FINAL R CODE RETAIL_PERFORMANCE_MINED.R LINK TO GITHUB GITHUB

EXECUTIVE SUMMARY

No data collection process was undergone for this assignment; our raw data is sourced from Kaggle, and initially includes 1,650 records and 12 fields. The structure of our dataset is cross-sectional, representing a sample of multiple stores observed at a specific point in time. Since the data is static, our project primarily focuses on inter-observational comparisons, descriptions, and visuals. Each record within the dataset represents a retail store in the greater California area, and our most frequently used performance metric was Monthly Sales Revenue, representing total sales revenue in thousands of dollars.

The remaining variables in our dataset include the following: StoreSize, representing the physical size of the store in square meters, was later renamed to Store_Size. MarketingSpend, measuring the monthly expenditure on marketing activities in thousands of dollars, was renamed Marketing_Expense. Customer visits were captured under CustomerFootfall, later renamed to Traffic, indicating the total number of customers visiting the store each month. The diversity of products available in a store was recorded as ProductVariety, renamed to Product_Variety, and employee performance was assessed using Employee Efficiency, a scaled score from 0 to 100, which was renamed Staff_Efficiency. Additionally, StoreAge, representing the number of years a store has been operational, was renamed Store_Age, while CompetitorDistance, indicating the distance to the nearest competitor in kilometers, was renamed Competitor_Distance. Monthly promotional efforts were tracked through PromotionsCount, renamed Monthly_Promotions, and the broader economic landscape was reflected in EconomicIndicator, a regional activity index renamed Economic_Indicator. Last but not least, store-specific categorical variables included StoreLocation, renamed to Location, and StoreCategory, renamed Store_Type, which identified the type of store, such as "Grocery," "Electronics," or "Clothing."

DATA CLEANING

The first step in our data cleaning process was to check for completeness as well as uniqueness and range constraint violations. No imputation or deletion methods were employed for outliers as we believe they are important for further analysis (i.e. how did other variables impact a store's ability to create an outlier in Monthly Sales, etc)

Null or Missing Values

```
> ## Check missing values ##
> sum(is.na(stores_data))
[1] 0
```

We began by checking for nulls in the data set to determine if we were going to have to delete the records or impute missing values using summarization arguments. We found that there were no missing values in the dataset and concluded that no data cleaning measures were necessary.

Full Duplicates

```
> ## Check duplicate records ##
> sum(duplicated(stores_data))
[1] 0
```

Because the data set did not contain a unique identifier field we wanted to make sure that there were no duplicate records that had to be evaluated and removed as needed. Because there was no key identifier we knew searching for full duplicates would be the most effective method, as full duplicates are considered records with matching observations across each field. We found that there were no full duplicates throughout the records in the dataset.

Values Out of Range

```
> ## Check range constraints of Employee Efficiency Field ##
> sum(stores_data$EmployeeEfficiency < 0 & stores_data$EmployeeEfficiency > 100)
[1] 0
```

In the codebook of the dataset it was mentioned that the values in the field measuring employee efficiency were a performance indicator on a scale of 1 to 100. We wanted to make sure that all values within this field respected this range constraint and made sense according to their intended use outlined in the codebook.

Categorical Variable Formatting Inconsistencies

```
## Check for inconsistency within categorical variables ##
Location_var_check <- stores_data %>%
  count(StoreLocation)
View(Location_var_check)
Category_var_check<- stores_data %>%
  count(StoreCategory)
View(Category_var_check)
```

We had to check for consistency across categorical variable classes to ensure that there were no typos, case inconsistencies, or duplicates due to similar errors to make our data filtering and manipulation easier and more accurate.

Converting Numeric Formats

```
## MonthlySalesRevenue*1000 for standardized units ##
stores_data5<- stores_data3 %>%
  mutate(MonthlySales = ifelse(MonthlySalesRevenue < 999, MonthlySalesRevenue*1000, 0))

## Check for successful field modification ##
Sales_before_vs_after<- stores_data5 %>%
  select(MonthlySalesRevenue, MonthlySales)
View(Sales_before_vs_after)
```

	MarketingSpend	MarketingExp
1	29	29000
2	31	31000
3	35	35000
4	9	9000
5	35	35000
6	43	43000

```
## MarketingSpend*1000 for standardized units ##
stores_data2<- stores_data %>%
  mutate(MarketingExp = ifelse(MarketingSpend < 100, MarketingSpend*1000, 0))

## Check for successful field modification ##
Marketing_spend_before_vs_after<- stores_data2 %>%
  select(MarketingSpend,MarketingExp)
View(Marketing_spend_before_vs_after)
```

	MonthlySalesRevenue	MonthlySales
1	284.90	284900
2	308.21	308210
3	292.11	292110
4	279.61	279610
5	359.71	359710
6	258.21	258210

We converted numeric columns into a standardized format to get a better picture of each value's magnitude. This was done by leveraging the DPLYR's mutate function with an if-else argument. The code book had noted that the original column values were in (thousands) which is how we were able to determine the scale of the new fields. Pictured are the befores and after of the fields once the conversion was performed. It is also worth noting that we removed the original fields once this was completed.

Renaming Columns

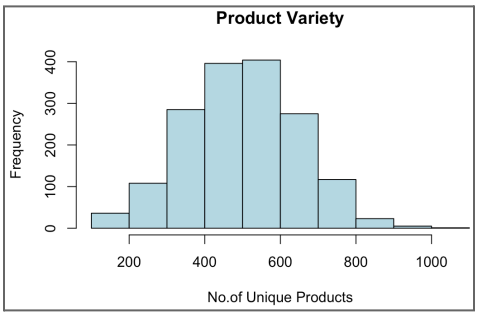
```
## Renaming columns for clarity purposes ##
col_rename_stores_data <- stores_data6 %>%
  rename(
    Product_Variety = ProductVariety,
    Traffic = CustomerFootfall,
    Store_Size = StoreSize,
    Staff_Efficiency = EmployeeEfficiency,
    Store_Age = StoreAge,
    Competitor_Distance = CompetitorDistance,
    Monthly_Promotions = PromotionsCount,
    Economic_Indicator = EconomicIndicator,
    Location = StoreLocation,
    Store_Type = StoreCategory,
    Marketing_Expense = MarketingExp,
    Monthly_Sales = MonthlySales)
```

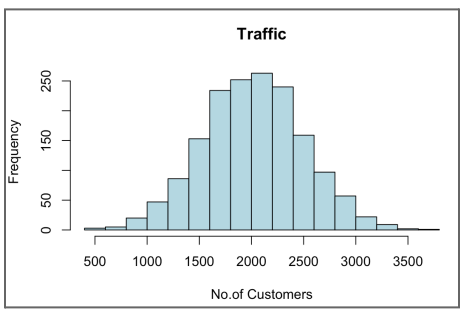
```
> head(col_rename_stores_data, 2)
  Product_Variety Traffic Store_Size Staff_Efficiency Store_Age Competitor_Distance Monthly_Promotions
1             581    1723        186             84.9         1                 12                 6
2             382    1218        427             75.8        18                 11                 6
  Economic_Indicator    Location  Store_Type Marketing_Expense Monthly_Sales
1              108.3 Los Angeles Electronics             29000        284900
2               97.8 Los Angeles Electronics             31000        308210
```

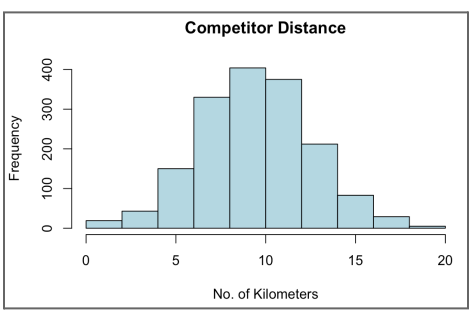
Originally, column headers with two words were blended together. We decided to add underscores to separate the words for easier readability. Additionally we changed the names of a few fields, for example "CustomerFootfall" was shortened to "Traffic" as we believe the condensed word was a better descriptor for what the code book assigned to it, the number of customers visiting the retail stores per month. We changed "PromotionsCount" to "Monthly_Promotions" to better describe the frequency of promotions counted that was reflected in the code book. We shortened "StoreCategory" to "Store_Type" to signify we were examining the types of goods sold (electronics, clothing, and groceries), and to shorten the name of the field. Finally, we changed "StoreLocation" to "Location" to shorten the column header.

