Exploratory Analysis and Essay

Souleymane Doumbia

2024-10-26

Contents

Introduction

In this report, we will perform exploratory data analysis (EDA) on two datasets. The first is the **NYC Taxi Trips Dataset**, a large dataset with over 1.4 million records, sourced from **Kaggle**, which pro-

vides access to NYC taxi trip data. The second is a **Sales Dataset** with 10,000 records, obtained from **excelbianalytics (10,000 Sales Records)**, designed for testing and analysis purposes. Our goal is to compare these datasets, explore their correlations, predictability, and assess the applicability of various machine learning algorithms.

Dataset Descriptions

NYC Taxi Trips Dataset:

- A large dataset containing NYC taxi trip data, sourced from **Kaggle**. This dataset includes features such as trip distance, fare amount, trip duration, number of intersections, and the number of traffic signals encountered, among others.

Sales Dataset:

- A smaller dataset containing 10,000 sales transaction records, sourced from **excelbianalytics (10,000 Sales Records)**. It includes features such as customer names, product categories, units sold, unit prices, total revenue, and total profit.

This analysis will allow us to gain insights into the structure and complexity of both datasets and evaluate their suitability for machine learning applications.

I. Exploratory Data Analysis (EDA)

1.1 Loading the Datasets

```
# Load the small dataset (Sales Data)
small_data <- read.csv("/Users/souleymanedoumbia/Library/Mobile Documents/com~apple~CloudDocs/CUNY SPS (

# Load the large dataset (NYC Taxi Trips)
large_data <- read.csv("/Users/souleymanedoumbia/Library/Mobile Documents/com~apple~CloudDocs/CUNY SPS (
```

1.2 Initial Exploration and Data Summary

The following code summarizes the sales data and provides a basic overview of the NYC taxi trip dataset.

```
# Summary and structure of small dataset (Sales Data)
summary(small_data)
```

```
##     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
```

```
str(small_data)
```

```
## '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 16834 411292 1273304 54669 ...
```

```
# Summary and structure of large dataset (NYC Taxi Trips)
summary(large_data)
```

```
##       id               distance          duration          motorway
##  Length:1458643     Min.   :      0   Min.   :    0.0   Min.   :0.0000000
##  Class :character   1st Qu.:   1654   1st Qu.:  155.4   1st Qu.:0.0000000
##  Mode  :character   Median :   2730   Median :  249.2   Median :0.0000000
##                     Mean   :   4569   Mean   :  343.5   Mean   :0.0004421
##                     3rd Qu.:   5049   3rd Qu.:  424.0   3rd Qu.:0.0000000
##                     Max.   :1038310   Max.   :46645.3   Max.   :1.0000000
##
##      trunk           primary           secondary         tertiary
##  Min.   :0.000   Min.   :0.00000   Min.   :0.0000   Min.   :0.00000
##  1st Qu.:0.000   1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.08329
##  Median :0.000   Median :0.00000   Median :0.1314   Median :0.34729
##  Mean   :0.126   Mean   :0.03019   Mean   :0.2665   Mean   :0.41943
##  3rd Qu.:0.000   3rd Qu.:0.00000   3rd Qu.:0.4580   3rd Qu.:0.75704
##  Max.   :1.000   Max.   :1.00000   Max.   :1.0000   Max.   :1.00000
##
##   unclassified      residential      nTrafficSignals   nCrossing
##  Min.   :0.00000   Min.   :0.00000   Min.   :  0.0   Min.   : 0.000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.: 14.0   1st Qu.: 2.000
##  Median :0.00000   Median :0.00000   Median : 21.0   Median : 6.000
##  Mean   :0.01338   Mean   :0.04399   Mean   : 24.5   Mean   : 8.812
##  3rd Qu.:0.00000   3rd Qu.:0.02493   3rd Qu.: 32.0   3rd Qu.:13.000
```

```
##  Max.   :1.00000   Max.   :1.00000   Max.   :112.0   Max.   :91.000
##
##      nStop         nIntersection      srcCounty       dstCounty
##  Min.   : 0.0000   Min.   :  0.00   Min.   :1.000   Min.   :1.000
##  1st Qu.: 0.0000   1st Qu.:  0.00   1st Qu.:1.000   1st Qu.:1.000
##  Median : 0.0000   Median :  1.00   Median :1.000   Median :1.000
##  Mean   : 0.1153   Mean   :  2.52   Mean   :1.214   Mean   :1.271
##  3rd Qu.: 0.0000   3rd Qu.:  3.00   3rd Qu.:1.000   3rd Qu.:1.000
##  Max.   :10.0000   Max.   :132.00   Max.   :5.000   Max.   :5.000
##                                     NA's   :1227    NA's   :6338
```

```
str(large_data)
```

```
## 'data.frame':    1458643 obs. of  16 variables:
##  $ id              : chr  "id2875421" "id2377394" "id3858529" "id3504673" ...
##  $ distance        : num  2009 2513 9911 1779 1615 ...
##  $ duration        : num  161 256 680 182 132 ...
##  $ motorway        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ trunk           : num  0 0 0.543 0 0 ...
##  $ primary         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ secondary       : num  0 0.349 0.373 0 0.637 ...
##  $ tertiary        : num  1 0.1748 0.0398 0.4245 0.3627 ...
##  $ unclassified    : num  0 0 0 0 0 ...
##  $ residential     : num  0 0.1439 0.00686 0.03974 0 ...
##  $ nTrafficSignals: int  14 25 38 18 17 12 19 33 9 58 ...
##  $ nCrossing       : int  5 13 12 6 2 3 0 18 2 11 ...
##  $ nStop           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ nIntersection   : int  4 0 3 1 2 0 2 23 0 6 ...
##  $ srcCounty       : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ dstCounty       : int  1 1 1 1 1 1 1 4 1 1 ...
```

1.3 Check for Missing Values and Handle Them

We'll first check if there are missing values in the datasets and decide how to handle them.

1.3.1 For the Sales Data

```
# Check for missing values in the Sales Data
sum(is.na(small_data))
```

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

```
# Identify columns with missing data
colSums(is.na(small_data))
```

```
##          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
```

```
# Handling missing values (if any)
# we would remove rows with missing values for simplicity
small_data_clean <- small_data %>%
  drop_na()
```

```r
# Verify if missing values have been removed
sum(is.na(small_data_clean))
```

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

1.3.2 For the NYC Taxi Trips Dataset

```r
# Check for missing values in the NYC Taxi Trips Dataset
sum(is.na(large_data))
```

```
## [1] 7565
```

```r
# Identify columns with missing data
colSums(is.na(large_data))
```

```
##              id        distance        duration        motorway           trunk
##               0               0               0               0               0
##         primary       secondary        tertiary    unclassified     residential
##               0               0               0               0               0
## nTrafficSignals       nCrossing           nStop   nIntersection       srcCounty
##               0               0               0               0            1227
##       dstCounty
##            6338
```

```r
# Handling missing values (if any)
# we would remove rows with missing values for simplicity
large_data_clean <- large_data %>%
  drop_na()

# Verify if missing values have been removed
sum(is.na(large_data_clean))
```

