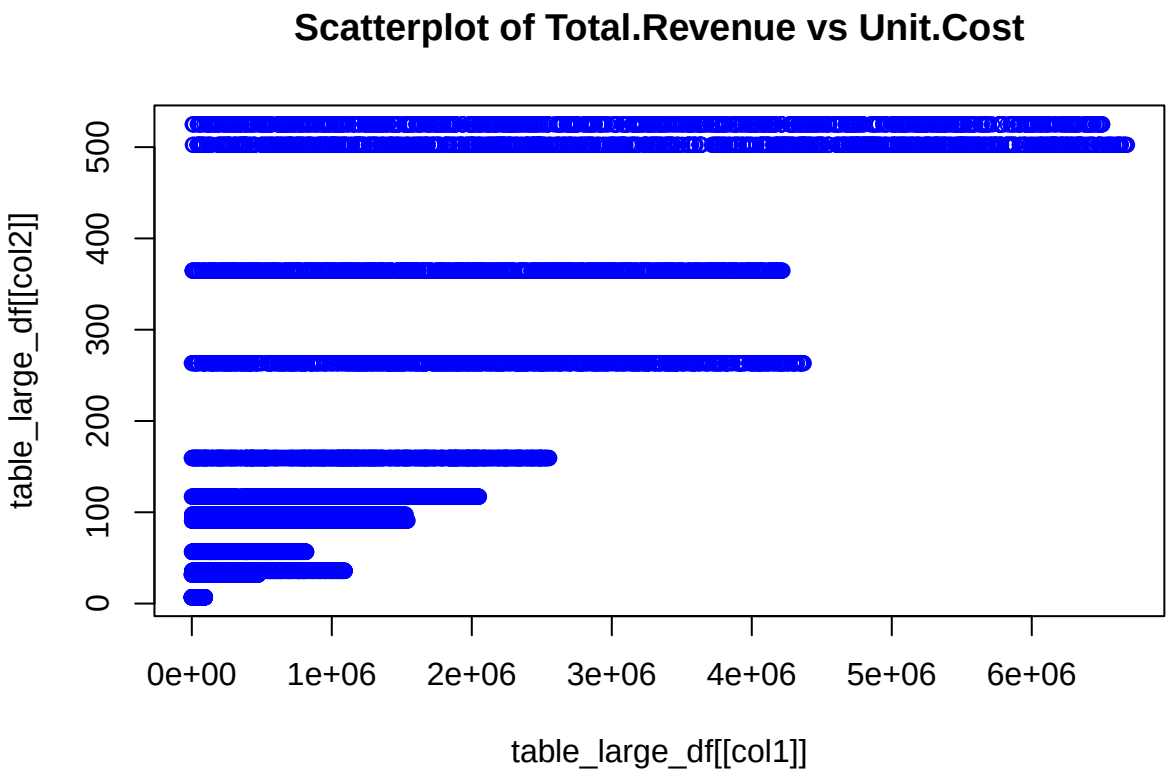

Scatterplot of Unit.Cost vs Total.Cost
table_large_df[[col2]]
0e+00
2e+06
4e+06
0
100
200
300
400
500
table_large_df[[col1]]

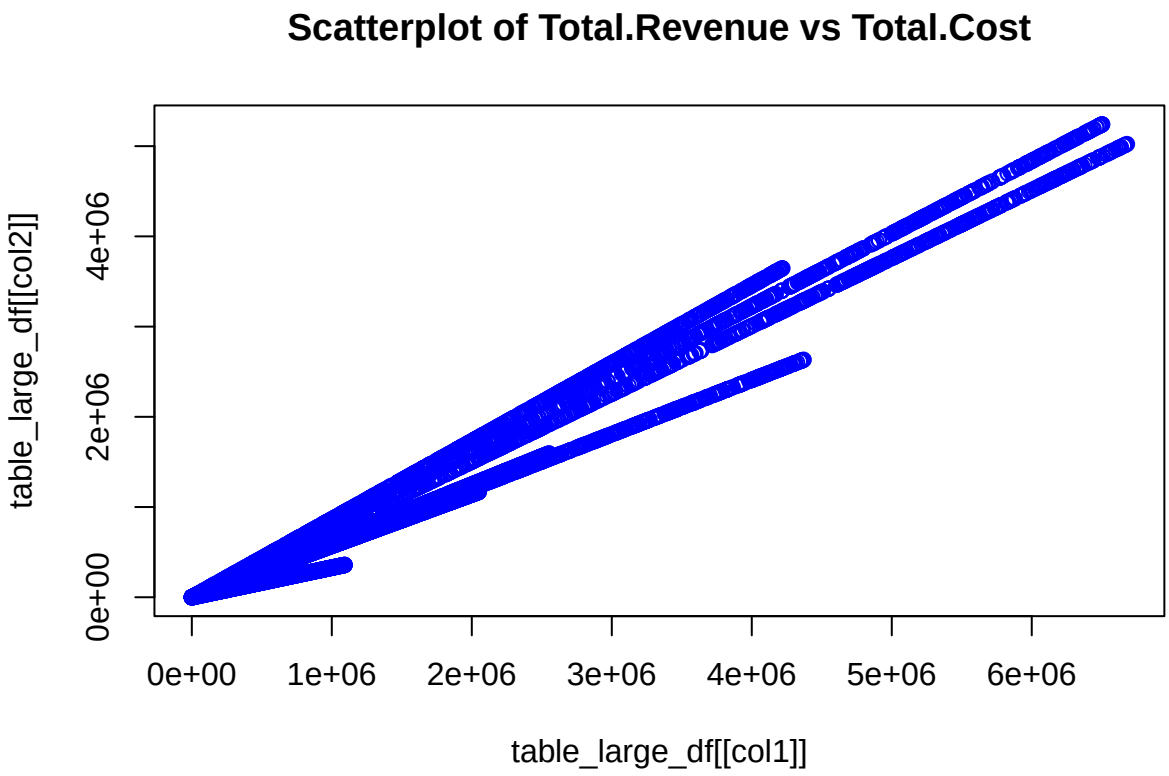

Scatterplot of Unit.Cost vs Total.Profit
table_large_df[[col2]]
0
500000
1500000
0
100
200
300
400
500
table_large_df[[col1]]

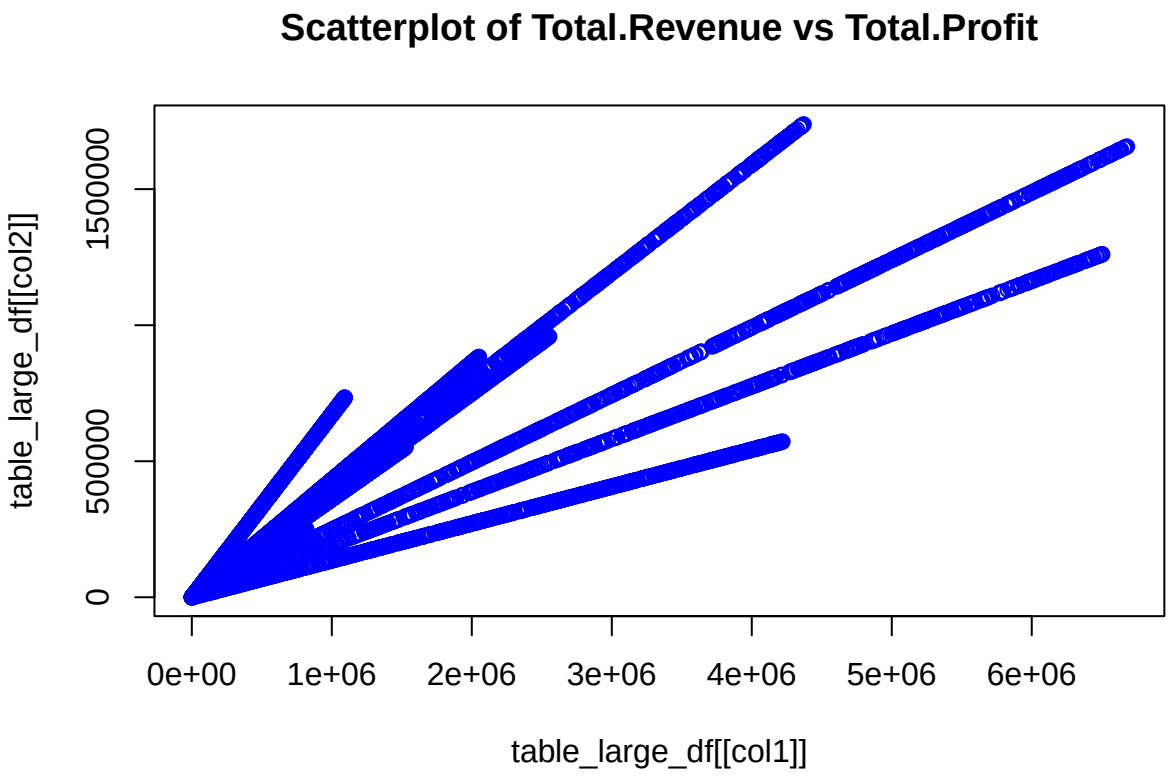

Scatterplot of Total.Revenue vs Order.ID
table_large_df[[col2]]
1e+09
6e+08
2e+08
0e+00
1e+06
2e+06
3e+06
4e+06
5e+06
6e+06
table_large_df[[col1]]