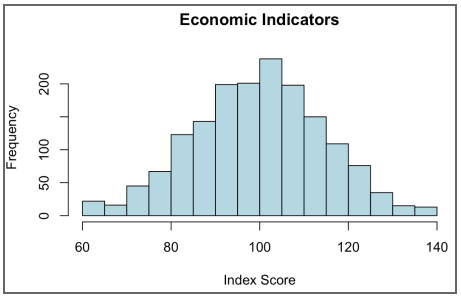
EXPLORATORY ANALYSIS

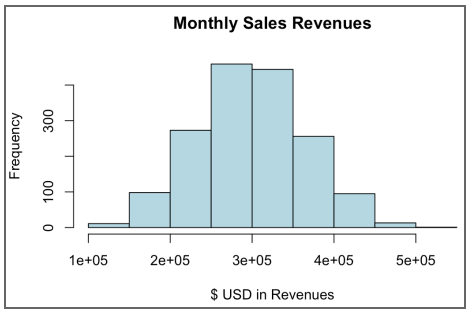
Variable Distributions

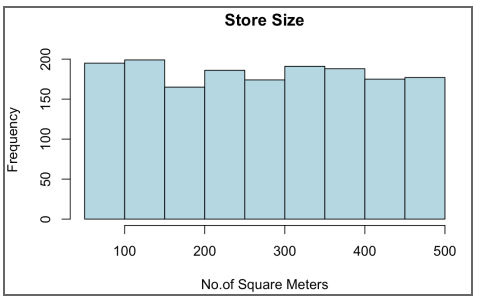


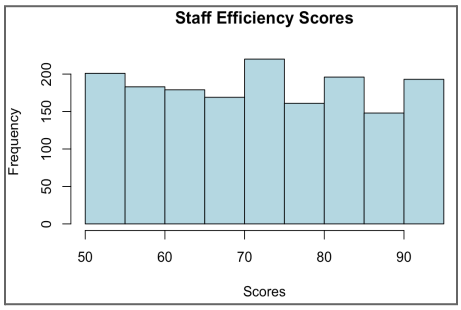


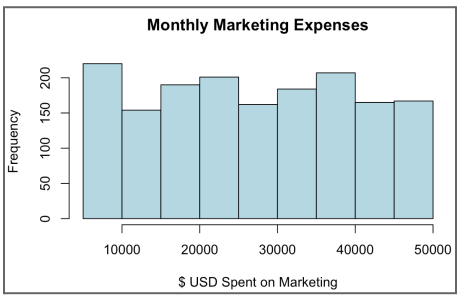


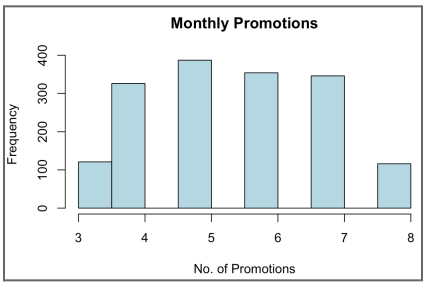








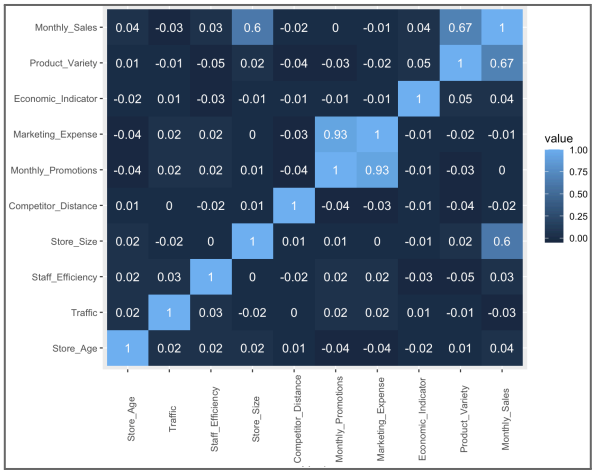




It can be seen that the distribution of observations for the Product Variety, Traffic, Competitor Distance, Economic Indicator and Monthly Sales Revenue variables are fairly normal. The distributions for Store Size, Staff Efficiency Scores, Store Age, and Monthly Marketing Expenses appear to be uniform. The variable Monthly Promotions appears to have no values that fall within the bins between 4 to 4.5, 5 to 5.5, 6 to 6.5, and 7 to 7.5.

Exploring Relationships: Correlations heatmap

```
### Correlation Heat Map ###
# Use DPLYR to only get numeric variables #
stores_data_numeric <- final_stores_data %>%
  select(-Store_Type, -Location)
# round correlation data to 2 decimal places) #
correlation_matrix<- round(cor(stores_data_numeric), 2)
head(correlation_matrix)
# re-ordering correlation matrix by coefficient value #
install.packages("reshape2")
library(reshape2)
distance<- as.dist((1-correlation_matrix)/2)
# cluster hierarchically #
clustered<- hclust(distance)
correlation_matrix<- correlation_matrix[clustered$order, clustered$order]
# melt correlation matrix #
melted_correlation_matrix <- melt(correlation_matrix)
melted_correlation_matrix
# plot heat map using ggplot #
correlation_heatmap<- ggplot(melted_correlation_matrix, aes(Var1,Var2, fill = value)) +
  geom_tile() +
  geom_text(aes(Var2, Var1, label = value), color = "white", size = 4) +
  theme(axis.text.x = element_text(angle = 90))
correlation_heatmap
```



The strongest positive relationships reside between Monthly Sales & Staff Efficiency, Marketing Expense & Monthly Promotions, and Monthly Sales & Product Variety. It is interesting to note that Marketing Expense does not have a relationship with Monthly Promotions as one would expect that an increase in marketing expenditures would have a positive effect on monthly sales. Both product variety and store size having a positive impact on monthly sales makes sense as a larger store size accommodates a larger assortment of products. There were no strong negative correlations among the variables. Source cited is where coding applied was learned from.

Leveraging SQL to Perform a Table Join to Explore Proportional Performance Metrics

```
### ***CREATE A DATAFRAME WITH SALES PERFORMANCE METRICS*** ###
### *** TO BE JOINED WITH ORIGINAL DATA FRAME***###
df2<- final_stores_data %>%
  mutate(Cost_Per_Customer = Marketing_Expense/ Traffic)
head(df2)
df3<- df2 %>%
  mutate(Sales_Per_Customer = Monthly_Sales / Traffic)
head(df3) #df3 is the data table we will join to explore sales per customer &
# marketing spend per customer

### DATA FRAME WITH SALES PER CUSTOMER AND MARKETING SPEND PER CUSTOMER FOR EACH STORE ###
stores_data_table2<- df3 %>%
  select(Location, Cost_Per_Customer, Sales_Per_Customer)
class(stores_data_table2)
head(stores_data_table2)
dim(stores_data_table2)
library(sqldf)

### USE A WHILE LOOP TO CREATE A VECTOR THAT LABELS OBSERVATION NUMBERS IN DATA TABLES ####
x<- 0
n<-1
while (n<=1650) {
  x[n]<-n
  n <- n +1
}
x
```

```
#### ADD VECTOR WITH OBSERVATION NUMBERS TO DATA FRAME AS A NEW COLUMN TO BOTH TABLES ###
fsd_x<- cbind(final_stores_data, data.frame(x))
head(fsd_x)
##fsd_x is final_stores_data with observation number column ##
fsd2_x<- cbind(stores_data_table2, data.frame(x))
head(fsd2_x)
##fsd2_x is stores_data_table2 with observation number column ##

### USE SQL TO JOIN TABLES WITH OBSERVATION NUMBER COLUMN AS PRIMARY KEY ###
joinquery<- "SELECT fsd_x.Location, fsd_x.Store_Type, fsd_x.Product_Variety,
fsd_x.Traffic, fsd_x.Store_size, fsd_x.Staff_Efficiency, fsd_x.Store_Age,
fsd_x.Competitor_Distance, fsd_x.Monthly_Promotions, fsd_x.Economic_Indicator,
fsd_x.Marketing_Expense, fsd_x.Monthly_sales, fsd2_x.Location, fsd2_x.Cost_Per_Customer,
fsd2_x.Sales_Per_Customer, fsd2_x.x, fsd_x.x
FROM fsd_x
FULL JOIN fsd2_x
USING (x)"
fsd_joined<- sqldf(joinquery)
```