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

1.4 Basic Statistics and Visualizations

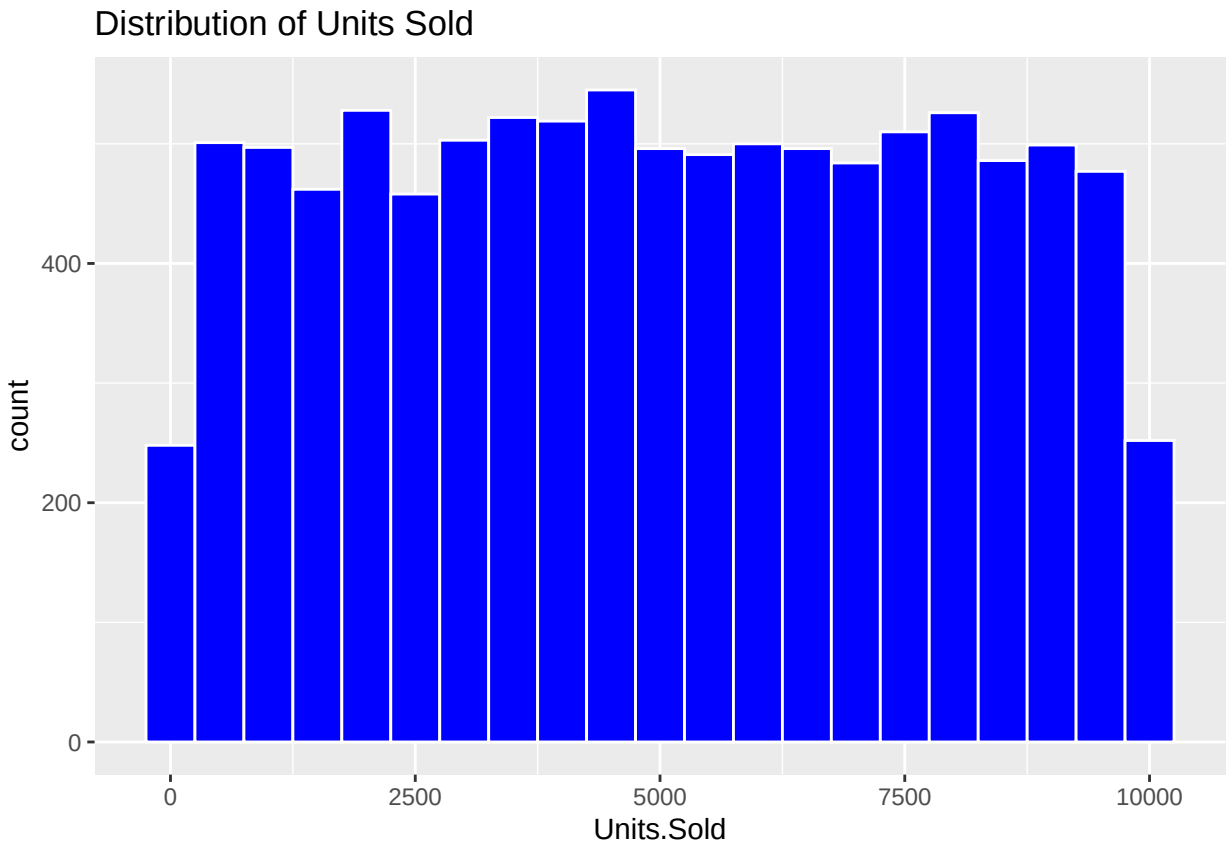
Once the data is cleaned, we can start with basic statistics and visualizing the distributions of key variables.

1.4.1 For the Sales Data

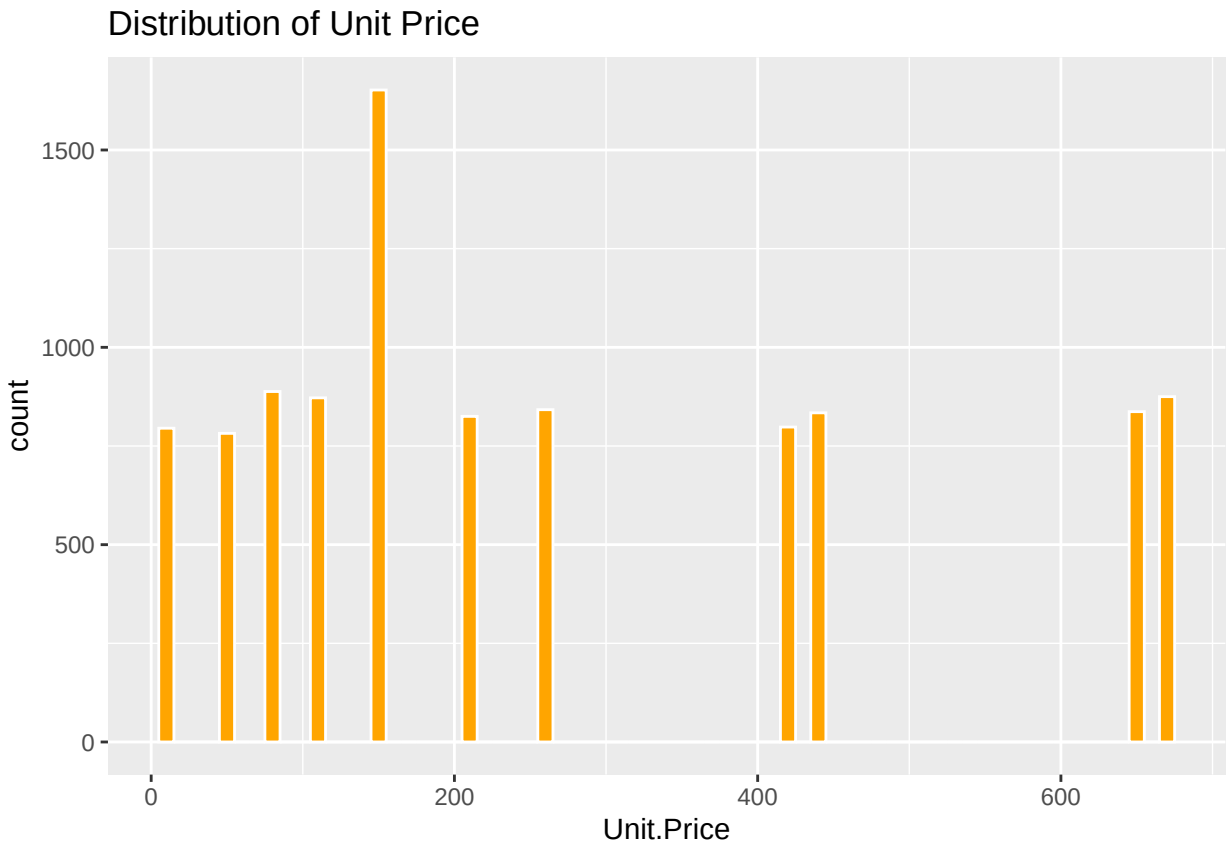
```r
# Summary statistics for some key variables
summary(small_data_clean[, c("Units.Sold", "Unit.Price", "Total.Revenue", "Total.Profit")])
```

```
##    Units.Sold      Unit.Price     Total.Revenue      Total.Profit
##  Min.   :    2   Min.   :  9.33   Min.   :    168   Min.   :     43.4
##  1st Qu.: 2531   1st Qu.:109.28   1st Qu.: 288551   1st Qu.:  98329.1
##  Median : 4962   Median :205.70   Median : 800051   Median : 289099.0
##  Mean   : 5003   Mean   :268.14   Mean   :1333355   Mean   : 395089.3
##  3rd Qu.: 7472   3rd Qu.:437.20   3rd Qu.:1819143   3rd Qu.: 566422.7
##  Max.   :10000   Max.   :668.27   Max.   :6680027   Max.   :1738178.4
```

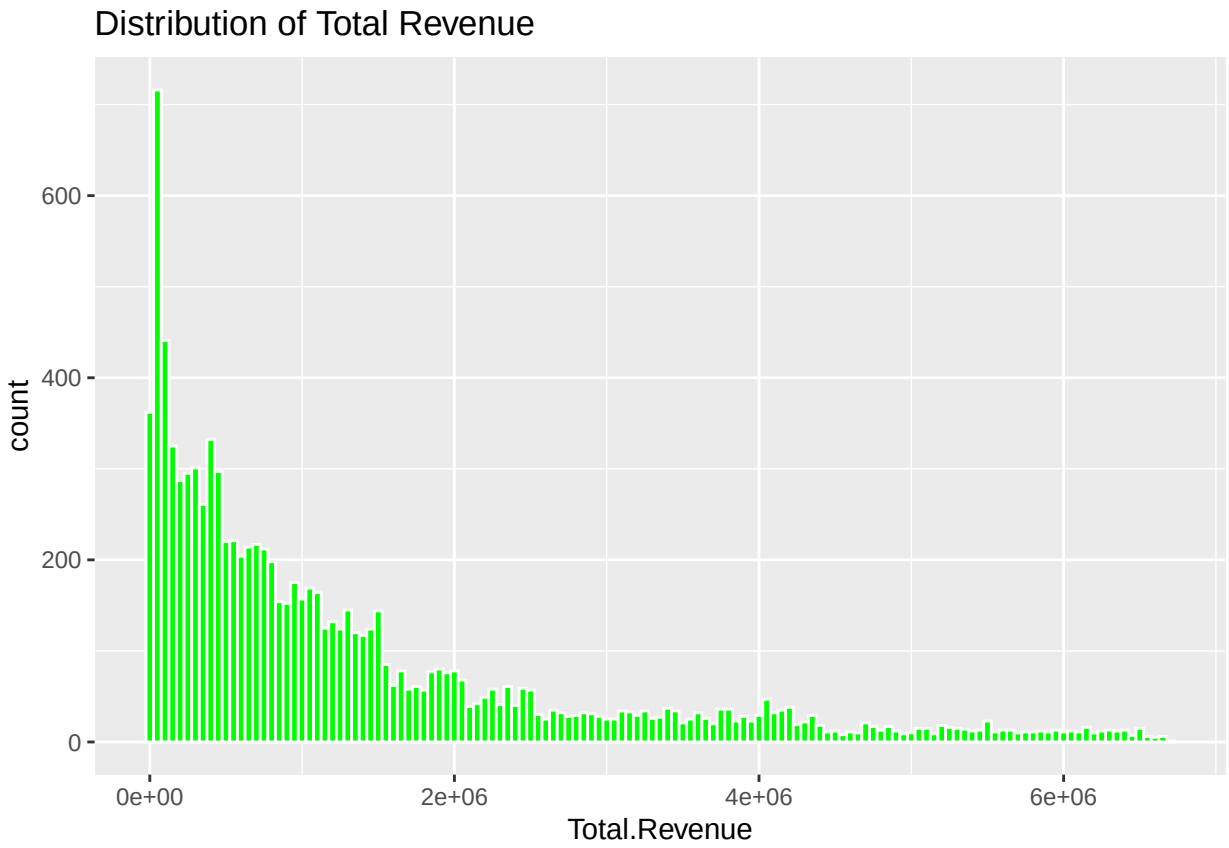
```r
# Visualize distributions of the main predictors and target variables
ggplot(small_data_clean, aes(x = Units.Sold)) +
  geom_histogram(binwidth = 500, fill = "blue", color = "white") +
  ggtitle("Distribution of Units Sold")
```