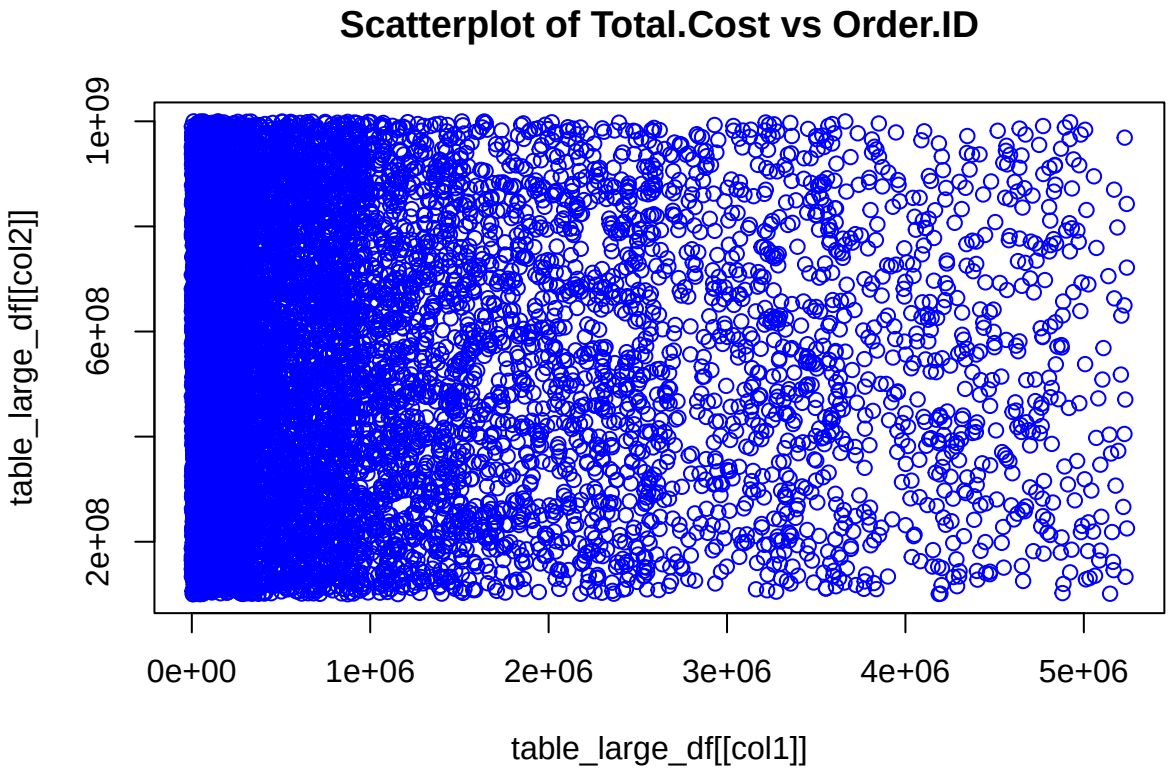

Scatterplot of Total.Revenue vs Units.Sold
table_large_df[[col2]]
0
2000
6000
10000
0e+00
1e+06
2e+06
3e+06
4e+06
5e+06
6e+06
table_large_df[[col1]]

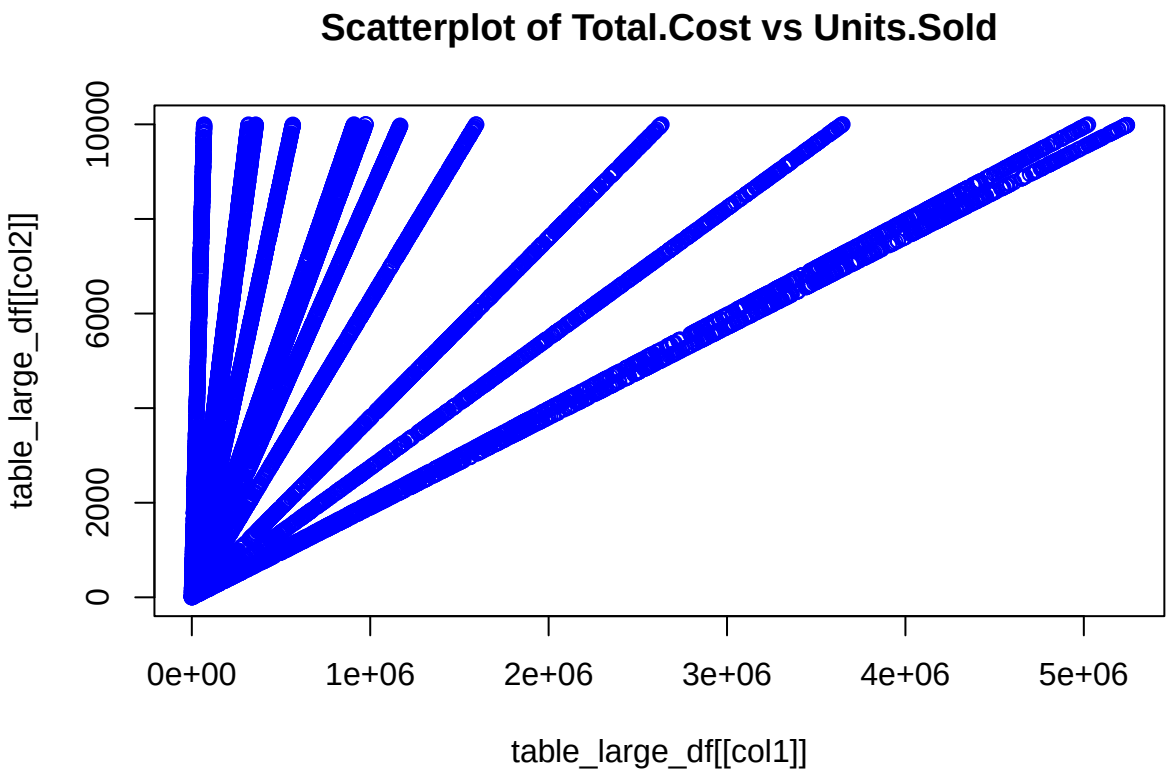

Scatterplot of Total.Revenue vs Unit.Price
table_large_df[[col2]]
0
100
300
500
0e+00
1e+06
2e+06
3e+06
4e+06
5e+06
6e+06
table_large_df[[col1]]

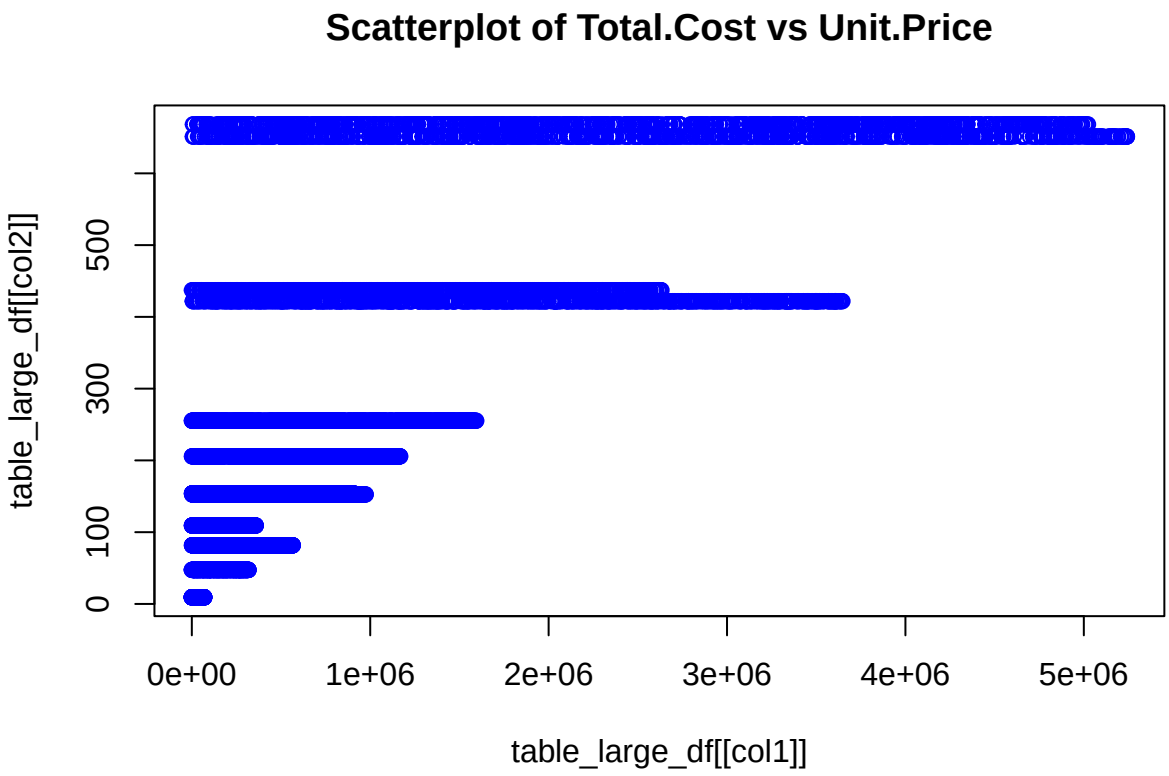

Scatterplot of Total.Revenue vs Unit.Cost
table_large_df[[col2]]
0
100
200
300
400
500
0e+00
1e+06
2e+06
3e+06
4e+06
5e+06
6e+06
table_large_df[[col1]]

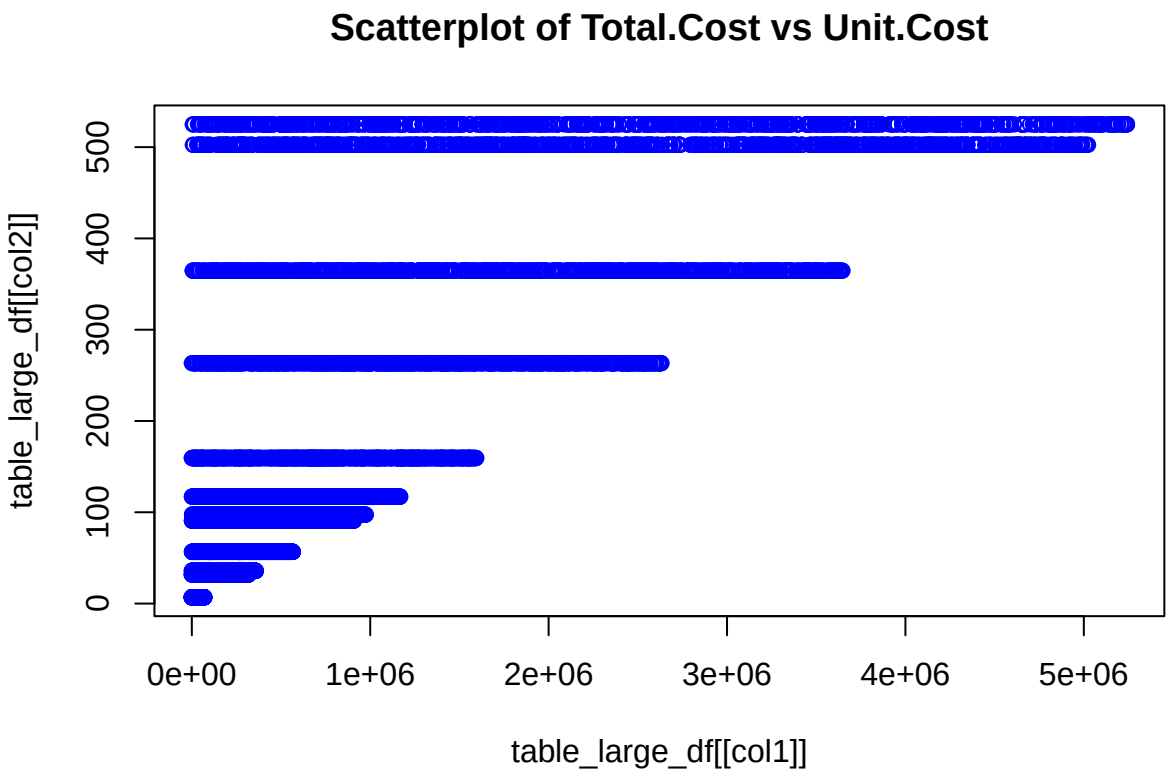

Scatterplot of Total.Revenue vs Total.Cost
table_large_df[[col2]]
0e+00
2e+06
4e+06
0e+00
1e+06
2e+06
3e+06
4e+06
5e+06
6e+06
table_large_df[[col1]]