To perform an SQL join we created a data frame where we added two columns measuring performance metrics using SQL's mutate function with if else statements. The "Sales Per Customer" field was created by taking Monthly Sales divided by Traffic. This metric is useful to evaluate how much customers are spending on their visit, on average. This can allow stores to evaluate whether their employees are able to upsell products to increase average order value. The "Cost Per Customer" field was created taking Marketing Expense divided by Traffic. This was helpful to measure how much each store was spending to get the customers in the door. If the cost per customer is high, they can determine how to make their marketing more effective to reach more customers, or experiment with marketing budget costs.

The while loop was used to create a primary key for the original data table and a foreign key for the data table with the performance ratios we created. A data frame of one column called "x" was created and column-binded to both the original data frame and the performance ratios table. "X" would be the primary key on the original table, and the foreign key in the ratios table, on which the tables would be joined on. A full join query was prompted as we knew there were an equal number of records in both tables.

Query 1: Top Sales Performance

```
# Sorting records in descending order to find top
# performers in monthly sales
query1 <- "SELECT *
          FROM final_stores_data
          ORDER BY Monthly_Sales DESC
          LIMIT 5"
sqldf(query1)
```

	Location	Store_Type	Product_Variety	Traffic	Store_Size	Staff_Efficiency	Store_Age
1	Palo Alto	Clothing	1092	2257	434	78.3	15
2	Los Angeles	Grocery	873	2747	480	76.0	25
3	Palo Alto	Grocery	879	1742	423	69.5	16
4	San Francisco	Grocery	762	1510	475	52.8	16
5	Sacramento	Clothing	779	1851	477	89.5	5

	Competitor_Distance	Monthly_Promotions	Economic_Indicator	Marketing_Expense	Monthly_Sales
1	10	7	100.9	42000	534260
2	7	8	98.4	45000	496700
3	8	6	108.9	28000	492380
4	7	5	107.1	18000	479270
5	9	5	111.4	26000	471580

To understand the data, we analyze the variables and their behavior, as seen above. By sorting the data in descending order of Monthly Sales, we focus on the top-performing stores based on revenue. The top five stores achieve significant figures, with the highest store bringing in 534,260 and the fifth 471,580. It is important to note where these stores are located, Palo Alto, Los Angeles, Palo Alto, San Francisco and Sacramento, all major cities in California. Geographical factors help us to understand how location may contribute to their success. Store Types such as Clothing or Grocery may also help us to predict which industries drive higher revenue. Other operational variables, such as Product Variety, allow us to highlight the diversity of offerings within each store. Other metrics, such as Staff Efficiency or Store Size could be correlated to better performance. Finally, examining Marketing Expense and the number of Monthly Promotions helps identify strategies that drive revenue growth. By observing these variables, we gain a clearer understanding of the factors contributing to the success of top-performing stores, which is essential for making accurate predictions and informed business decisions.

Query 2: Finding the Average Monthly Sales, Marketing Expenses, and Traffic by Store Type

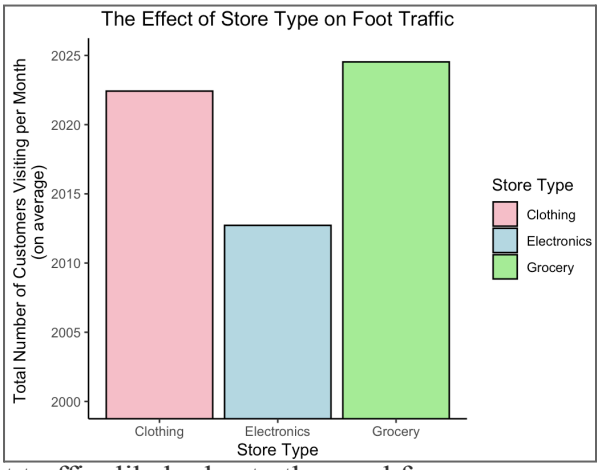
```
# Find average marketing expense, average traffic, and average monthly sales
# for each store type
query2 <-"SELECT Store_Type, AVG(Marketing_Expense) AS AVG_Marketing_Expense,
          AVG(Traffic) AS AVG_Traffic,
          AVG(Monthly_Sales) AS AVG_Monthly_Sales
          FROM final_stores_data
          GROUP BY Store_Type
          ORDER BY AVG_Monthly_Sales DESC"
```

	Store_Type	AVG_Marketing_Expense	AVG_Traffic	AVG_Monthly_Sales
1	Grocery	27456.56	2024.532	301325.6
2	Clothing	26945.08	2022.424	300863.1
3	Electronics	27971.96	2012.723	295390.8

The primary goal of this query was not only to gain more insight into specific ad spend benchmarks but also to establish dataset credibility. For example, it is reasonable to infer that grocery stores experience the highest amount of foot traffic because food is an essential, replenishable necessity. If our data were to show otherwise, it could indicate altercations with the sample observations, such as outliers and missing/null values. Additionally, since it will not be used in the visual below, we could potentially omit the AVG_Marketing_Expense mutation.

Visualization: The Effect of Store Type on Foot Traffic by Store Type

```
# Create a working data frame to generate a bar chart
avg_foottraffic <- data.frame(Store_Type = c("Grocery", "Clothing", "Electronics"),
                              Traffic = c(2024.532, 2022.424, 2012.723))
summary(avg_foottraffic)
as_factor(avg_foottraffic$Store_Type)
class(avg_foottraffic$Traffic)
class(avg_foottraffic$Store_Type)
# Plot the bar chart to visualize the effect of store type on foot traffic
ggplot(avg_foottraffic, aes(x = Store_Type, y = Traffic, fill = Store_Type)) +
  geom_col(stat = "identity", color = "black") +
  labs(title = "The Effect of Store Type on Foot Traffic",
       x = "Store Type",
       y = "Total Number of Customers Visiting per Month
       (on average)") +
  coord_cartesian(ylim=c(2000, 2025)) +
  theme_classic() +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_fill_manual(values = c("Grocery" = "lightgreen",
                               "Electronics" = "lightblue",
                               "Clothing" = "pink"),
                    name = "Store Type")
```



As expected, grocery stores show the highest average foot traffic, likely due to the need for frequent visits. Clothing stores closely follow, with fairly high foot traffic that aligns with

shopping for seasonal or lifestyle needs. It could also be inferred from the data that fashion consumers like in-person shopping for the convenience of trying on products before making a purchase. Lastly, electronics stores have the lowest average foot traffic, which could be attributed to their products being higher-ticket, less frequently purchased items, often researched or purchased online.