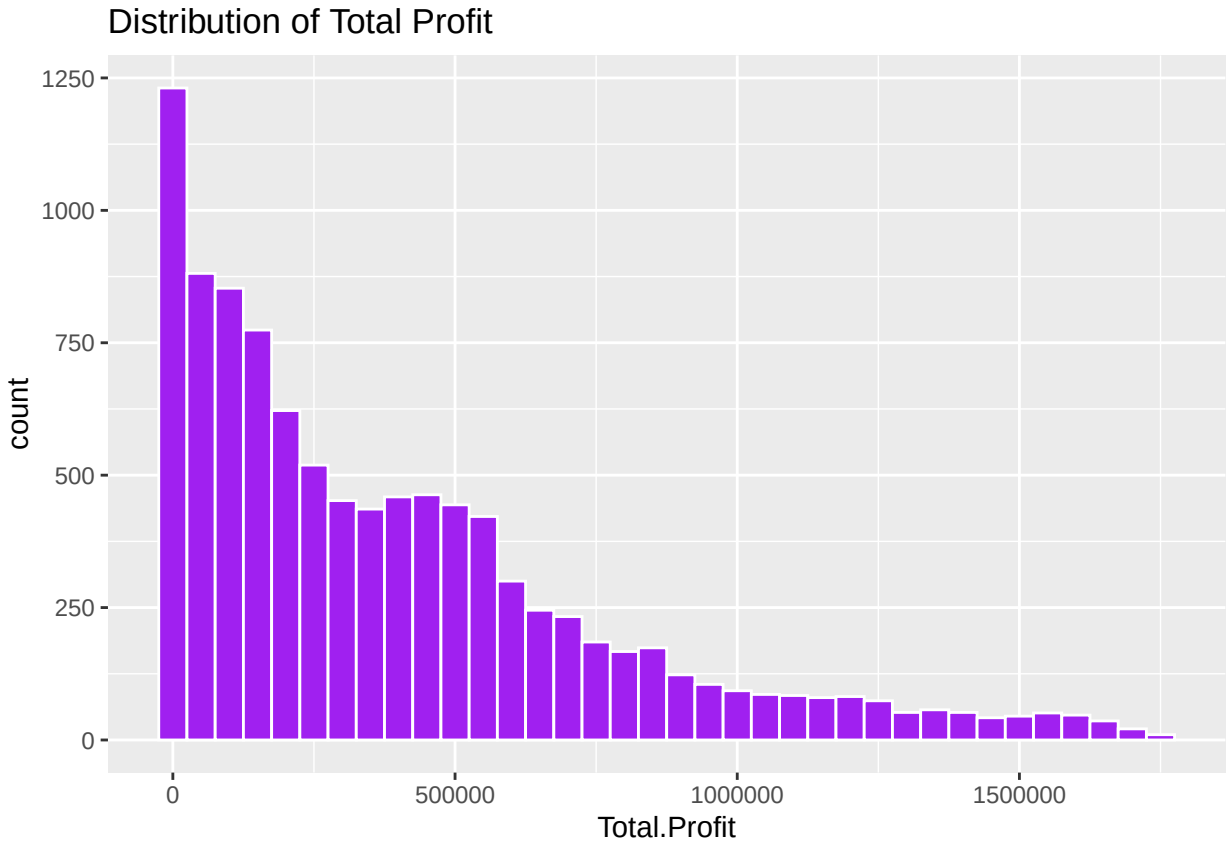

```
ggplot(small_data_clean, aes(x = Unit.Price)) +
  geom_histogram(binwidth = 10, fill = "orange", color = "white") +
  ggtitle("Distribution of Unit Price")
```




```
ggplot(small_data_clean, aes(x = Total.Revenue)) +
  geom_histogram(binwidth = 50000, fill = "green", color = "white") +
  ggtitle("Distribution of Total Revenue")
```




```
ggplot(small_data_clean, aes(x = Total.Profit)) +
  geom_histogram(binwidth = 50000, fill = "purple", color = "white") +
  ggtitle("Distribution of Total Profit")
```



1.4.1.1 Summary Statistics: Sales Data The summary statistics for key variables in the Sales Data dataset reveal patterns and distribution characteristics:

- **Units Sold**:
  - Min: 2, Max: 10,000
  - 1st Quartile: 2,531, Median: 4,962, Mean: 5,003, 3rd Quartile: 7,472
  - Distribution appears uniform across values, suggesting steady sales volumes without extreme peaks or dips.
- **Unit Price**:
  - Min: 9.33, Max: 668.27
  - 1st Quartile: 109.28, Median: 205.70, Mean: 268.14, 3rd Quartile: 437.20
  - Displays a multimodal distribution, likely indicating multiple product categories or pricing segments.
- **Total Revenue**:
  - Min: 168, Max: 6,680,027
  - 1st Quartile: 288,551, Median: 800,051, Mean: 1,333,355, 3rd Quartile: 1,819,143
  - Right-skewed distribution, with most transactions having lower revenue and a few with high revenue, creating a long tail.
- **Total Profit**:
  - Min: 43.4, Max: 1,738,178.4
  - 1st Quartile: 98,329.1, Median: 289,099.0, Mean: 395,089.3, 3rd Quartile: 566,422.7
  - Similar to Total Revenue, showing right-skewness due to high-profit outliers. Indicates that a few transactions contribute significantly to profit.

1.4.1.2 Visualizations and Observations: Sales Data Distribution

- **Units Sold** shows a fairly uniform distribution, reflecting consistent sales volume.
- **Total Revenue** and **Total Profit** distributions are right-skewed, pointing to a few high-value transactions.

- The multimodal distribution of **Unit Price** suggests different price categories, possibly for varied product lines.