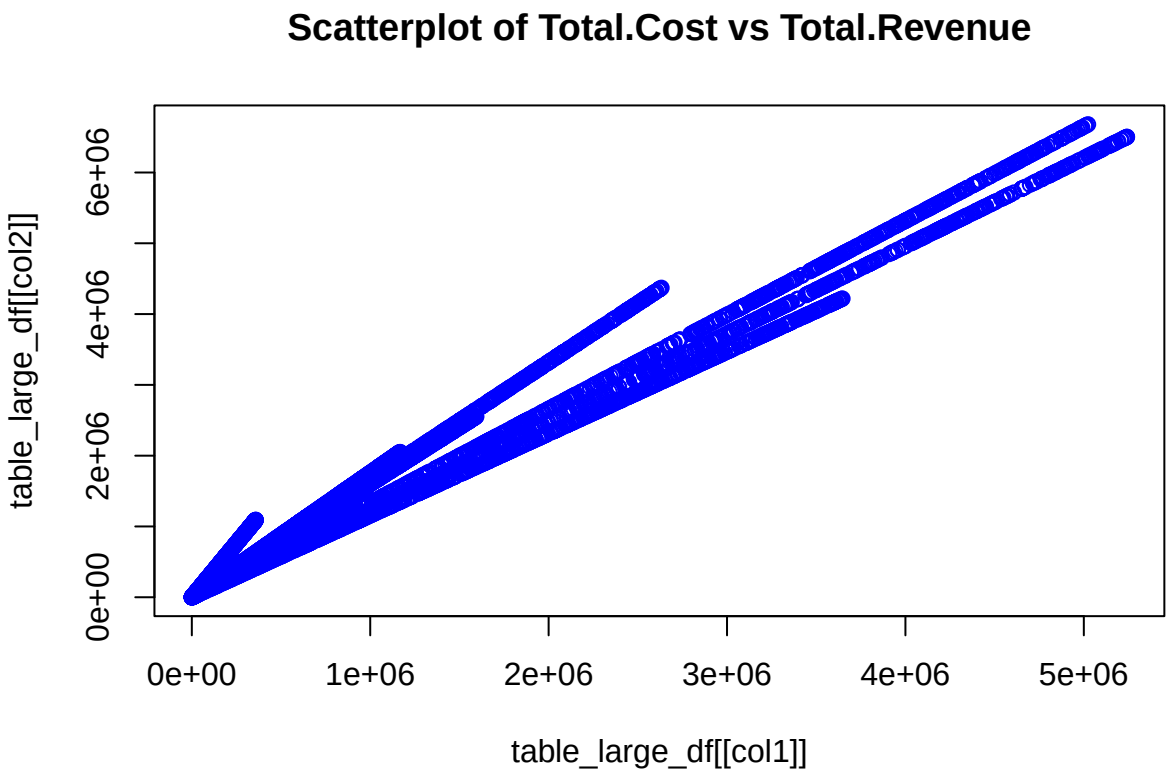

Scatterplot of Total.Revenue vs Total.Profit
table_large_df[[col2]]
0
500000
1500000
0e+00
1e+06
2e+06
3e+06
4e+06
5e+06
6e+06
table_large_df[[col1]]

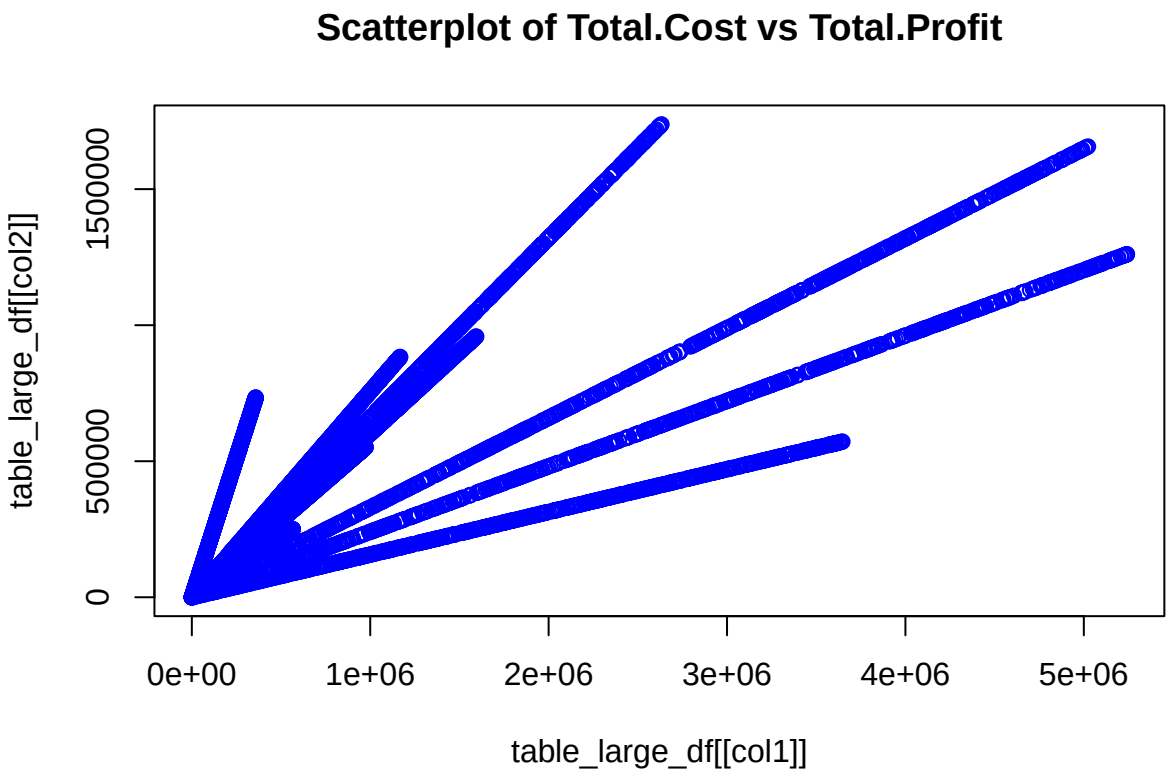

Scatterplot of Total.Cost vs Order.ID
table_large_df[[col2]]
1e+09
6e+08
2e+08
0e+00
1e+06
2e+06
3e+06
4e+06
5e+06
table_large_df[[col1]]

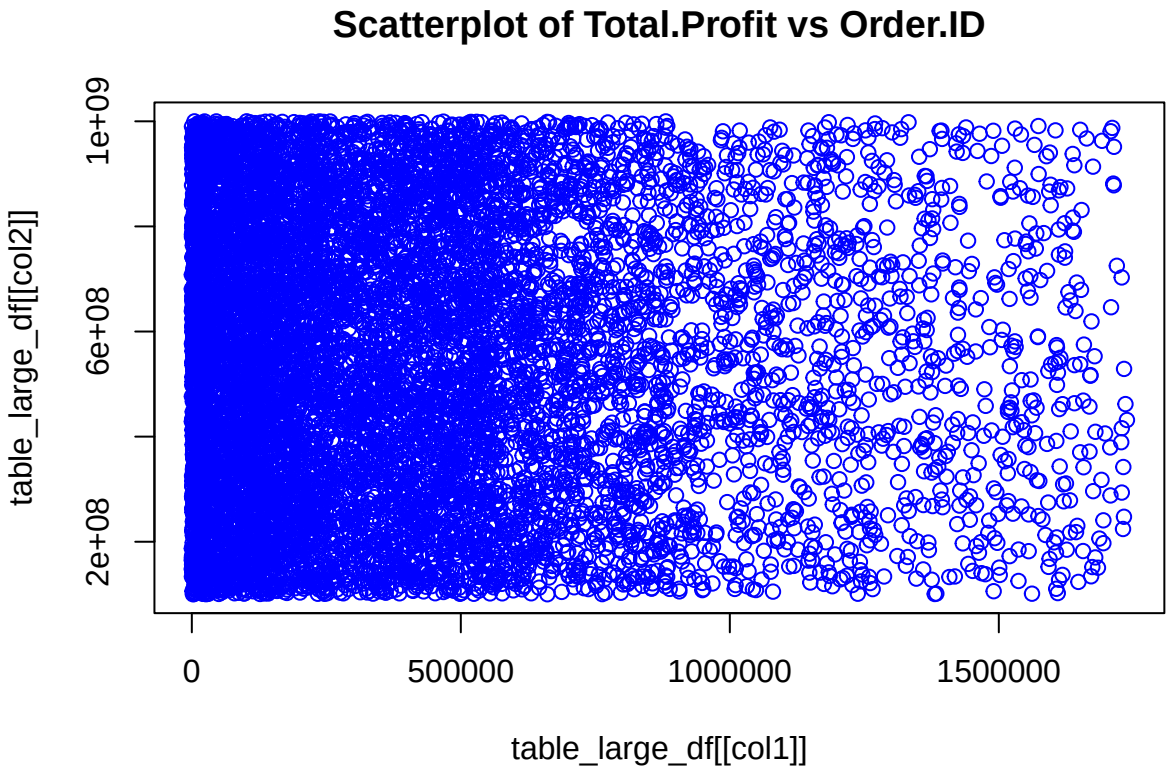

Scatterplot of Total.Cost vs Units.Sold
table_large_df[[col2]]
0
2000
6000
10000
0e+00
1e+06
2e+06
3e+06
4e+06
5e+06
table_large_df[[col1]]