Query 3: Finding Top Performer in Average Monthly Sales by Store Location

```
# Sort records in descending order to find top performer
# in average monthly sales by location
query3 <-"SELECT Location, AVG(Marketing_Expense) AS AVG_Marketing_Expense,
          AVG(Monthly_Sales) AS AVG_Monthly_Sales
          FROM final_stores_data
          GROUP BY Location
          ORDER BY AVG_Monthly_Sales DESC"
sqldf(query3)
```

	Location	AVG_Marketing_Expense	AVG_Monthly_Sales
1	Los Angeles	26832.13	301191.3
2	San Francisco	27594.20	299414.1
3	Palo Alto	27454.55	298366.1
4	Sacramento	27966.02	298006.5

To gain deeper insights into the dataset, we created this query to analyze key metrics across each location. By grouping the data by major cities, we can observe patterns in marketing expenses and their relationship to average monthly sales. Sorting the results in descending order of average monthly sales reveals that Los Angeles leads with the highest average sales of $301,191.30 and shows significant marketing investment, with an average marketing expense of $26,832.13. This trend suggests a potential relationship between higher marketing expenditures and stronger sales performance. San Francisco, Palo Alto, and Sacramento follow closely behind, with lower average sales but similar marketing expenses. Understanding these location-specific trends allows businesses to evaluate the effectiveness of their marketing strategies and allocate resources more efficiently to maximize revenue growth.

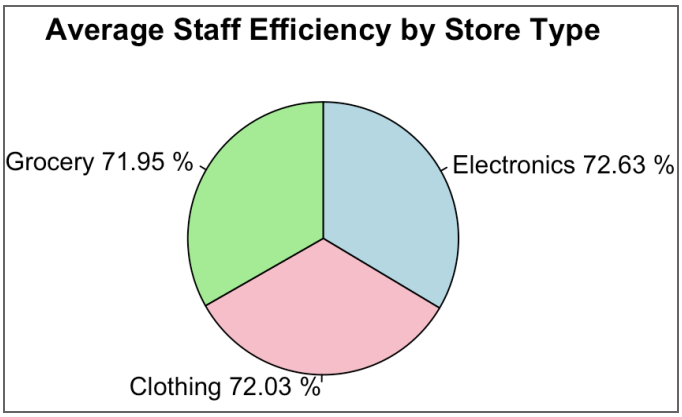
Query 4: Explore Staff Efficiency by Store Type

```
# Explore level of staff efficiency by store type
query4 <-"SELECT Store_Type, AVG(Staff_Efficiency) AS AVG_Staff_Efficiency
          FROM final_stores_data
          GROUP BY Store_Type
          ORDER BY AVG(Staff_Efficiency) DESC"
```

The query analyzes average staff efficiency across different store types by grouping the data and calculating the mean staff efficiency for each group. The results are then ordered in descending order to identify which store type has the highest efficiency. This analysis reveals how staff performance varies between store types, providing valuable insights into operational efficiency. By ranking store types based on average staff efficiency, businesses can identify where staff performance is strongest and explore potential reasons behind these differences, such as streamlined processes, training programs, or workload variations. This information can be used to set benchmarks and implement strategies to improve staff efficiency across all store categories.

Visualization: Average Staff Efficiency by Store Type

```
## Pie chart to visualize average staff efficiency by store type#
pie_data <- sqldf(query4)
labels <- paste(pie_data$Store_Type, round(pie_data$AVG_Staff_Efficiency, 2), "%")
colors <- c("lightblue", "pink", "lightgreen")
pie(pie_data$AVG_Staff_Efficiency, labels = labels, col = colors, edges = 250,
    radius = 1 , clockwise = TRUE, main = "Average Staff Efficiency by Store Type")
```



The analysis reveals that average staff efficiency is fairly similar across different store types, ranging from 71.95 to 72.63. By grouping the data by store type and ordering the results in descending order of average staff efficiency, we observe that Electronics stores hold a slight lead with an average efficiency of 72.63, followed closely by Clothing stores at 72.03 and Grocery stores at 71.95. Although the differences are minimal, this ranking suggests that Electronics stores may operate with slightly more efficient staff, potentially due to factors like better training, streamlined operations, or a lower staff-to-customer ratio.

Query 5: Creating a Binary Variable Field for if Staff Efficiency is Above Average or Not

```
#Use DPLYR to create a field where "YES" means staff efficiency is above the average
#across all stores for that particular store, and "NO" means staff is below average
fsd_w_YN_staff<- final_stores_data %>%
  mutate(SE_aboveavg = ifelse(Staff_Efficiency< mean(Staff_Efficiency), "NO", "YES"))
head(fsd_w_YN_staff)
#Use sql to transform into a binary field for aggregation & analysis
query5<- "SELECT *,
CASE
WHEN SE_aboveavg = 'YES' THEN 1
WHEN SE_aboveavg = 'NO' THEN 0
END AS staff_effic_aboveavg
FROM fsd_w_YN_staff"
fsd_w_binary_staff_effic<- sqldf(query5)
```

Used the “CASE” statement to create conditions so SQL could properly encode the binary variable column named staff_effic_aboveavg” using the “YES/NO” statements mutated to the original data frame. A “YES” means staff efficiency at that store was above average, and corresponds to a 1 in the binary field of the SQL table. A “NO” means staff efficiency at that store was below average, and corresponds to a 0 in the binary field of the SQL average.

Query 6: Counting Stores With Staff Efficiency Above Average by Store Type and Location

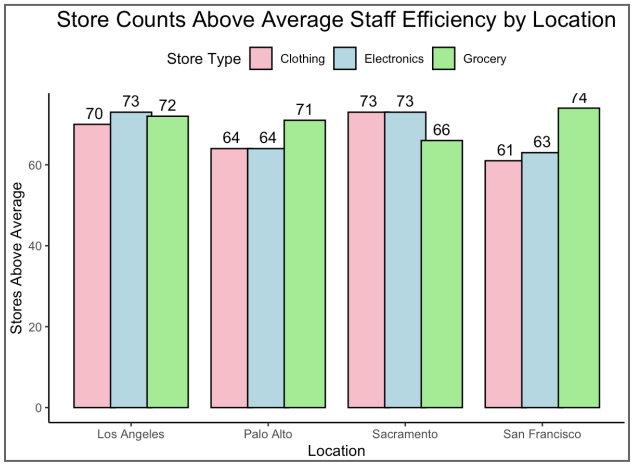
```
query6 <- "SELECT Location, Store_Type, COUNT(*) AS stores_above_avg
        FROM fsd_w_binary_staff_effic
        WHERE staff_effic_aboveavg = 1
        GROUP BY Location, Store_Type
        ORDER BY Location, Store_Type"
count_above_each <- sqldf(query6)
count_above_each
```

	Location	Store_Type	stores_above_avg
1	Los Angeles	Clothing	70
2	Los Angeles	Electronics	73
3	Los Angeles	Grocery	72
4	Palo Alto	Clothing	64
5	Palo Alto	Electronics	64
6	Palo Alto	Grocery	71
7	Sacramento	Clothing	73
8	Sacramento	Electronics	73
9	Sacramento	Grocery	66
10	San Francisco	Clothing	61
11	San Francisco	Electronics	63
12	San Francisco	Grocery	74

Query 6 counts the number of stores with staff efficiency above average by store type and location. The query filters the data only to include stores where the staff efficiency is above the average, as indicated by the staff_effic_aboveavg field being equal to 1. It then groups the results by location and store type, providing a count of stores in each group where staff efficiency is above the average. The results are ordered by location and store type.

Visualization:Store Counts Above Average Staff Efficiency by Location and Type

```
## column chart, clustered by location and filled by store type #
ggplot(count_above_each, aes(x = Location, y = stores_above_avg, fill = Store_Type)) +
  geom_col(position = position_dodge(width = 0.8), color = "black") +
  scale_fill_manual(
    values = c("Grocery" = "lightgreen", "Clothing" = "pink", "Electronics" = "lightblue")) +
  geom_text( aes(label = stores_above_avg),position = position_dodge(width = 0.8),
             vjust = -0.5, size = 4) +
  labs(
    title = "Store Counts Above Average Staff Efficiency by Location",
    x = "Location",
    y = "Stores Above Average",
    fill = "Store Type") +
  theme_classic() +
  theme(
    plot.title = element_text(hjust = 0.5,  size = 16),
    legend.position = "top")
```



This analysis highlights the number of stores performing above the average staff efficiency, categorized by store type and region. Los Angeles and Sacramento show strong performance across all store types, with Electronics and Clothing stores leading in both cities. In San Francisco, Grocery stores outperform others, with an impressive 74 stores exceeding the average, while Electronics and Clothing stores lag behind. These trends suggest that regional factors such as customer demand, operational practices, and local conditions influence staff efficiency. Overall, the analysis offers insights into how location and store type impact staff performance.