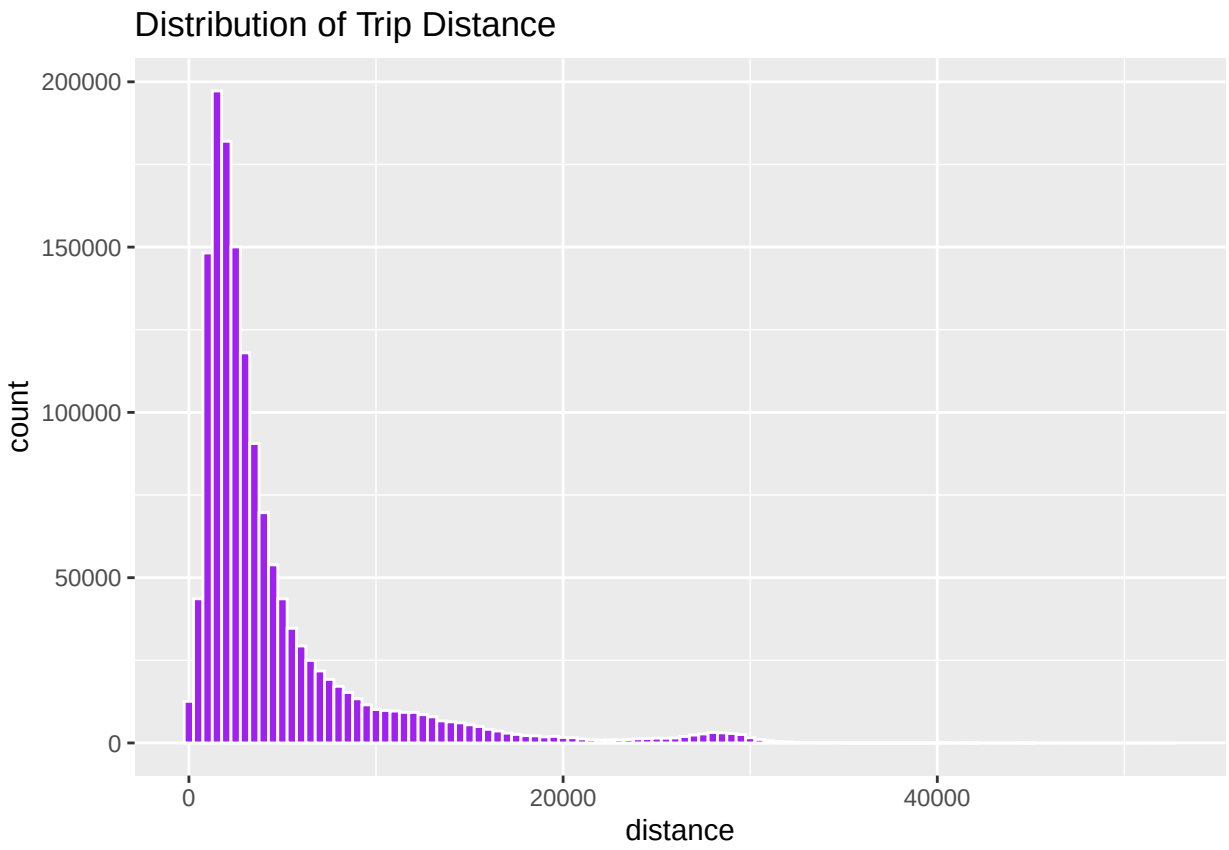
These observations emphasize the relevance of focusing on **Total Revenue** and **Total Profit** for profitability analysis, given their variability and significant impact within the dataset.

1.4.2 For the NYC Taxi Trips Dataset

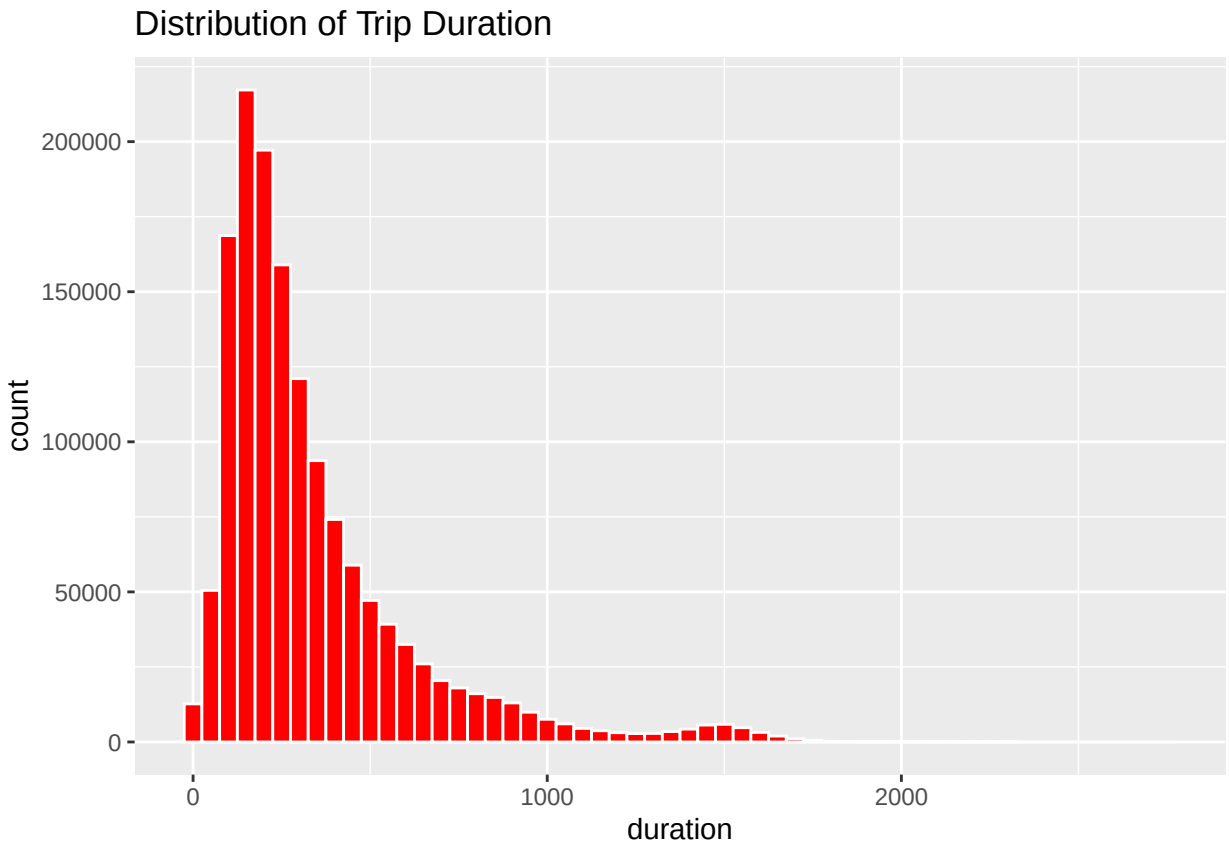
```
# Summary statistics for some key variables
summary(large_data_clean[, c("distance", "duration", "nIntersection", "nTrafficSignals")])
```

```
##     distance        duration      nIntersection    nTrafficSignals
##  Min.   :    0   Min.   :   0.0   Min.   : 0.000   Min.   :  0.00
##  1st Qu.: 1652   1st Qu.: 155.2   1st Qu.: 0.000   1st Qu.: 14.00
##  Median : 2721   Median : 248.5   Median : 1.000   Median : 21.00
##  Mean   : 4494   Mean   : 340.0   Mean   : 2.505   Mean   : 24.53
##  3rd Qu.: 5005   3rd Qu.: 421.3   3rd Qu.: 3.000   3rd Qu.: 32.00
##  Max.   :52309   Max.   :2725.1   Max.   :99.000   Max.   :112.00
```

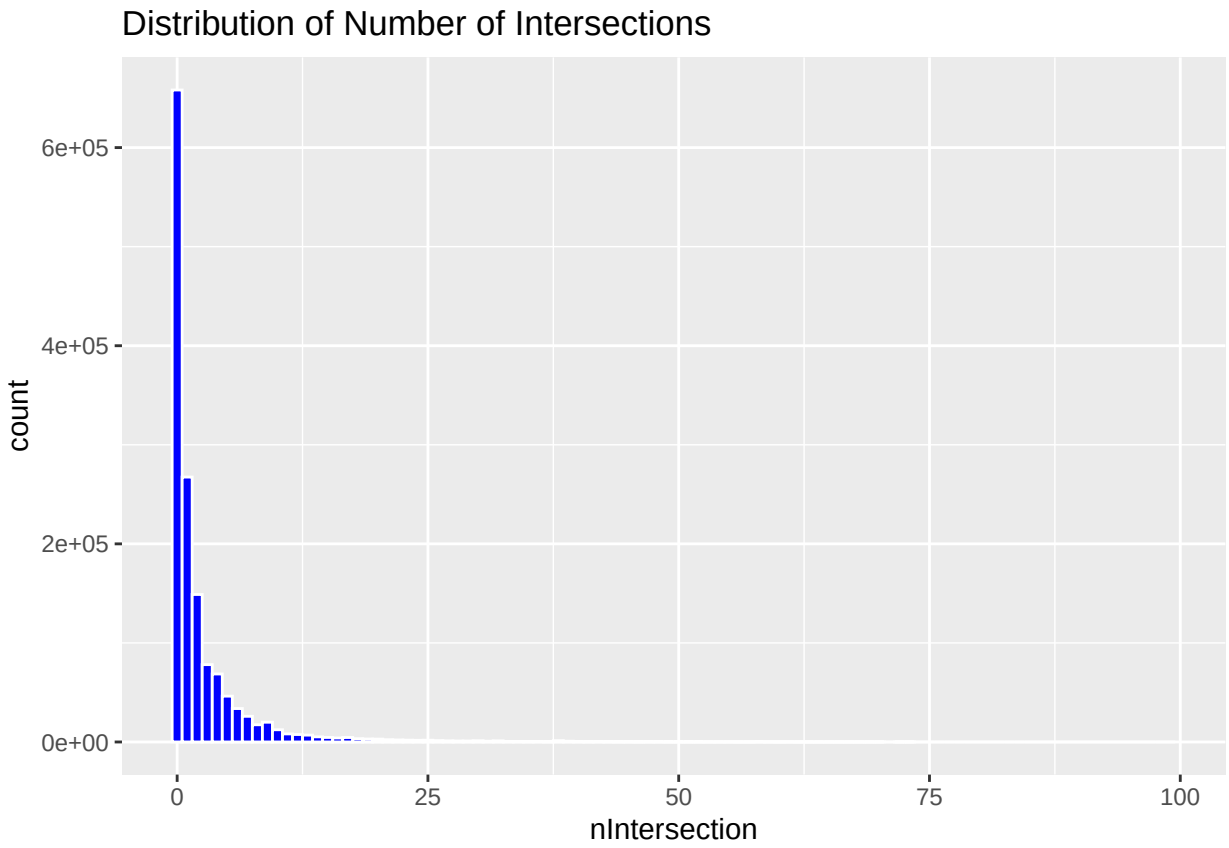
```
# Visualize distributions of the main predictors and target variable
ggplot(large_data_clean, aes(x = distance)) +
  geom_histogram(binwidth = 500, fill = "purple", color = "white") +
  ggtitle("Distribution of Trip Distance")
```