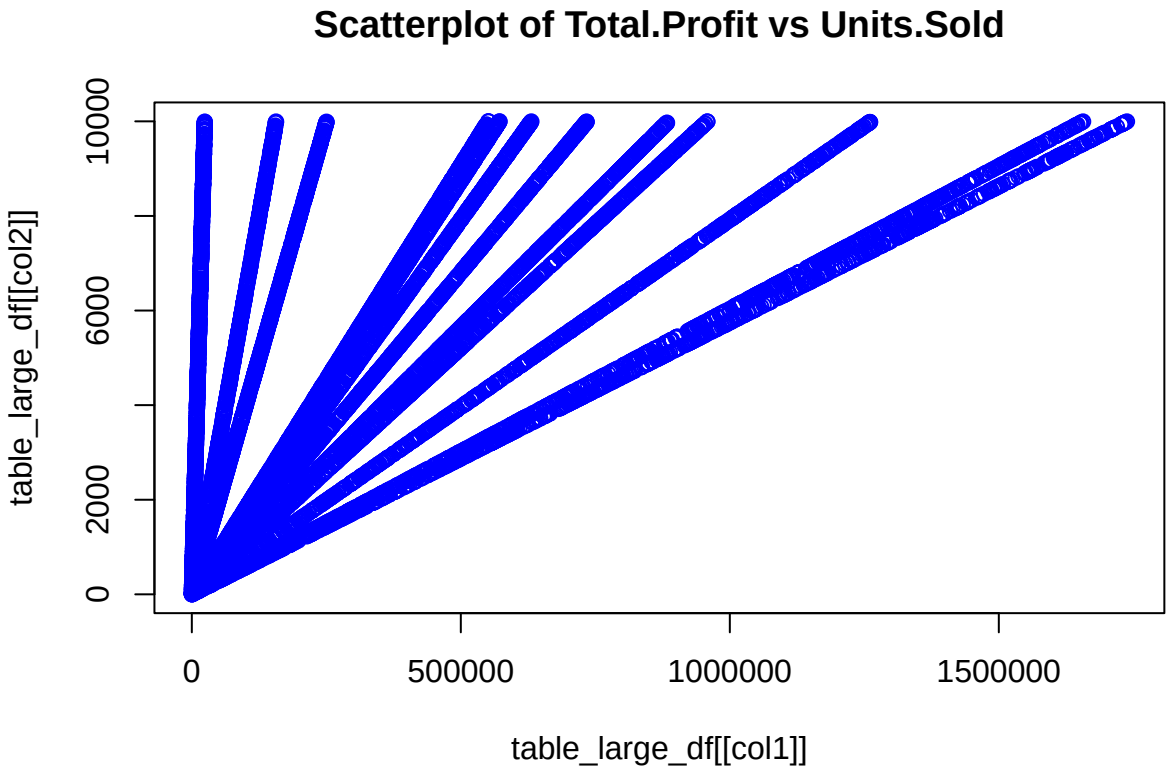

Scatterplot of Total.Cost vs Unit.Price
table_large_df[[col2]]
0
100
300
500
0e+00
1e+06
2e+06
3e+06
4e+06
5e+06
table_large_df[[col1]]

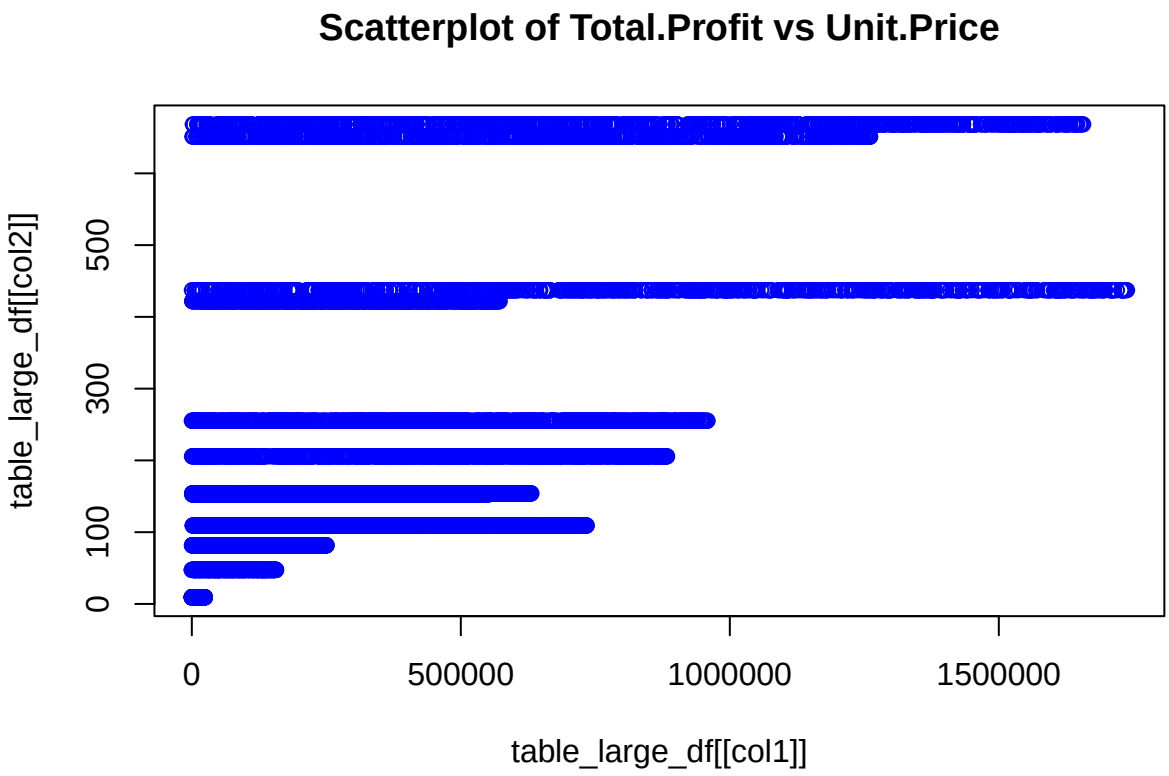

Scatterplot of Total.Cost vs Unit.Cost
table_large_df[[col2]]
0
100
200
300
400
500
0e+00
1e+06
2e+06
3e+06
4e+06
5e+06
table_large_df[[col1]]

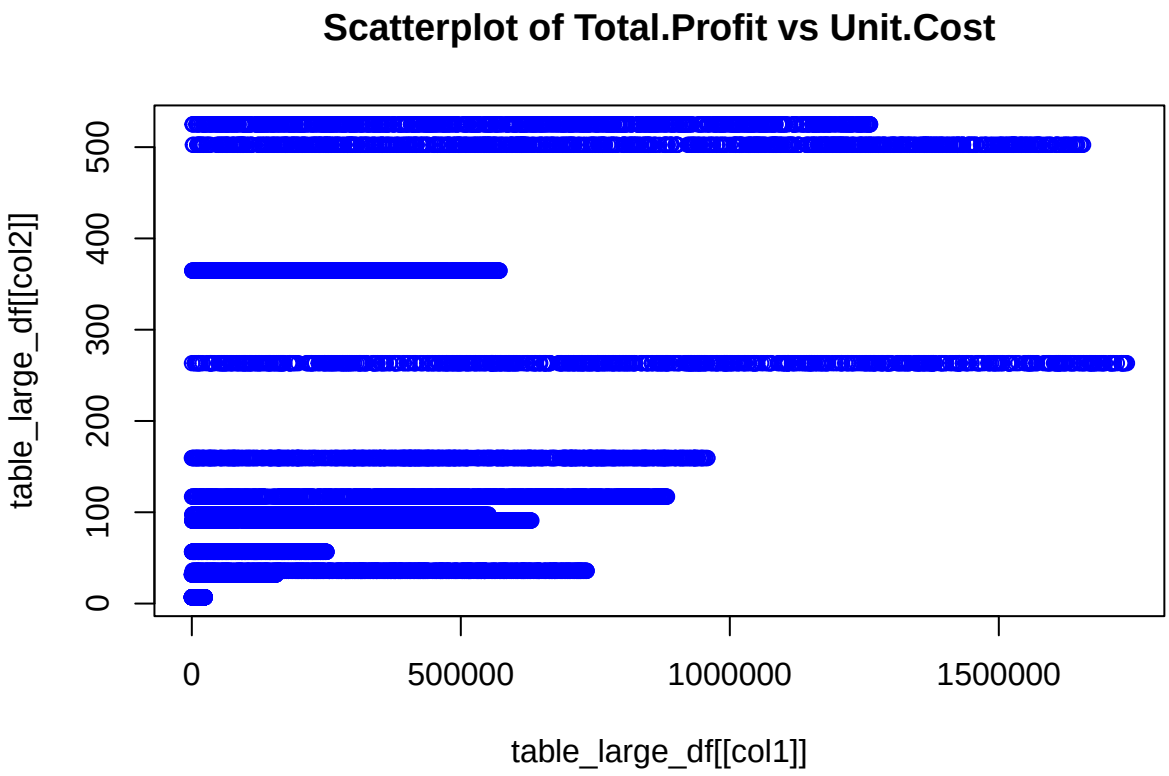

Scatterplot of Total.Cost vs Total.Revenue
table_large_df[[col2]]
0e+00
2e+06
4e+06
6e+06
0e+00
1e+06
2e+06
3e+06
4e+06
5e+06
table_large_df[[col1]]