Query 7: Analyzing Amount Spent Per Customer Based on Local Economic Indicator Score

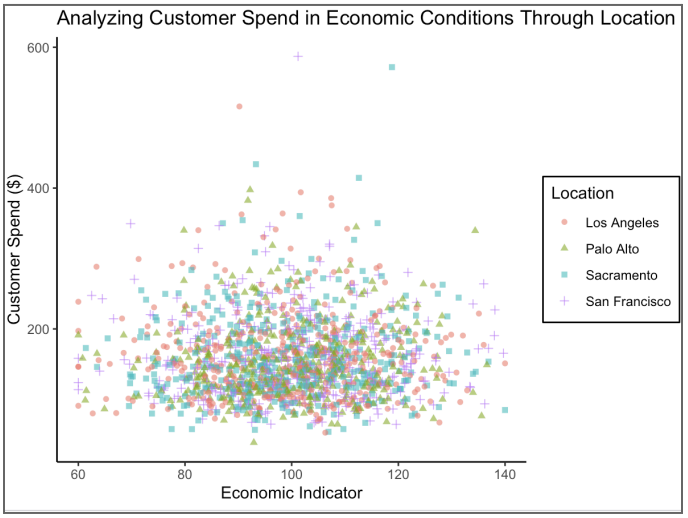
```
query7 <-"SELECT Location, Economic_Indicator,
          Monthly_Sales/Traffic AS Traffic_Sales
          FROM final_stores_data
          ORDER BY Traffic_Sales DESC"
econ_spend<- sqldf(query7)
```

This query calculates the average spending per customer by dividing monthly sales by monthly traffic, resulting in a metric known as Traffic_Sales. By categorizing the data by location and economic indicator, we can better understand regional variations in customer spending behavior. Sorting the results in descending order of Traffic_Sales shows which locations are most effective at converting foot traffic into sales, providing insights into customer purchasing habits across different regions. Higher Traffic_Sales indicate areas where stores are better at generating revenue per customer visit, which could reflect more effective sales strategies, higher-priced products, or better customer engagement. This analysis allows businesses to identify regions with

strong conversion rates and tailor marketing or operational strategies to replicate successful patterns in other areas.

Visualization: Customer Expenditure vs. Economic Indicator Score

```
# Generate a scatter plot to explore
ggplot(econ_spend, aes(x = Economic_Indicator, y = Traffic_Sales,
                       shape = Location, color = Location)) +
  geom_point(alpha = 0.6) +
  labs(
    title = "Analyzing Customer Spend in Economic Conditions Through Location",
    x = "Economic Indicator",
    y =  "Customer Spend ($)") +
  theme_classic() +
  theme(
    legend.background = element_rect(size = 0.5, color = "black" ))
```



This scatter plot illustrates the relationship between customer spending and economic conditions across various locations. The clustering of data points in the lower spending range ($150–$300), despite fluctuations in the economic indicator, suggests that average customer spending tends to remain stable regardless of economic cycles, whether during periods of growth or downturn. While a few outliers show instances of higher spending, these are rare and not representative of overall trends. This pattern reflects the resilience of consumer behavior, indicating that retailers can focus on maintaining consistent operations and efficient resource allocation without overreacting to short-term economic changes. The stability in spending provides confidence for businesses to plan long-term strategies, emphasizing the importance of operational continuity and adaptability in the face of economic shifts.

Query 8: Averaging Monthly Sales by Store Age, Sorted by Store Type

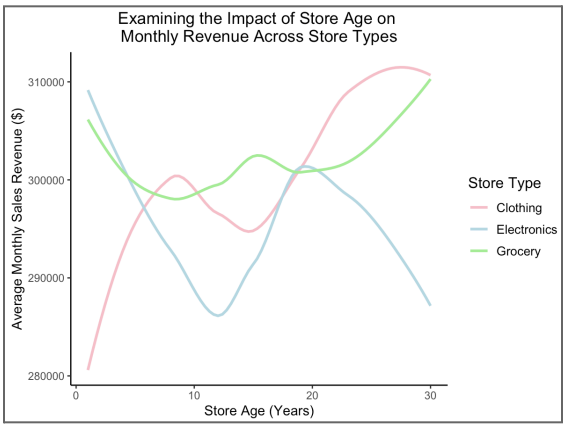
```
query8 <- "SELECT Store_Age, Store_Type, AVG(Monthly_Sales) AS Average_Monthly_Sales
           FROM final_stores_data
           GROUP BY Store_Age, Store_Type
           ORDER BY Store_Age, Store_Type;"
query8df <- sqldf(query8)
head(query8df)
```

	Store_Age	Store_Type	Average_Monthly_Sales
1	1	Clothing	271833.2
2	1	Electronics	306264.6
3	1	Grocery	307972.7
4	2	Clothing	297633.5
5	2	Electronics	299621.2
6	2	Grocery	296772.9

This query analysis aims to identify whether older stores tend to generate higher or lower sales compared to newer ones and how this trend varies across store types. A common assumption tied to this relationship is that as a store's age increases, its monthly revenue would also grow. This hypothesis is based on the idea that as businesses gain more experience in their industry, they are likely to refine their operations, optimize strategies, and build customer loyalty, all of which contribute to increased sales and improved revenue over time. Please note the console output is far greater than pictured above.

Visualization:Impact of Store Age on Monthly Revenues Across Store Types

```r
#Generate smooth line graph showing effect of a store's age on their average monthly sales
ggplot(query8df, aes(x = Store_Age, y = Average_Monthly_Sales, color = Store_Type)) +
  geom_smooth(method = "loess", se = FALSE) +
  labs(
    title = "Examining the Impact of Store Age on \nMonthly Revenue Across Store Types",
    x = "Store Age (Years)",
    y = "Average Monthly Sales Revenue ($)",
    color = "Store Type") +
  scale_color_manual(values = c("Grocery" = "lightgreen",
                                "Electronics" = "lightblue",
                                "Clothing" = "pink")) +
  theme_classic() +
  theme(
    plot.title = element_text(hjust = 0.5),
    legend.title = element_text(size = 12),
    legend.text = element_text(size = 10))
```



This graph demonstrates that the relationship between store age and average monthly revenue is far from straightforward. While it's often assumed that older stores generate higher revenue due to operational improvements and established customer bases, the data suggests that other external factors play a significant role (i.e. economic indicators, market trends, location dynamics, etc.). As shown above, we opted for a LOESS geom_smoother to give the visualization greater depth rather than simply fitting a line of best fit.