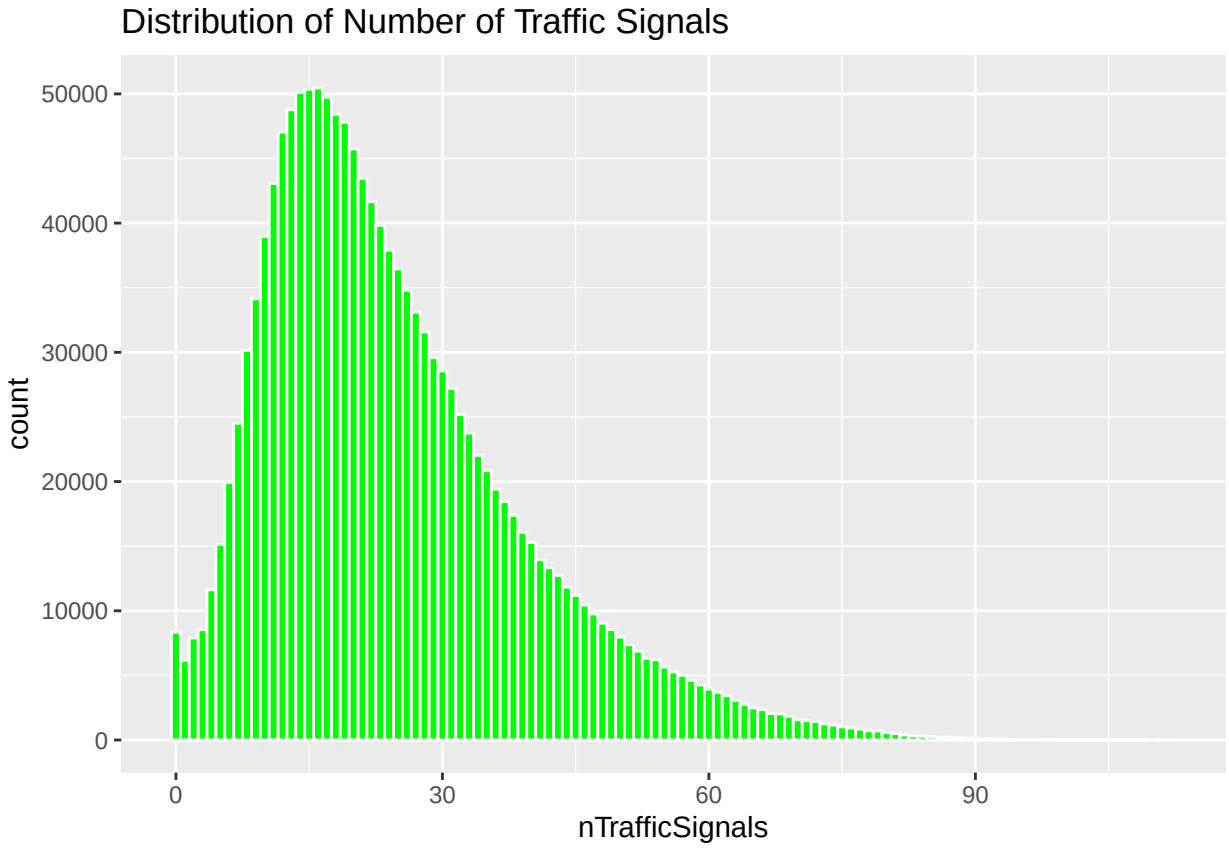

```
ggplot(large_data_clean, aes(x = duration)) +
  geom_histogram(binwidth = 50, fill = "red", color = "white") +
  ggtitle("Distribution of Trip Duration")
```




```
ggplot(large_data_clean, aes(x = nIntersection)) +
  geom_histogram(binwidth = 1, fill = "blue", color = "white") +
  ggtitle("Distribution of Number of Intersections")
```




```
ggplot(large_data_clean, aes(x = nTrafficSignals)) +
  geom_histogram(binwidth = 1, fill = "green", color = "white") +
  ggtitle("Distribution of Number of Traffic Signals")
```



1.4.2.1 Summary Statistics: NYC Taxi Data

- **Trip Distance:**
  - The distribution of `distance` is highly skewed, with the majority of trips covering short distances (median: 2721).
  - There is a long tail, indicating a few trips with significantly larger distances, up to a maximum of 52,309.
- **Trip Duration:**
  - The `duration` distribution is also right-skewed, with most trips being relatively short (median: 248.5 seconds).
  - A few outliers have much longer durations, extending up to 2725.1 seconds.
- **Number of Intersections (`nIntersection`):**
  - This variable is concentrated at the lower end of the scale (mean: 2.5), indicating that most trips encounter only a few intersections.
  - Some trips pass through as many as 99 intersections, adding variability.
- **Number of Traffic Signals (`nTrafficSignals`):**
  - The `nTrafficSignals` distribution follows a skewed pattern, with a mean of 24.53.
  - Most trips pass through fewer than 32 signals, with a small number of trips encountering up to 112 signals.

1.4.2.2 Interpretation: Distribution in NYC Taxi Data

- The skewed distributions in both `distance` and `duration` suggest variability in trip lengths, likely due to urban vs. suburban trip characteristics.
- High variability in intersections and traffic signals aligns with different traffic conditions across routes, highlighting the importance of these features in the model.

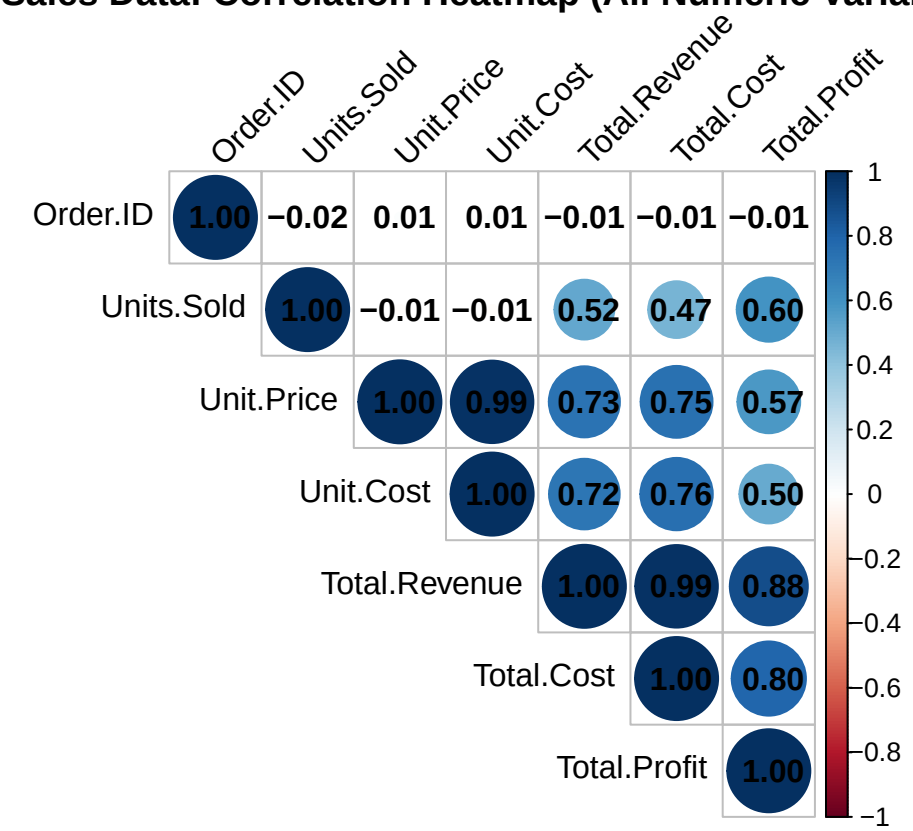
- This analysis supports the selection of features such as `distance`, `duration`, `nIntersection`, and `nTrafficSignals` as predictive variables for modeling trip characteristics.