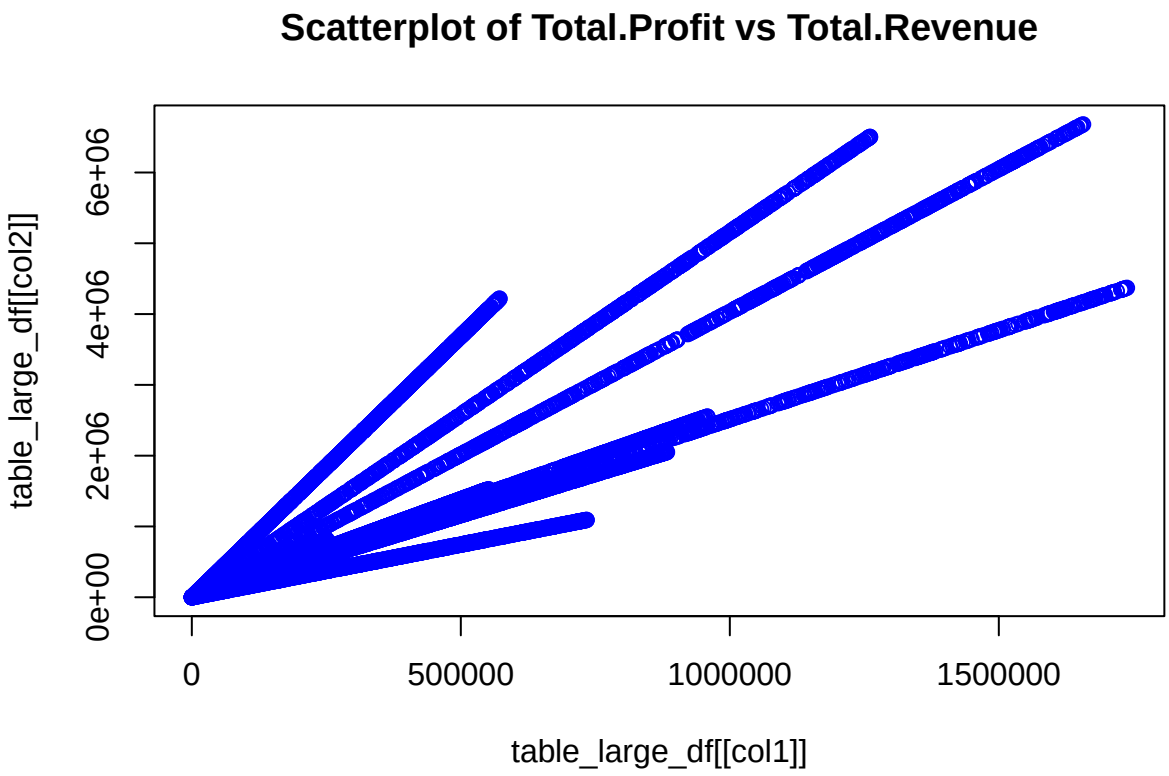

Scatterplot of Total.Cost vs Total.Profit
table_large_df[[col2]]
0
500000
1500000
0e+00
1e+06
2e+06
3e+06
4e+06
5e+06
table_large_df[[col1]]

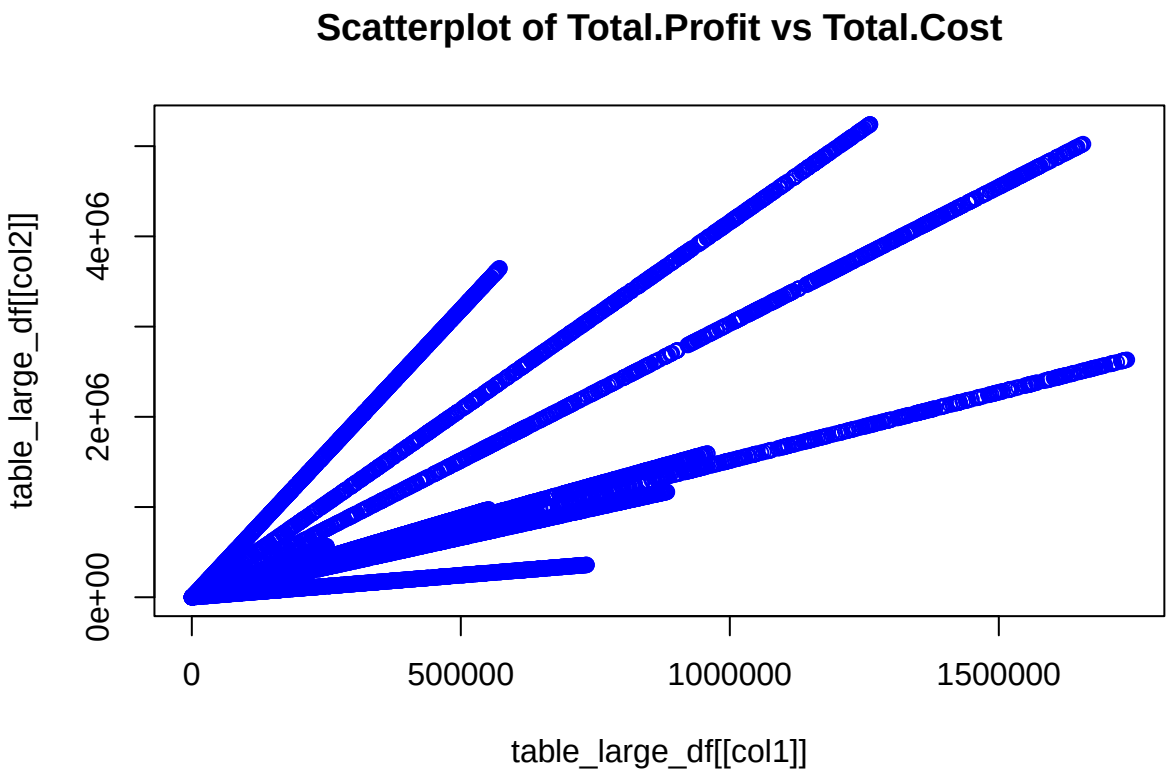

Scatterplot of Total.Profit vs Order.ID
table_large_df[[col2]]
1e+09
6e+08
2e+08
0
500000
1000000
1500000
table_large_df[[col1]]


Scatterplot of Total.Profit vs Units.Sold
table_large_df[[col2]]
0
2000
6000
10000
0
500000
1000000
1500000
table_large_df[[col1]]


Scatterplot of Total.Profit vs Unit.Price
table_large_df[[col2]]
0
100
300
500
0
500000
1000000
1500000
table_large_df[[col1]]


Scatterplot of Total.Profit vs Unit.Cost
table_large_df[[col2]]
0
100
200
300
400
500
0
500000
1000000
1500000
table_large_df[[col1]]


Scatterplot of Total.Profit vs Total.Revenue
table_large_df[[col2]]
0e+00
2e+06
4e+06
6e+06
0
500000
1000000
1500000
table_large_df[[col1]]



A scatter plot with scattered points suggests a lack of a strong linear relationship between variables, indicating their independence. The randomness in the distribution implies no structured or predictable connection. This phenomenon holds whether analyzing numeric or categorical variables. In statistical terms, this scattered appearance may signify homoscedasticity, indicating consistent variability across levels. While the lack of a clear pattern hints at independence or absence of correlation, further analysis is essential for a comprehensive understanding of the data dynamics.

A scatter plot showing points clustered at the top and bottom with a straight line in the middle suggests a potential linear relationship between the variables. This pattern indicates a positive correlation, implying that as one variable increases, the other tends to increase, and vice versa. Further statistical analysis, such as calculating correlation coefficients, is needed to quantify and confirm the strength of this relationship.

When scatter plots display a pattern with outliers concentrated at the top and the majority of data points clustered towards the bottom, it implies potential non-linear relationships or the presence of influential data points. These outliers can strongly affect correlation or regression analyses, necessitating careful investigation. Understanding the nature of these outliers is crucial, as they might indicate unique characteristics or anomalies in the dataset. Further analysis, such as examining residuals and exploring alternative modeling techniques, may be needed to accurately capture underlying patterns in the data.

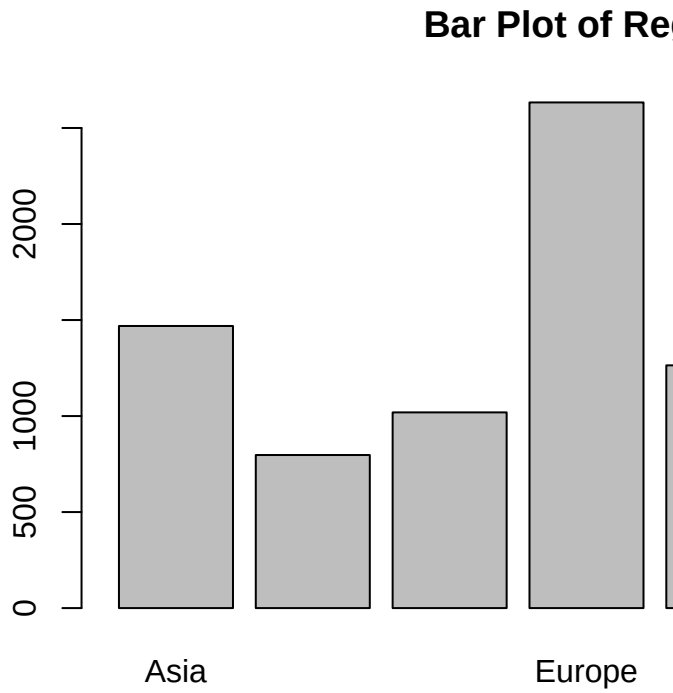
A diagonal line in a scatter plot from the bottom-left to the top-right indicates a positive linear relationship between the variables being plotted. This suggests that as one variable increases, the other also tends to increase. The steeper the slope, the stronger the positive correlation. Further analysis, such as calculating correlation coefficients and conducting regression analysis, can provide a more quantitative understanding of the relationship.