Extensions of Query 8: Averaging Monthly Sales by Store Age, Sorted by Store Type (Broken out by Store Type) Queries & Visuals

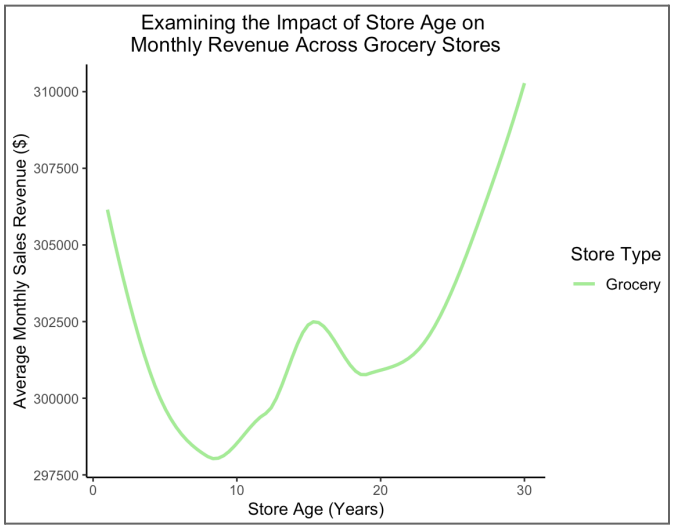
Visualization:Impact of Store Age on Monthly Revenues Across Clothing Stores

```r
query8_clothingvar <- "SELECT Store_Age, Store_Type,
AVG(Monthly_Sales) AS Average_Monthly_Sales
             FROM final_stores_data
             WHERE Store_Type = 'Clothing'
             GROUP BY Store_Age, Store_Type
             ORDER BY Store_Age, Store_Type"
query8_clothinggraph <- sqldf(query8_clothingvar)
```

To visualize each store type independently, we added a WHERE clause to the original query and replotted the data. The only alterations we made to the visualization code were changing the title and removing the manual color scales for the other store types. While there is a slight dip in stores aging between 10-20 years, the visualization above generally supports the conclusion that older apparel storefronts tend to generate higher monthly revenue, compared to their younger counterparts. Although we cannot account for the various external factors influencing this result, we can infer that workflow enhancements and stronger brand loyalty may play a role.

Visualization:Impact of Store Age on Monthly Revenues Across Grocery Stores

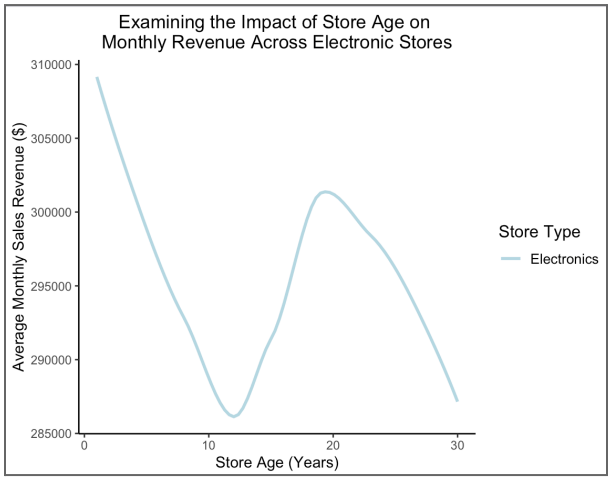
```
query8_groceryvar <- "SELECT Store_Age, Store_Type,
AVG(Monthly_Sales) AS Average_Monthly_Sales
            FROM final_stores_data
            WHERE Store_Type = 'Grocery'
            GROUP BY Store_Age, Store_Type
            ORDER BY Store_Age, Store_Type"
query8_grocerygraph <- sqldf(query8_groceryvar)
```



As shown, we applied the same code and query changes to generate the grocery store version of this graph. While we can seemingly draw a similar overarching conclusion about store age and revenue, there are some notable differences. For instance, newly opened grocery stores report significantly higher monthly revenue compared to clothing stores. One possible explanation is that clothing store launches tend to thrive through e-commerce, whereas groceries continue to dominate the in-person shopping space. Lastly, we observed a spike in monthly revenue for grocery stores aged 10-20 years, which contrasts with the trend seen in clothing stores. We hypothesize that this may be influenced by an external economic factor.

Visualization:Impact of Store Age on Monthly Revenues Across Electronic Stores

```
query8_electronicvar <- "SELECT Store_Age, Store_Type,
AVG(Monthly_Sales) AS Average_Monthly_Sales
            FROM final_stores_data
            WHERE Store_Type = 'Electronics'
            GROUP BY Store_Age, Store_Type
            ORDER BY Store_Age, Store_Type"
query8_electronicgraph <- sqldf(query8_electronicvar)
```



For this visualization, our commentary will primarily focus on the start and end points. Generally speaking, electronic stores tend to hit the ground running in terms of monthly sales, likely driven

by in-person promotions, grand openings, and other marketing efforts centered around the launch of new stores. However, by the time stores reach the age of 20, we observe a slight decline, which may be attributed to a loss of relevance in the rapidly evolving technology sector (e.g., Redbox, Blockbuster, RadioShack).

Query 9: How Store Size is Factored into the Relationship Between Product Variability & Sales

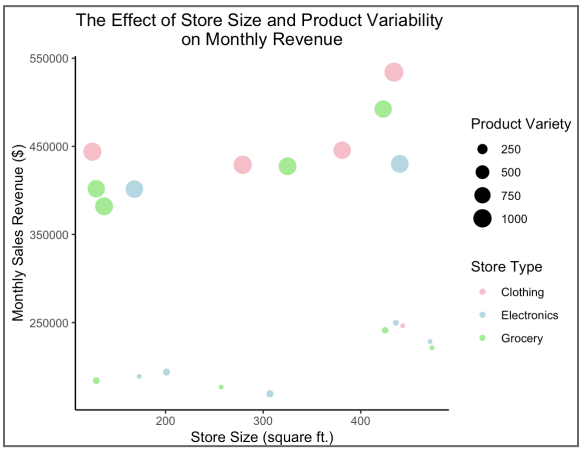
```
#Query data for the selected fields to be arranged for the top 10 highest product varieties
query9top10 <- "SELECT Store_Size, Monthly_Sales, Product_Variety, Store_Type
            FROM final_stores_data
            ORDER BY Product_Variety DESC
            LIMIT 10"
graphtop10 <- sqldf(query9top10)
#Query data for the selected fields to be arranged for the top 10 lowest product varieties
query9bottom10 <- "SELECT Store_Size, Monthly_Sales, Product_Variety, Store_Type
            FROM final_stores_data
            ORDER BY Product_Variety
            LIMIT 10"
graphbottom10 <- sqldf(query9bottom10)
query9_combined_data <- rbind(graphtop10, graphbottom10)
```

	Store_Size	Monthly_Sales	Product_Variety	Store_Type
1	434	534260	1092	Clothing
2	279	429020	998	Clothing
3	125	443950	970	Clothing
4	137	381990	943	Grocery
5	325	427450	931	Grocery
6	440	430080	916	Electronics
7	381	445620	890	Clothing
8	168	401490	889	Electronics
9	423	492380	879	Grocery
10	129	401920	878	Grocery
11	257	176830	100	Grocery
12	173	188830	100	Electronics
13	443	246430	100	Clothing
14	473	221470	100	Grocery
15	471	228590	100	Electronics
16	436	249840	105	Electronics
17	425	241240	116	Grocery
18	129	184140	120	Grocery
19	201	193790	123	Electronics
20	307	169110	128	Electronics

The objective of this code and query is to explore the relationship between product variety, revenue, and store size by analyzing the top 10 and bottom 10 observations ranked by product variety. We specifically chose these extreme values (top 10 and bottom 10) from a visualization and coding perspective; plotting all the data points would have resulted in intense overlapping, making it difficult to identify meaningful patterns and relationships. By focusing on these subsets, we created a clearer and more insightful visualization. Additionally, this approach encouraged us to try the 'rbind' function, which combined outputs from two separate queries.