1.5 Correlation Analysis

1.5.1 For the Sales Data

```
# Correlation matrix for numeric variables in Sales Data
sales_data_numeric <- small_data_clean %>% select_if(is.numeric)

# Calculate and plot the correlation matrix
sales_corr_matrix <- cor(sales_data_numeric, use = "complete.obs")
corrplot::corrplot(sales_corr_matrix, method = "circle", type = "upper",
                   addCoef.col = "black", tl.col = "black", tl.srt = 45,
                   title = "Sales Data: Correlation Heatmap (All Numeric Variables)", mar = c(0, 0, 1, 0
```



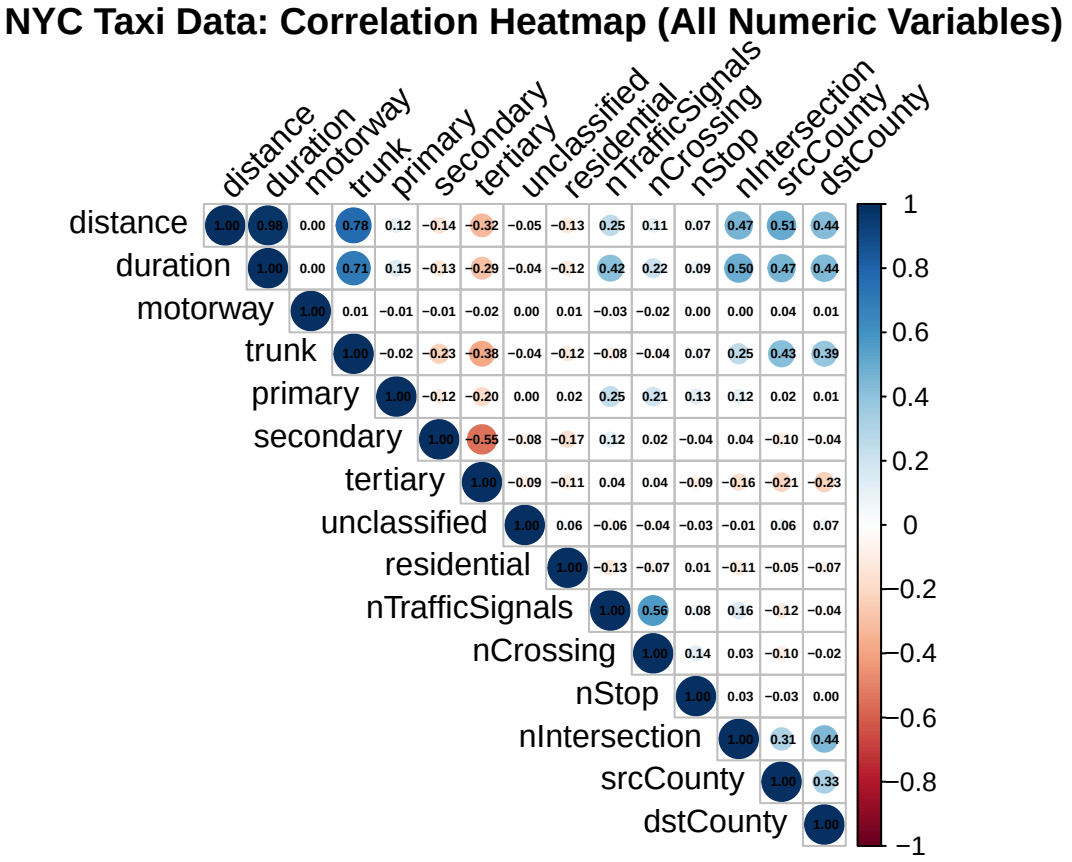
- *Correlation Analysis: Sales Data*
  - **Total Revenue** is highly correlated with both **Units Sold** and **Unit Price**. This is intuitive since revenue is directly impacted by the number of units sold and their price.
  - **Total Cost** is also strongly correlated with **Total Revenue**, which is expected since costs increase with the number of units sold.
  - **Total Profit** is positively correlated with **Total Revenue** and **Units Sold**, as higher revenue typically leads to higher profit.

1.5.2 For the NYC Taxi Trips Dataset

```
# Correlation matrix for numeric variables in NYC Taxi Data
taxi_data_numeric <- large_data_clean %>% select_if(is.numeric)

# Calculate and plot the correlation matrix
taxi_corr_matrix <- cor(taxi_data_numeric, use = "complete.obs")

# Plot correlation heatmap with smaller text for coefficients
corrplot::corrplot(taxi_corr_matrix, method = "circle", type = "upper",
                   addCoef.col = "black", tl.col = "black", tl.srt = 45,
                   number.cex = 0.4, # By adjusting this value we can to resize the coef. text in the c
                   title = "NYC Taxi Data: Correlation Heatmap (All Numeric Variables)",
                   mar = c(0, 0, 1, 0))
```



- *Correlation Analysis: Taxi Trips Dataset*
  - There is a strong positive correlation between **Distance** and **Duration**. Longer trips naturally tend to have longer durations.
  - **nIntersection** has a moderate positive correlation with **Duration** and **Distance**, which makes sense as more intersections might indicate a more urban, congested trip.
  - Some variables, like **nTrafficSignals** and **nCrossing**, show weak correlations with **Distance** and **Duration**, suggesting that traffic-related features might not have a significant impact on trip length in this dataset.

1.6 Moving on to Machine Learning Algorithms

- Now that we have a clear understanding of the data, let's move on to applying machine learning algorithms.
- Based on our earlier discussion, we will proceed with **Linear Regression** for continuous variables and **K-Nearest Neighbors (KNN)** for more complex, non-linear relationships.

II. Applying Machine Learning Algorithms

2.1 Linear Regression on Sales Data

We can use **Linear Regression** to predict **Total Profit** based on variables like **Units Sold**, **Total Revenue**, and **Unit Price**.

```
# Linear Regression on Sales Data
lm_model_sales <- lm(Total.Profit ~ Units.Sold + Unit.Price + Total.Revenue, data = small_data_clean)

# Summary of the model
summary(lm_model_sales)
```

```
##
## Call:
## lm(formula = Total.Profit ~ Units.Sold + Unit.Price + Total.Revenue,
##     data = small_data_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -531043  -95592    1994   71189  604582
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    6.786e+02  5.333e+03   0.127    0.899
## Units.Sold     2.510e+01  9.221e-01  27.226   <2e-16 ***
## Unit.Price    -1.763e+00  1.529e+01  -0.115    0.908
## Total.Revenue  2.020e-01  2.659e-03  75.945   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 166800 on 9996 degrees of freedom
## Multiple R-squared:  0.8048, Adjusted R-squared:  0.8047
## F-statistic: 1.374e+04 on 3 and 9996 DF,  p-value: < 2.2e-16
```

Interpretation of Linear Regression Model Output for Sales Data

- **Model Summary**:
  - *Intercept*: The estimated intercept is 678.6, suggesting that if all predictor variables (**Units Sold**, **Unit Price**, and **Total Revenue**) were zero, the baseline starting point for **Total Profit** would theoretically be 678.6.
  - **Coefficients**:
    - **Units Sold**: The coefficient of 25.10 indicates that for each additional unit sold, the **Total Profit** increases by approximately 25.10 units, assuming all other factors remain constant.
    - **Unit Price**: The coefficient of -1.763 suggests that each additional unit increase in **Unit Price** slightly decreases the **Total Profit** by 1.763 units, holding other variables constant.