A scatter plot displaying a straight vertical line from bottom to top suggests that the two variables being compared have a perfect linear relationship. This means that as one variable increases, the other also increases proportionally. The correlation coefficient would be $+1$, indicating a strong positive correlation. However, it's important to note that this ideal scenario is less common in real-world data, and some variations

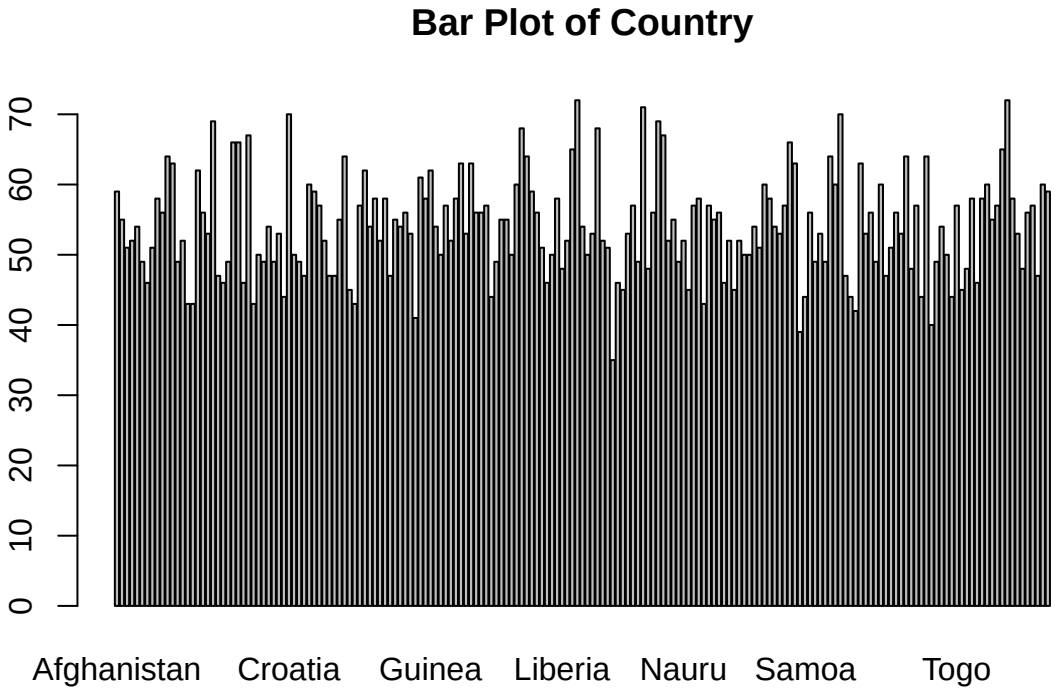
or deviations may be present due to other factors or measurement errors.

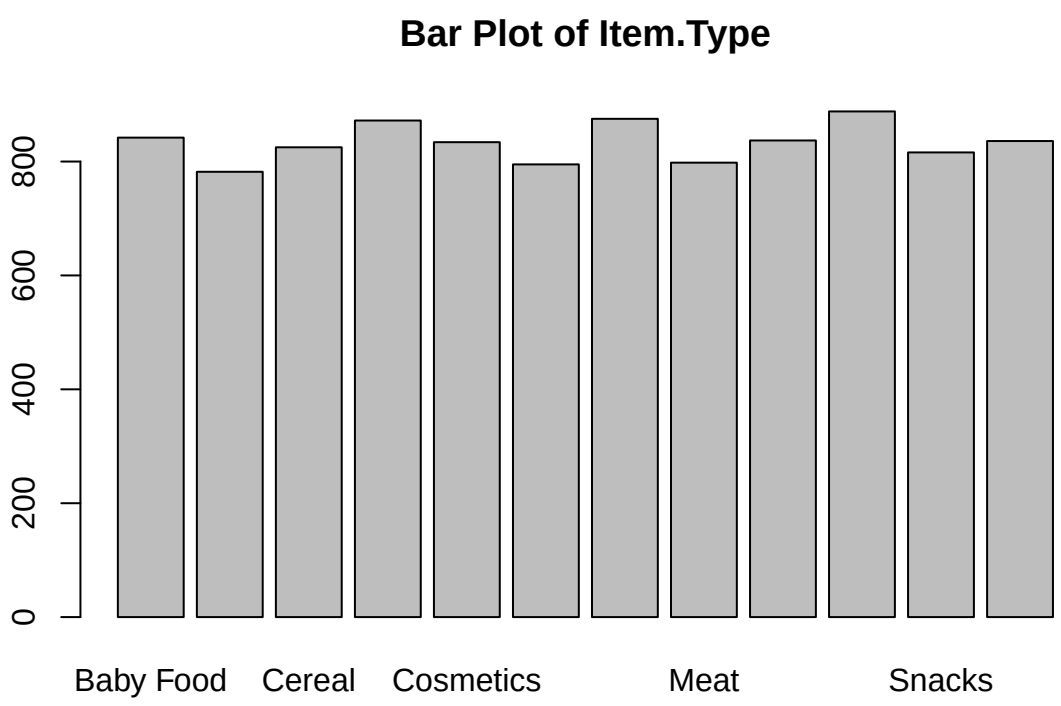
A scatter plot with points aligned horizontally indicates a perfect linear relationship where the two variables being compared have a constant value for one of them, regardless of the changes in the other variable. This implies a correlation coefficient of -1, representing a strong negative correlation. In simpler terms, as one variable increases, the other decreases proportionally. As with other ideal scenarios, variations and deviations may occur in real-world data due to external factors or measurement errors.

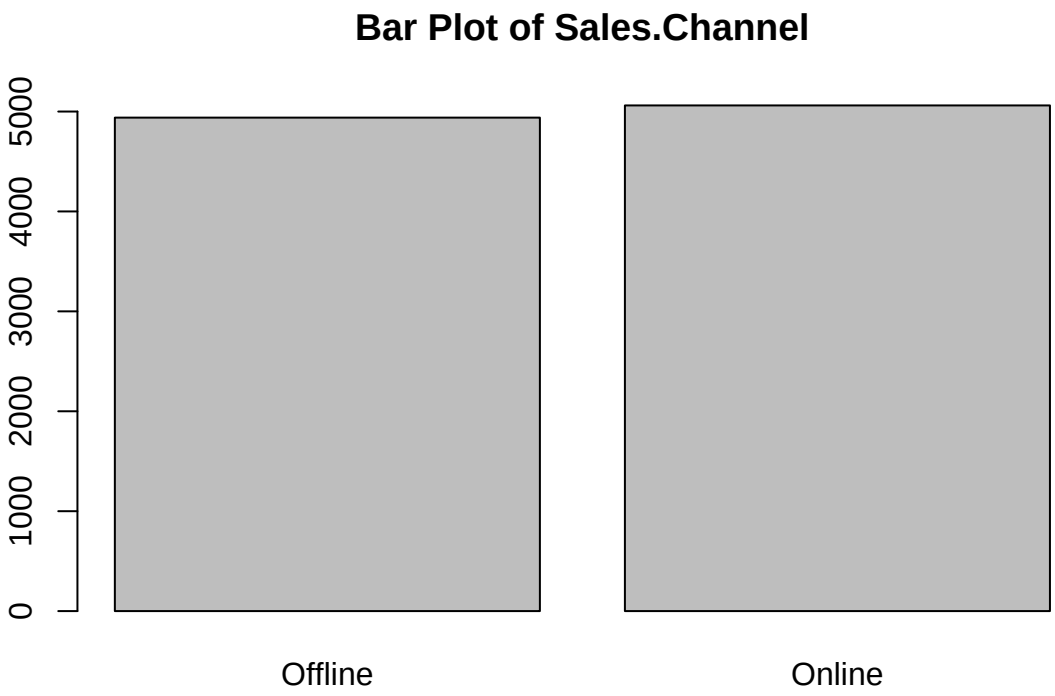
```
for (column_name in names(table_large_df)) {
  if (!is.numeric(table_large_df[[column_name]])) {
    barplot(table(table_large_df[[column_name]]), main = paste("Bar Plot of", column_name))
  }
}
```



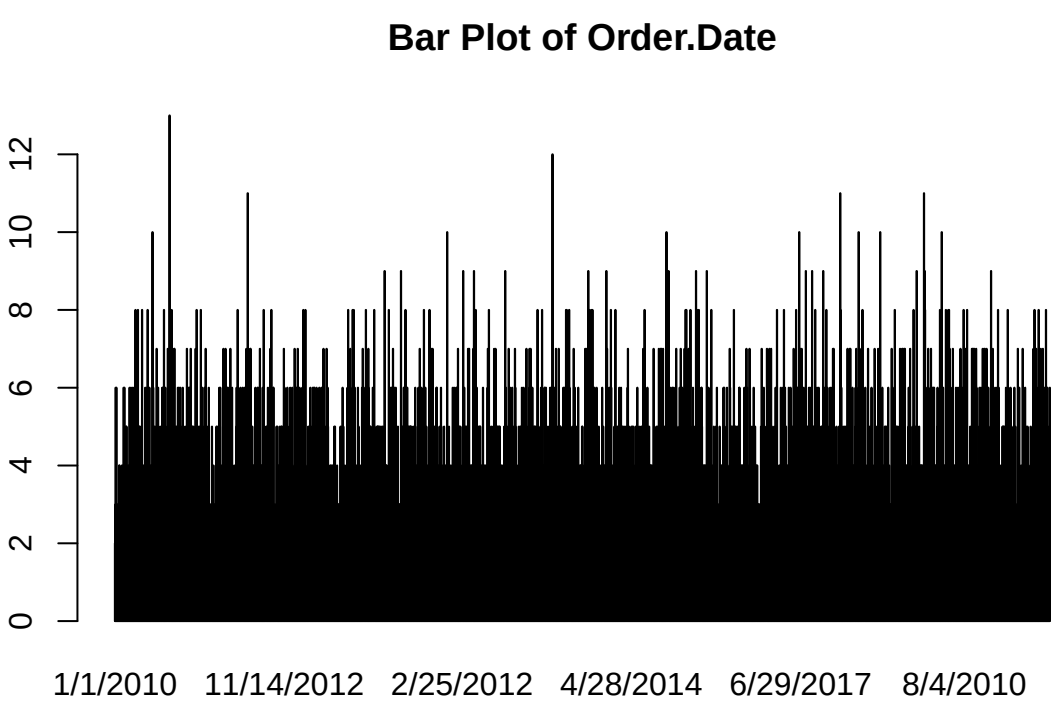
Frequency table for each variables in each columns:

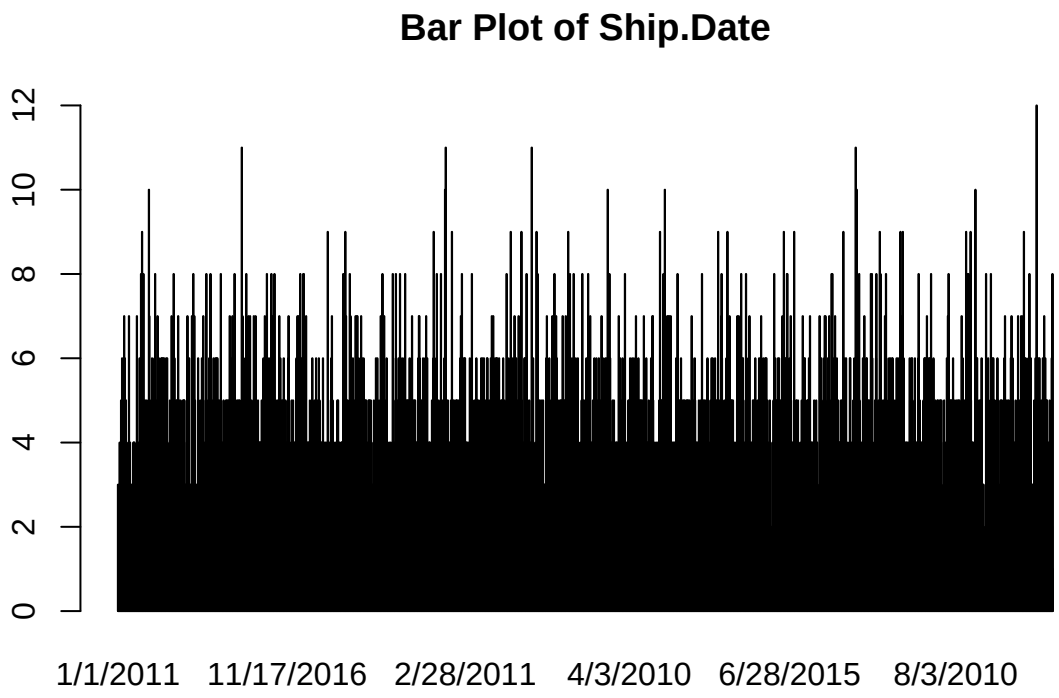



Bar Plot of Item.Type
0
200
400
600
800
Baby Food
Cereal
Cosmetics
Meat
Snacks


Bar Plot of Sales.Channel
0
1000
2000
3000
4000
5000
Offline
Online


Bar Plot of Order.Priority
0
500
1000
1500
2000
2500
C
H
L
M

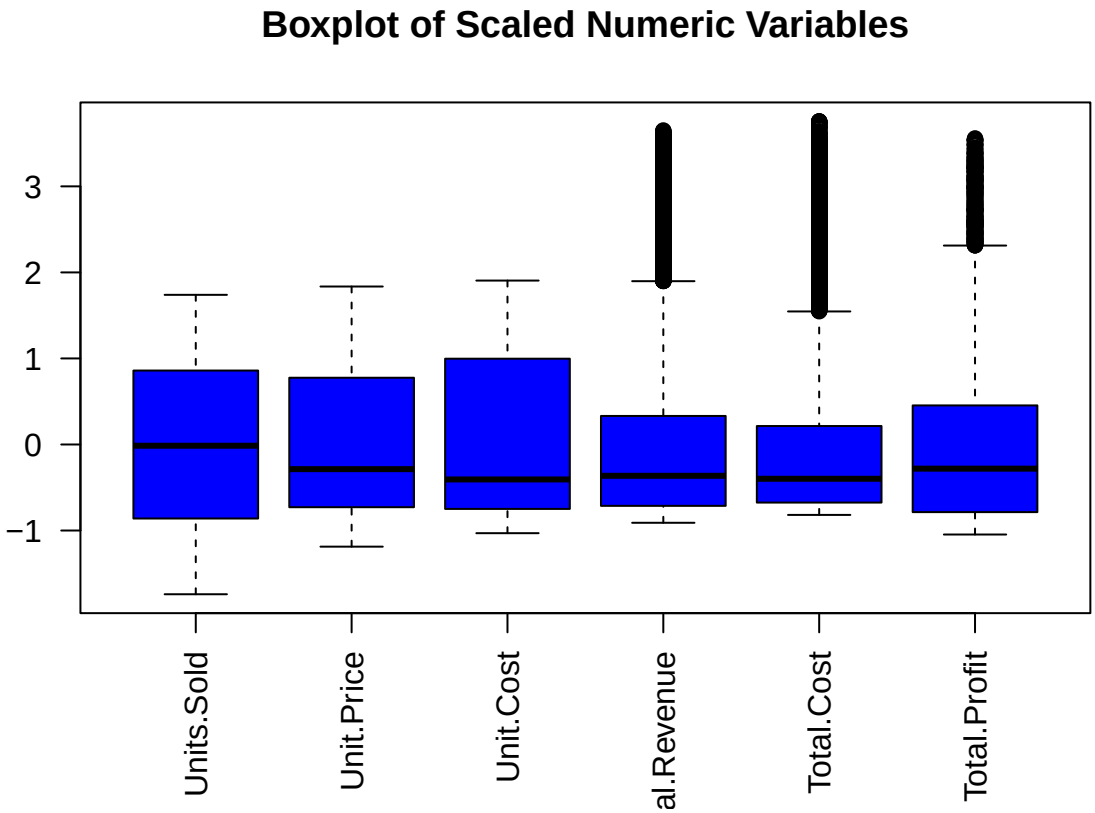

Bar Plot of Order.Date
0
2
4
6
8
10
12
1/1/2010
11/14/2012
2/25/2012
4/28/2014
6/29/2017
8/4/2010



Boxplots for all numeric columns. From the above summary of your dataset, it appears that the numeric variables (Units.Sold, Unit.Price, Unit.Cost, Total.Revenue, Total.Cost, Total.Profit) have a wide range of values with varying scales. The Units.Sold variable, for instance, has a minimum value of 2 and a maximum of 10,000, while the Total.Revenue variable ranges from 168 to 6,680,027.

Given the diverse scales of these variables, when I create a boxplot for all of them at once, some may dominate the plot, making it challenging to see the details of others. I considered normalizing or scaling these variables to bring them to a similar scale for better visualization.

```
# Assuming 'df' is your dataframe
scaled_df <- table_large_df[, c("Units.Sold", "Unit.Price", "Unit.Cost", "Total.Revenue", "Total.Cost",

# Scale the numeric variables
scaled_df <- scale(scaled_df)

# Create a boxplot
boxplot(scaled_df, col = "blue", main = "Boxplot of Scaled Numeric Variables", las = 2)
```



The boxplot analysis reveals a positive distribution, indicating a concentration of data towards higher values. The right-skewed pattern suggests a majority of observations falling on the higher end of the axis. Additionally, the presence of too much outliers in Total Revenue, Total Cost, and Total Profit highlights extreme values that significantly deviate from the general trend, adding an extreme complexity to the dataset.

Machine Learning Algorithms

Large Dataset

Decision Tree

```
set.seed(123)
rpart.control(maxdepth = 30)
```

```
## $minsplit
## [1] 20
##
## $minbucket
## [1] 7
##
## $cp
## [1] 0.01
##
```

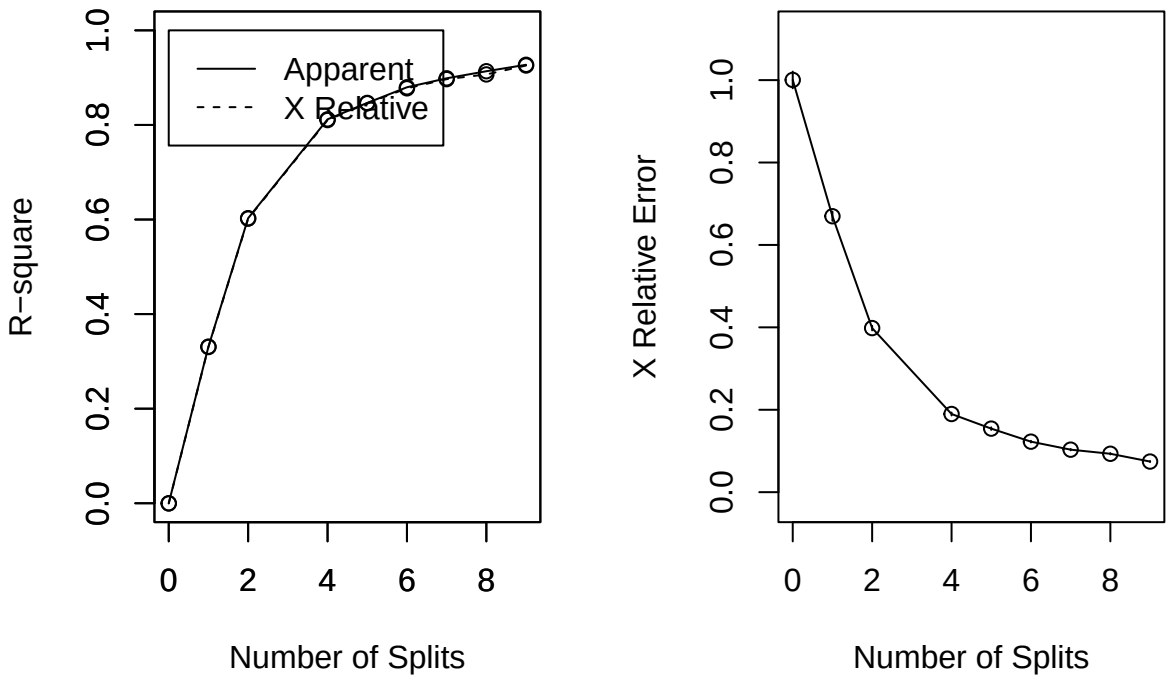
```
## $maxcompete
## [1] 4
##
## $maxsurrogate
## [1] 5
##
## $usesurrogate
## [1] 2
##
## $surrogatestyle
## [1] 0
##
## $maxdepth
## [1] 30
##
## $xval
## [1] 10
```

```r
set.seed(42)
# Assuming your large dataset is named "table_large_df"
train_indices <- sample(1:nrow(table_large_df), 0.8 * nrow(table_large_df))
train_data <- table_large_df[train_indices, ]
test_data <- table_large_df[-train_indices, ]

# Remove unnecessary variables
train_data <- train_data[, !(names(train_data) %in% c("Order.ID", "Order.Date", "Ship.Date"))]
test_data <- test_data[, !(names(test_data) %in% c("Order.ID", "Order.Date", "Ship.Date"))]

fit <- rpart( Total.Profit ~ Item.Type + Region + Sales.Channel + Order.Priority + Country + Units.Sold

# detailed summary of splits
#summary(fit)
```

```r
# create additional plots
par(mfrow=c(1,2)) # two plots on one page
rsq.rpart(fit) # visualize cross-validation results
```