Visualization: The Effect of Store Size & Product Variety on Monthly Revenues by Store Type

```
# Plot points to show magnitude of product variety in relation to sales
#Plot points to position of store size in relation to monthly sales
ggplot(query9_combined_data, aes(x = Store_Size, y = Monthly_Sales, size = Product_Variety,
                                 colour = Store_Type)) +
  geom_point(alpha = 1) +
  scale_size_continuous(name = "Product Variety") +
  scale_color_manual(values = c("Grocery" = "lightgreen",
                                "Electronics" = "lightblue",
                                "Clothing" = "pink")) +
  labs(title = "The Effect of Store Size and Product Variability \non Monthly Revenue",
       x = "Store Size (square ft.)",
       y = "Monthly Sales Revenue ($)",
       color = "Store Type") +
  theme_classic() +
  theme(
    plot.title = element_text(hjust = 0.5),
    legend.position = "right")
```

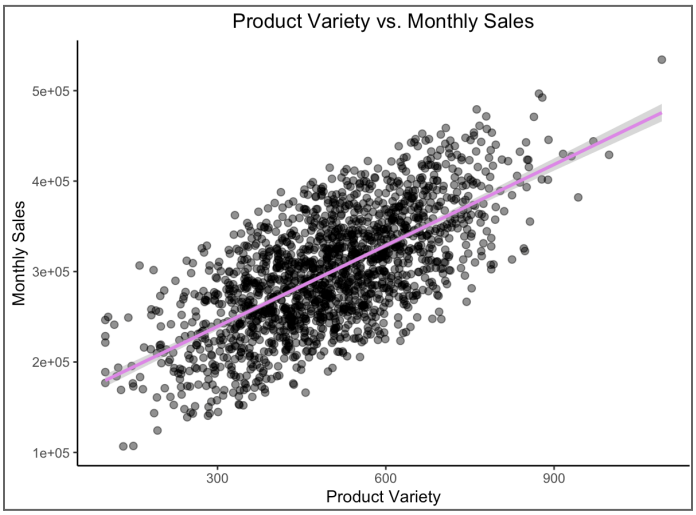


Since we are utilizing extreme values in this visualization, we should avoid placing too much weight on the conclusions derived from it. With that being said, this graph highlights that retail

stores in the upper quadrant, regardless of type, consistently have high product variety (greater than 500), which correlates with higher revenue. For grocery and electronics stores, lower product variety is associated with smaller store sizes and lower revenue, emphasizing the importance of variety in driving performance. There are notably some outliers with large store sizes but lower product variety and revenue; these may represent outlet spaces or specialized formats, deviating from the general trend.

Visualization: Linear Relationship Between Product Variety & Monthly Sales

```
# Scatter plot with trend line exploring how a store's product variety
# explains variation in monthly sales
ggplot(final_stores_data, aes(x = Product_Variety, y = Monthly_Sales)) +
  geom_point(size = 2, color = "black", alpha = 0.5) +
  geom_smooth(method = "lm", se = TRUE, color = "violet") +
  labs(title = "Product Variety vs. Monthly Sales",
       x = "Product Variety",
       y = "Monthly Sales") +
  theme_classic() +
  theme(plot.title = element_text(hjust = 0.5))
```




```
lm(formula = Monthly_Sales ~ Product_Variety, data = final_stores_data)

Residuals:
    Min      1Q  Median      3Q     Max
-120007  -35165   -1606   34856  115850

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     1.498e+05  4.207e+03   35.60   <2e-16 ***
Product_Variety 2.984e+02  8.055e+00   37.05   <2e-16 ***
```

Using the entire dataset, we created a scatter plot to analyze the relationship between product variety and monthly sales. By adding a regression line to the plot, we can clearly observe a positive correlation, indicating that stores offering a greater variety of products tend to generate higher sales.

Our analysis reveals a strong positive relationship between product variety and revenue, indicating that increasing the number of products available significantly boosts sales. The slope of the regression line is 298.4, meaning that for every 1-unit increase in product variety, we can predict a $298.40 increase in sales. This insight is particularly valuable for inventory planning, as it helps businesses determine the optimal number of products to stock to achieve future sales targets. By strategically expanding product offerings, stores can better meet customer demands, enhance traffic, and drive higher revenue, ultimately improving overall performance.

Query 10: Querying Data table for Product Variety and Monthly Sales Fields for Visual Analysis

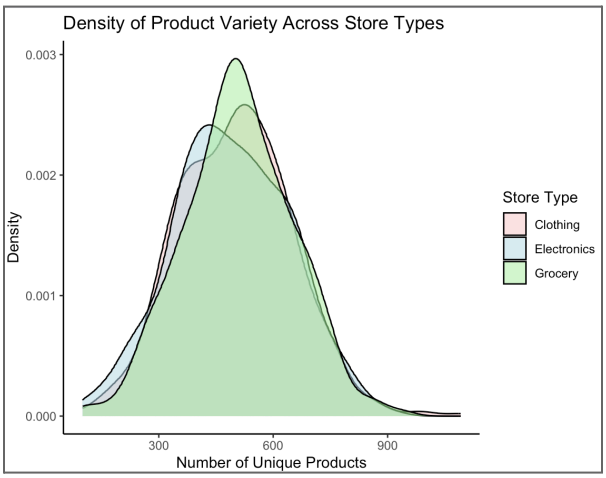
```
# Query data table for product variety & store type
Q10prodvar_st<- "SELECT product_variety, Store_Type
                 FROM final_stores_data"
prodvar_st<- sqldf(Q10prodvar_st)
```

In this query, SQL was used to pull the two fields, “Product Variety” and “Store Type”, within the dataframe to further explore how they are related. Creating a separate relational data table for filtering fields was helpful, as it could specifically be referred to for future use when further

analysis was conducted on the relationship between those two fields. The primary purpose of conducting this analysis would be to compare it to the distribution of monthly sales by store type, since product variety and monthly sales are highly correlated, we wanted to see how visually similar the distributions would be.

Visualization: Density curve of Product Variety by Store Type

```
# use df from query to map variables
plot_variet<-ggplot(prodvar_st,aes(x=Product_Variety, fill = Store_Type)) +
  theme_classic()
#Add density plot geom layers to aesthetic mapping, use position arguments for clarity
prodvar_storetype<-plot_variet +
  geom_density(aes(y=..density..), position = "jitter", alpha = 0.5) +
  scale_fill_manual(values = c("Grocery" = "lightgreen", "Electronics" = "lightblue",
                               "Clothing" = "pink")) +
  labs(title = "Density of Product Variety Across Store Types",
       x = "Number of Unique Products", fill = "Store Type", y = "Density")
```




```
        Jarque Bera Test

data:  mean_grocery$Product_Variety
X-squared = 0.067219, df = 2, p-value = 0.9669
```

For this ggplot, the alpha and jitter arguments were used within the geometry layer to improve readability of the plot, without them it would be difficult to inspect the different shapes of the curves for each store type, thereby defeating its purpose. The density curve for the number of unique products for electronic stores displays a positive skew indicating that the mean is being pulled up higher than the median by a handful of stores. We can think of it as the median being mostly defined by smaller boutique electronic stores such as the Apple Store or the stores of cell carriers, whose product assortment is limited to their brand or the few brands they carry. If we were to think of the small group of stores pulling the mean up, we could imagine all of those stores being only different Best Buys who carry all sorts of electronics from many brands and for many different utilities. That alone would cause the mean number of unique products to be pulled upwards. The density plot for clothing stores is slightly bimodal, pointing to the possibility that two types of clothing stores with strictly different product varieties were sampled resulting in two modes. An example being boutique stores that carry only their own brand versus department stores carrying many different brands, being sampled. Grocery stores have a normal distribution (confirmed by a failure to reject the null hypothesis of the jarque bera test that the distribution is normal). This means that the distribution of product variety among grocery stores is centered around the mean of 510 unique products, in other words you have a 50% chance of entering a grocery store with >= 510 unique products <= from this sample.