However, this relationship is not statistically significant, as indicated by its high p-value (0.908).
    * **Total Revenue**: The coefficient of 0.202 implies that for each additional unit of revenue, the **Total Profit** increases by 0.202 units, assuming other factors remain constant.

- **Statistical Significance**:
  - The predictors **Units Sold** and **Total Revenue** are highly statistically significant, with p-values of less than 0.001, indicated by the $***$ significance codes. This suggests a strong relationship between these variables and **Total Profit**.
  - **Unit Price** does not significantly affect **Total Profit** (p-value = 0.908), indicating it may not be a useful predictor in this model.
- **Model Fit**:
  - *R-squared*: The Multiple R-squared of 0.8048 indicates that approximately 80.48% of the variance in **Total Profit** is explained by the model, suggesting a good fit.
  - *Residual Standard Error*: The residual standard error of 166,800 reflects the average deviation of predictions from actual values in the data, indicating the model's prediction error.
- **F-statistic**: The high F-statistic value (1.374e+04) with a p-value of $< 2.2e-16$ indicates that the model as a whole is statistically significant.
- The model demonstrates that **Units Sold** and **Total Revenue** are strong predictors of **Total Profit**, while **Unit Price** has limited predictive power in this context.

2.2 K-Nearest Neighbors (KNN) on NYC Taxi Data

For the NYC Taxi Data, we can use **KNN** to predict **Trip Duration** based on features like **Distance***, **nIntersection**, and **nTrafficSignals**.

2.2.1 K-Nearest Neighbors (KNN) Modeling on NYC Taxi Data - "too many ties in knn" issues

In this section, we encountered an error while attempting to apply the K-Nearest Neighbors (KNN) algorithm to predict the **Trip Duration** from features like **Distance**, **nIntersection**, and **nTrafficSignals**. The error we encountered was: `"too many ties in knn"`. Below is the initial code used, which led to this issue:

```
### KNN Code Leading to "too many ties in knn" issues

# Prepare data (scale features)
scaled_data <- scale(large_data_clean[, c("distance", "nIntersection", "nTrafficSignals")])

# Create training and testing datasets
set.seed(123)
train_indices <- sample(1:nrow(scaled_data), size = 0.8 * nrow(scaled_data))
train_data <- scaled_data[train_indices, ]
train_labels <- large_data_clean$duration[train_indices]
test_data <- scaled_data[-train_indices, ]
test_labels <- large_data_clean$duration[-train_indices]

## Running KNN = Leading to "too many ties in knn" Error
#k_value <- 7
#knn_model <- knn(train = train_data, test = test_data, cl = train_labels, k = k_value, use.all = FALSE

# Evaluate the model
#table(predicted = knn_model, actual = test_labels)
```

- **Why the Code above is Not Working:**

- The error "too many ties in knn" occurred because many data points in the dataset had identical or very similar feature values.
- KNN calculates the distance between data points to determine the closest neighbors. When multiple points have the same or very close distances to the target, the algorithm cannot resolve the ties, leading to the error.
- In this case, features like **distance**, **nIntersection**, and **nTrafficSignals** in the taxi dataset likely had limited variation, especially in urban trips where trips may have similar characteristics.
- The result was multiple data points with the same or very similar distances, which caused the algorithm to fail.

2.2.2 Updated K-Nearest Neighbors (KNN) Modeling on NYC Taxi Data

- **Solution: Adding Small Random Noise to Break Ties:**
  - To overcome this issue in the previous approach, we introduced **slight random noise** to the features after scaling.
  - This technique, known as adding noise, ensures that no two data points are exactly the same by adding a tiny random value to each feature.
  - The noise is small enough (mean = 0, sd = 0.0001) that it does not alter the meaning of the data but helps the KNN algorithm by breaking ties in the distance calculation.

```
# Sample a subset of the data
set.seed(123)
subset_indices <- sample(1:nrow(large_data_clean), size = 500000)
large_data_sample <- large_data_clean[subset_indices, ]

# Scale features
scaled_data <- scale(large_data_sample[, c("distance", "nIntersection", "nTrafficSignals")])

# Split into training and testing datasets
set.seed(123)
train_indices <- sample(1:nrow(scaled_data), size = 0.8 * nrow(scaled_data))
train_data <- scaled_data[train_indices, ]
train_labels <- large_data_sample$duration[train_indices]
test_data <- scaled_data[-train_indices, ]
test_labels <- large_data_sample$duration[-train_indices]

# Add random noise to break ties
train_data_noisy <- train_data + matrix(rnorm(n = nrow(train_data) * ncol(train_data), mean = 0, sd = 0
test_data_noisy <- test_data + matrix(rnorm(n = nrow(test_data) * ncol(test_data), mean = 0, sd = 0.000

# Run KNN with noisy data
k_value <- 7
knn_model_noisy <- knn(train = train_data_noisy, test = test_data_noisy, cl = train_labels, k = k_value
```

- **Why This Approach Works:**
  - The slight random noise introduced in the features ensures that no two data points have identical values, breaking the ties that were previously causing the algorithm to fail.
  - **Distance Calculation**: KNN relies on calculating the distance between data points. Without noise, multiple data points could be exactly the same in their feature values, leading to tied distances and confusion in the algorithm.
  - **Impact of Noise**: The noise added (mean = 0, sd = 0.0001) is so small that it does not distort the meaning of the data. For example, if a feature like **distance** is originally 10, adding noise might change it to 10.0001, which is imperceptible but enough to break a tie when calculating distances between points.
  - This approach ensures that the KNN algorithm can function correctly by always finding the nearest

neighbors without running into tie issues.
  - I initially chose to work with a sample of the data to reduce computational load and speed up the model testing process, allowing for quicker iterations and evaluations before scaling the model to the entire dataset.

```
# Calculate accuracy
accuracy <- sum(knn_model_noisy == test_labels) / length(test_labels)
accuracy
```

```
## [1] 0.00679
```

- **0.68% Accuracy (0.00679):**
  - *Very Low*: An accuracy of 0.68% suggests that the K-Nearest Neighbors (KNN) model is performing poorly, correctly predicting only a very small fraction of the time.
  - *Likely Issue*: This low accuracy may stem from the high dimensionality of the feature space or the insufficient separation of data points using KNN's distance-based approach.
- **Need for a New Model**:
  - Given the low accuracy observed with K-Nearest Neighbors (KNN) on the NYC Taxi data, it's sensible to consider **Linear Regression** as an alternative model. Linear Regression is particularly effective when dealing with continuous outcome variables like **Trip Duration**, as it assumes a linear relationship between the predictors (such as **Distance**, **nIntersection**, and **nTrafficSignals**) and the target. This model is less computationally intensive than KNN, making it more suitable for large datasets, especially when the relationships in the data can be adequately captured by a linear combination of features. Additionally, Linear Regression has shown strong performance in our analysis of the Sales Data, making it a promising approach to test with the NYC Taxi data.

2.3 Linear Regression for NYC Taxi Data

```
# Fit Linear Regression model on the NYC Taxi data
linear_model <- lm(duration ~ distance + nIntersection + nTrafficSignals, data = large_data_clean)

# View the summary of the model
summary(linear_model)
```

```
##
## Call:
## lm(formula = duration ~ distance + nIntersection + nTrafficSignals,
##     data = large_data_clean)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -261.590  -21.778   -2.401   19.675  252.303
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     1.648e+01  5.890e-02   279.7   <2e-16 ***
## distance        5.128e-02  6.689e-06  7665.9   <2e-16 ***
## nIntersection   2.530e+00  6.437e-03   393.1   <2e-16 ***
## nTrafficSignals 3.537e+00  2.072e-03  1707.2   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 35.38 on 1452045 degrees of freedom
## Multiple R-squared:  0.9848, Adjusted R-squared:  0.9848
## F-statistic: 3.126e+07 on 3 and 1452045 DF,  p-value: < 2.2e-16
```

```r
# Predict on the dataset
predicted_duration <- predict(linear_model, large_data_clean)

# Calculate RMSE to evaluate model performance
actual_duration <- large_data_clean$duration
rmse <- sqrt(mean((predicted_duration - actual_duration)^2))
rmse
```

```
## [1] 35.37794
```

- **Model Summary**:
  - *Intercept*: The estimated intercept is 16.48, suggesting that when all predictor values are zero, the trip duration would start from this baseline.
  - *Coefficients*:
    - **Distance**: The coefficient of 0.05128 indicates that for each additional unit increase in distance, the trip duration increases by approximately 0.051 units, holding other variables constant.
    - **nIntersection**: The coefficient of 2.530 suggests that each additional intersection contributes approximately 2.53 units to the trip duration, holding other variables constant.
    - **nTrafficSignals**: Each additional traffic signal adds approximately 3.537 units to the trip duration.
- **Statistical Significance**:
  - All predictors are statistically significant at the $p < 0.001$ level, as shown by the $***$ significance codes, indicating a strong relationship between these variables and trip duration.
- **Model Fit**:
  - *R-squared*: The Multiple R-squared of 0.9848 indicates that approximately 98.48% of the variance in trip duration is explained by the model, suggesting an excellent fit.
  - *Residual Standard Error*: The residual standard error of 35.38 reflects the average amount by which the predictions differ from the actual trip durations. While small, this still indicates some variation that the model does not capture.
- **F-statistic**: The very high F-statistic value (3.126e+07) with a p-value of $< 2.2e-16$ indicates that the model, as a whole, is statistically significant.