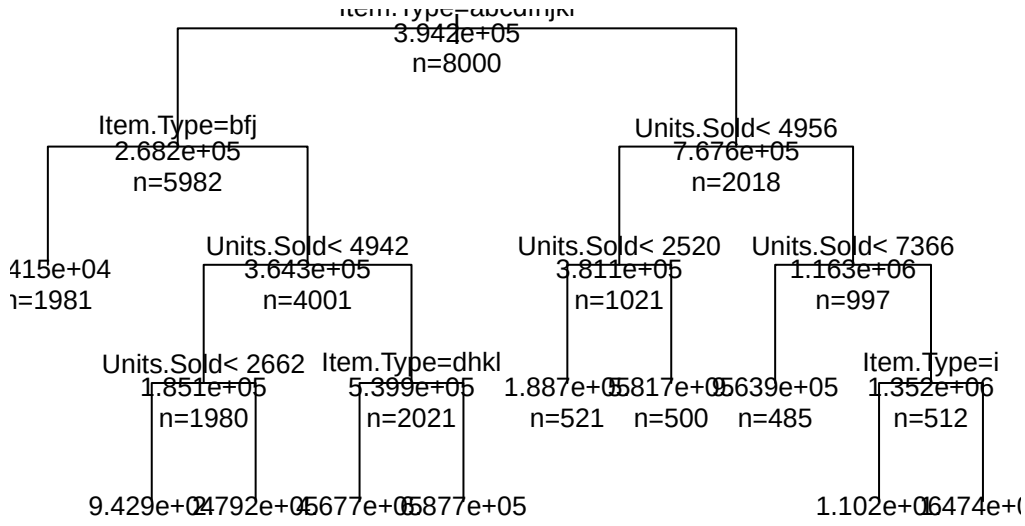
```
##
## Regression tree:
## rpart(formula = Total.Profit ~ Item.Type + Region + Sales.Channel +
##     Order.Priority + Country + Units.Sold, data = train_data,
##     method = "anova")
##
## Variables actually used in tree construction:
## [1] Item.Type  Units.Sold
##
## Root node error: 1.1362e+15/8000 = 1.4202e+11
##
## n= 8000
##
##         CP nsplit rel error   xerror      xstd
## 1 0.331177      0  1.000000 1.000433 0.0202950
## 2 0.271669      1  0.668823 0.669623 0.0112266
```

```
## 3 0.104528      2  0.397154 0.398184 0.0055617
## 4 0.034684      4  0.188099 0.189715 0.0035696
## 5 0.033076      5  0.153415 0.154213 0.0031046
## 6 0.018992      6  0.120339 0.122592 0.0020515
## 7 0.014891      7  0.101347 0.103165 0.0016952
## 8 0.013673      8  0.086456 0.093172 0.0016603
## 9 0.010000      9  0.072783 0.074309 0.0011999
```

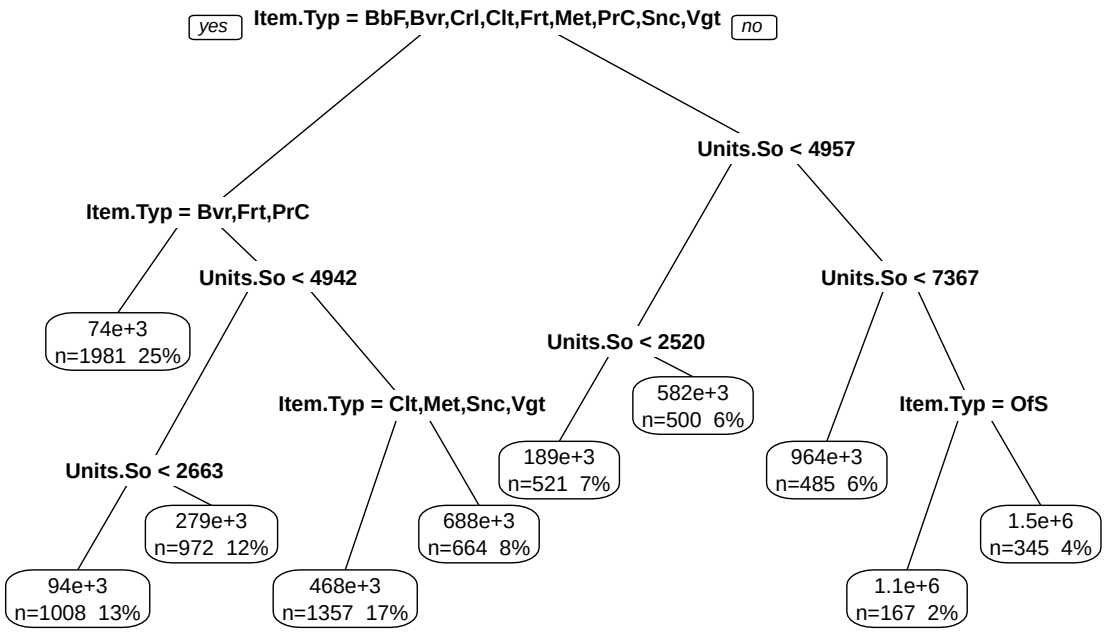


The regression tree utilized two variables, Item.Type and Units.Sold, to construct the tree. These variables were deemed the most relevant for predicting Total.Profit based on the tree-building algorithm's criteria. The root node error is a measure of the initial error before any splits are made. In this case, the sum of squared errors (SSE) divided by the number of observations (8000) results in a root node error of approximately 1.42e+11. CP (Complexity Parameter): A measure of tree complexity, where smaller values indicate simpler trees. nsplit: Number of splits. rel error: Relative error reduction at each split. xerror: Cross-validated error rate, an estimate of the model's prediction error on unseen data. xstd: Standard deviation of the cross-validated error. These values show how the cross-validated error changes as the tree grows. The goal is often to identify a level of complexity (number of splits) that minimizes the cross-validated error, ensuring good generalization to new data. In summary, the regression tree was built using Item.Type and Units.Sold as key predictors. The cross-validated error rates provide insights into the model's performance at different levels of complexity, helping to balance simplicity and predictive accuracy.

```r
# plot tree
plot(fit, uniform=TRUE,
   main="Regression Tree for Total profit ")
text(fit, use.n=TRUE, all=TRUE, cex=.8)
```




```r
# Create an attractive postscript plot of the tree
prp(fit, main = "Regression Tree for Total profit", extra = 101)
```




```
# Make predictions on the test set
predictions <- predict(fit, test_data)

# Extract the actual Total.Profit values from the test set
actual_values <- test_data$Total.Profit

# Calculate Mean Squared Error
mse <- mean((predictions - actual_values)^2)

# Print the MSE
cat("Mean Squared Error (MSE):", mse, "\n")
```

```
## Mean Squared Error (MSE): 10369206128
```

The Mean Squared Error (MSE) value for a decision tree regression model represents the average squared difference between the actual and predicted values. A MSE of 10369206128 indicates the average squared error across all predictions made by the decision tree. The MSE of 10369206128 suggests that, on average, the squared difference between the predicted and actual values is relatively high.

KNN Model

```
set.seed(42)
```

```r
# Split the dataset into training and testing sets
train_indices <- sample(1:nrow(table_large_df), 0.8 * nrow(table_large_df))
train_data <- table_large_df[train_indices, ]
test_data <- table_large_df[-train_indices, ]

# Select relevant predictor variables (assuming all numeric)
predictors <- train_data[, c("Units.Sold", "Unit.Price", "Unit.Cost", "Total.Revenue", "Total.Cost")]

# Standardize the predictors (optional but often recommended)
scaled_predictors <- scale(predictors)

# Select the target variable
target <- train_data$Total.Profit

# Build the KNN model
knn_model <- knn(train = scaled_predictors, test = scaled_predictors, cl = target, k = 2)

# Evaluate the model on the test set
test_predictions <- knn(train = scaled_predictors, test = scaled_predictors, cl = target, k = 2)

# Assess accuracy or other metrics
accuracy <- sum(test_predictions == target) / length(target)

# Print or visualize the results as needed
print(accuracy)
```

```
## [1] 0.5125
```

The output [1] 0.5125 likely represents a result or metric from a k-Nearest Neighbors (kNN) model. Depending on the specific context, it could be a predicted value or a performance metric such as accuracy or mean squared error. The exact interpretation depends on the type of problem the model is solving (e.g., regression or classification) and the evaluation metric used.

The analysis of the two models, Decision Tree and KNN, on the given datasets reveals differences in their performance metrics. The Decision Tree model, evaluated using Mean Squared Error (MSE), yielded a value of 10,369,206,128, indicating the average squared difference between predicted and actual values. Meanwhile, the KNN model was assessed with a metric of [1] 0.5125, likely representing classification accuracy or error for discrete outcomes. It's crucial to note that these metrics are not directly comparable, as MSE is tailored for regression tasks, while the KNN metric is more relevant to classification tasks. Further exploration of the models' strengths and weaknesses is necessary to determine their suitability for the specific dataset and analysis goals.

In conclusion, this exploratory analysis offers a nuanced perspective on our business metrics. Beyond the numerical values, it provides a narrative of our business journey, highlighting areas of strength, potential concerns, and opportunities for refinement. Armed with these insights, we are better equipped to make informed decisions that will steer our business towards sustained growth and success.

Reference: https://excelbianalytics.com/wp/downloads-18-sample-csv-files-data-sets-for-testing-sales/