With this strong performance, Linear Regression appears well-suited for predicting trip duration in the NYC Taxi data.

2.4 Algorithm Selection Rationale

- **Linear Regression**:
  - We selected Linear Regression for both the **Sales Data** and the **NYC Taxi Data** due to the continuous nature of the target variables (**Total Profit** in the Sales Data and **Trip Duration** in the NYC Taxi Data). Linear Regression is ideal when there is a linear relationship between the predictors and the target variable.
  - In the Sales Data, **Total Profit** likely has a direct relationship with variables like **Units Sold** and **Total Revenue**, which justifies using Linear Regression.
  - Similarly, in the NYC Taxi Data, **Trip Duration** might be linearly affected by factors like **Distance**, **nIntersection**, and **nTrafficSignals**, making Linear Regression a strong candidate for capturing these relationships.
- **K-Nearest Neighbors (KNN)**:
  - K-Nearest Neighbors was initially chosen for the NYC Taxi Data to test its ability to capture any non-linear relationships among the features, which could be relevant in predicting trip duration in varied traffic and route conditions.
  - KNN is a non-parametric algorithm, which can be beneficial when data relationships are not strictly linear. However, we encountered limitations with KNN due to high computational costs and low accuracy, prompting a switch to Linear Regression for better performance.

2.5 Model Limitations

- **Linearity Assumption**:
  - Linear Regression assumes a linear relationship between predictors and the target variable. In the case of the NYC Taxi Data, this model may not fully capture the complexity of factors affecting **Trip Duration**. For instance, the effects of **distance**, **intersections**, and **traffic signals** may not be entirely linear in real-world conditions, where variations in traffic or road conditions can impact trip duration non-linearly.
- **Sensitivity to Outliers**:
  - Linear Regression is sensitive to outliers, which can distort predictions and reduce model accuracy. In both datasets, unusually high or low values in **Units Sold** or **Trip Duration** can disproportionately influence the model. For the NYC Taxi Data, trips with very high durations or distances may not align well with typical travel patterns, affecting the model's overall predictive accuracy.
- **Limitations of K-Nearest Neighbors (KNN)**:
  - KNN, which was initially considered for the NYC Taxi Data, encountered computational limitations and performed poorly, especially due to its high reliance on calculating distances in high-dimensional spaces. This challenge, coupled with low accuracy, prompted a switch to Linear Regression.
- **Potential for Non-Linear Models**:
  - While Linear Regression provides interpretable and computationally efficient results, exploring non-linear models (e.g., Decision Trees or Random Forests) could capture more complex relationships within the data, especially in the NYC Taxi Data where non-linear interactions among **distance** and **traffic features** may exist.
- Despite these limitations, Linear Regression proved effective for both datasets, particularly in explaining **Total Profit** in the Sales Data. Future analyses could incorporate non-linear models to potentially improve predictive performance and capture more complex patterns in the NYC Taxi Data.

III. Comparison of Model Performance and Dataset Size Impact

3.1 Evaluate Performance with Metrics

- **Sales Data**:
  - For the Sales Data model, we observed an *R^2 of 80.48%*, meaning that **Units Sold** and **Total Revenue** explained around 80% of the variance in **Total Profit**. The *Residual Standard Error* (166,800) reflects a relatively high degree of variability in predictions, indicating that while the model fit is strong, additional features could improve predictive power.
  - *Impact of Size on Metrics*: The smaller size of the Sales Data provided a manageable dataset for efficient model fitting. However, the limited sample size and feature set constrained the model's explanatory power, resulting in a lower R^2 compared to the larger dataset.
- **NYC Taxi Data**:
  - For the NYC Taxi Data, the model achieved an *R^2 of 98.48%* with a *Residual Standard Error* of 35.38, showing an exceptionally good fit. This high R^2 suggests that **Distance**, **nIntersection**, and **nTrafficSignals** accounted for nearly all variance in **Trip Duration**. The large dataset size contributed to this high accuracy by allowing the model to capture subtle patterns and relationships.
  - *Impact of Size on Metrics*: The large sample size led to a much higher R^2, as the abundance of data enabled the model to generalize more effectively. The rich dataset likely allowed Linear Regression to capture nuanced relationships, resulting in a more precise model with lower residual error compared to the smaller Sales Data.

3.2 Comparing Computational Time and Efficiency

- **Large Dataset (NYC Taxi Data)**:
  - The NYC Taxi Data's size posed significant computational challenges, especially with K-Nearest Neighbors (KNN), which was computationally expensive and low-performing, achieving an accuracy

of just 0.68%. This prompted a shift to Linear Regression, which was computationally faster and more compatible with large datasets.
  - While Linear Regression processed the larger dataset with relatively good efficiency, the model fitting and prediction times were naturally longer than with the smaller Sales Data.
- **Small Dataset (Sales Data)**:
  - The smaller dataset size allowed for quick computation with Linear Regression, providing efficient and manageable processing times. However, the smaller data volume meant that the model may not have captured all nuances, as reflected by the lower R^2.

3.3 Challenges and Conclusion

- **Comparative Impact of Dataset Size**:
  - The large dataset size in the NYC Taxi Data facilitated a high R^2 and precise model, likely due to more comprehensive data coverage. Conversely, while the Sales Data model fit well, the smaller sample size limited its predictive capability.
- **Practical Implications**:
  - The NYC Taxi Data's larger size allowed for more robust predictions but required processing adaptations to balance computational demands with model effectiveness. The Sales Data model provided quick insights with efficient computation, but adding more data or features could enhance its predictive power.

IV. Final Conclusion and Recommendations}

4.1 Conclusion:

- This project aimed to explore and analyze two datasets, the **Sales Data** and the **NYC Taxi Data**, using Linear Regression as the primary modeling approach.
- For the **Sales Data**, Linear Regression performed well, with an *R^2 of 80.48%*, indicating that **Units Sold** and **Total Revenue** are strong predictors of **Total Profit**. The model, however, showed a relatively high Residual Standard Error (RSE is 166,800), partly due to outliers in Total Profit.
- In the **NYC Taxi Data**, Linear Regression achieved an *R^2 of 98.48%*, suggesting that **Distance**, **nIntersection**, and **nTrafficSignals** together explained nearly all the variation in **Trip Duration**. The large dataset size contributed to this high R^2, though it required careful handling to manage computational demands.

4.2 Recommendations for Future Analysis:

- **Explore Non-Linear Models**: For the NYC Taxi Data, future work could benefit from testing non-linear models, such as Decision Trees or Random Forests, to potentially capture any complex interactions among predictors.
- **Additional Features**: In the Sales Data, adding more features related to pricing, regional factors, or customer segments could provide further insights and potentially improve model performance by reducing unexplained variance.
- **Outlier Handling**: To improve RSE and overall model stability, consider handling outliers in Total Profit through capping extreme values or using robust regression methods, especially for business decisions that prioritize consistency.

4.3 Practical Applications:

- **Sales Data**: Insights from the Sales Data model can help businesses understand profitability drivers. For example, identifying how **Units Sold** and **Total Revenue** influence **Total Profit** can aid in decision-making for sales strategies and resource allocation.
- **NYC Taxi Data**: The high R^2 in the NYC Taxi Data model suggests that distance and traffic-related factors are critical to predicting trip duration, which can help improve route planning and time

estimations, benefiting both operational efficiency and customer satisfaction.

In summary, Linear Regression proved effective for both datasets, providing interpretable and useful insights. While the current analysis is informative, further exploration with additional features and advanced models could yield even more accurate predictions, especially for datasets with complex relationships or outliers.