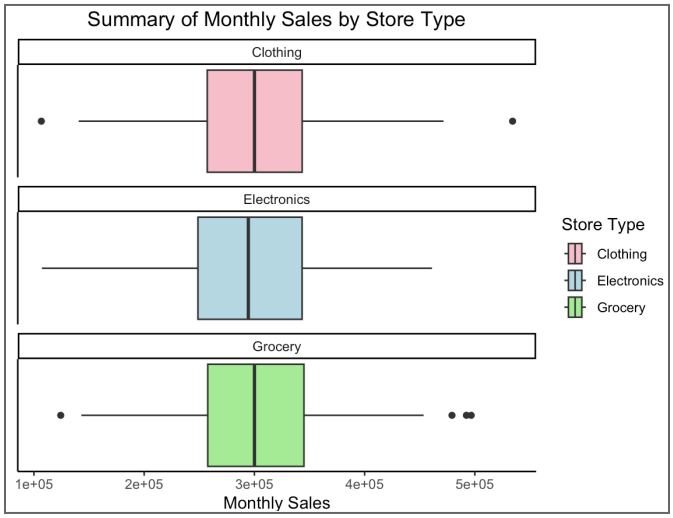
Query 11: Querying Data table for Monthly Sales and Store Type Fields for Visual Analysis

```
# Query data table for monthly sales & store type fields
Q11sales_st<- "SELECT Store_Type, Monthly_sales
               FROM final_stores_data"
sales_st<- sqldf(Q11sales_st)
```

Similar to Query 10, SQL was used in this query to select the fields, “Monthly_Sales” and “Store_Type” within the dataframe to view their relationship visually and serve as a reference for future analysis into the effect Store Type has on Monthly Sales.

Visualization: Box Plot Distribution of Monthly Sales Facet Wrapped by Store Type

```
### Generate Box plot of monthly sales distribution by store type ###
bxplt_sales_by_storetype<- ggplot(sales_st, aes(Monthly_Sales, fill = Store_Type)) +
  geom_boxplot() +
  facet_wrap(~Store_Type, ncol=1) +
  scale_fill_manual(values = c("Grocery" = "lightgreen", "Electronics" = "lightblue",
                               "Clothing" = "pink"), name = "Store Type") +
  labs(title = "Summary of Monthly Sales by Store Type", x = "Monthly Sales") +
  theme_classic() +  theme(axis.text.y=element_blank(), axis.ticks.y=element_blank(),
        plot.title = element_text(hjust = 0.5))
bxplt_sales_by_storetype
```



In this particular ggplot, the ncol argument was set to 1 so the distributions were plotted along the same axis they were being comparatively measured on for easier interpretation of the results. Again, since product variety & monthly sales are highly correlated we wanted to explore how this representation parallels that of the density plots for product variety by store type. Upon a very close look, we can see that the positive skew for electronics stores is barely reflected within their monthly sales distribution showing the positive correlation between monthly sales and product variety we expected. The reason why the difference in sales distribution across different store types is not as drastic as that of the density plots is because there are many variables whose effects have an influence over monthly sales effectively diluting the effect product variety has on sales on its own. Outliers within grocery store sales could be explained by either end of the grocery affordability spectrum. On the low end it could be that a “hard discounted” food store, such as Food 4 Less or on the high end a few luxury grocery stores, such as Erewhon, were sampled. The same kind of juxtaposition can be used to explain the outliers among clothing store sales with brands like H&M on one end of the spectrum with brands like Chanel on the other.

Query 12: Querying Joined Data Table for Analysis on Ratio Variable & Adding a Column Using “Cause”

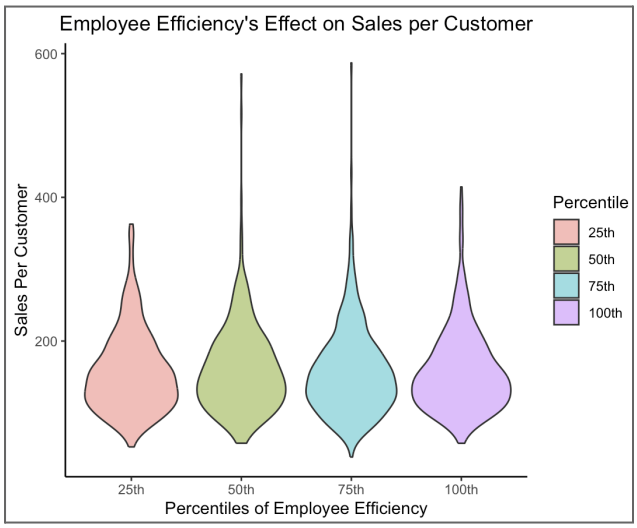
```
# Query data table for Sales Per Customer and Employee Efficiency ###
Q12spc_ee<- "SELECT Sales_Per_Customer, Staff_Efficiency
                  FROM fsd_joined"
spc_eedf<- sqldf(Q12spc_ee)
# Add a column where the staff efficiency scores are labeled by quartile number ###
spc_eedf$Quartile <- with(spc_eedf, cut(Staff_Efficiency,
                       breaks=quantile(Staff_Efficiency, probs=seq(0,1, by=0.25),
                       na.rm=TRUE), include.lowest=TRUE))
head(spc_eedf,10)
#Query to create column labeling percentiles
Q12.2<- "SELECT *,
          CASE
WHEN Quartile = '(83.1,94.9]' THEN '100th'
WHEN Quartile = '(72.1,83.1]' THEN '75th'
WHEN Quartile = '(61,72.1]' THEN '50th'
WHEN Quartile = '[50,61]' THEN '25th'
END AS Percentile
FROM spc_eedf"
ee_percentiles<- sqldf(Q12.2)
```

Sales_Per_Customer	Staff_Efficiency	Quartile	Percentile
165.35113	84.9	(83.1,94.9]	100th
253.04598	75.8	(72.1,83.1]	75th
110.06405	92.8	(83.1,94.9]	100th
107.91586	66.3	(61,72.1]	50th
167.22920	89.1	(83.1,94.9]	100th
144.33203	88.8	(83.1,94.9]	100th
128.64026	65.2	(61,72.1]	50th
318.42239	54.6	[50,61]	25th

Similar to queries 10 and 11 this SQL query was used to filter the fields “Sales Per Customer” and “Staff Efficiency” to define the relationship between these two variables. Querying the joined data table was helpful as it kept our code organized as it distinguished that we were dealing with a mutated field that was calculated and was not a part of the original dataframe. Then a column was added using the cut function, where staff efficiency was broken into quartiles and each record was assigned to the interval of their quartile. This set the “bins” for the levels all of the records would be distributed in. Finally, SQL was used to create a column that labeled which percentile each record of Staff Efficiency belonged to based on the quartile they were in.

Visualization: Violin Plots of Sales Per Customer Facet Wrapped by Percentiles of Staff Efficiency

```
# Plot violin plots for each quartile
Emp_eff_SPC<- ggplot(ee_percentiles, aes(as.factor(Quartile), Sales_Per_Customer, fill = Quartile)) +
  geom_violin(alpha = 0.5)+
  labs(title = "Employee Efficiency's Effect on Sales per Customer",
       x = "Percentiles of Employee Efficiency",
       y = "Sales Per Customer", fill = "Percentile") +
  scale_x_discrete(labels = c("25th", "50th", "75th", "100th"))+
  scale_fill_discrete(labels = c("25th", "50th", "75th", "100th"))+
  theme_classic() +
  theme(plot.title = element_text(hjust = 0.5))
Emp_eff_SPC
```



Considering the fact that we were not given more information inside of the codebook on how employee performance was measured, we figured a good place to start was measuring the distribution of sales per customer for each percentile. As previously mentioned, the ratio of sales to customer traffic, known as “Sales Per customer” in our joined data table, can measure how well employees (specifically sales associates) are able to sell the products and increase average order value. We hypothesized that as employee efficiency scores increased, so would sales per customer based on the definition described. It can be seen that the 25th percentile of employee efficiency scores had less than favorable sales per customer outcomes. An example scenario would be that an inefficient employee has low engagement with customers or low product knowledge resulting in an inability to drive up sales per customer. The 50th and 75th percentiles had the most instances of high sales per customer values. This aligned with our hypothesis that as scores increased so would the performance ratio they are being measured against. Finally, employees in the 100th percentile still had high sales per customer but not nearly as high as the interquartile range. This may be due to the work their efficiency was evaluated upon was less customer centered, thereby causing them to be viewed as less effective against this metric.

Query 13: Analyzing Ad Spend and Revenues across Store Types and Cities

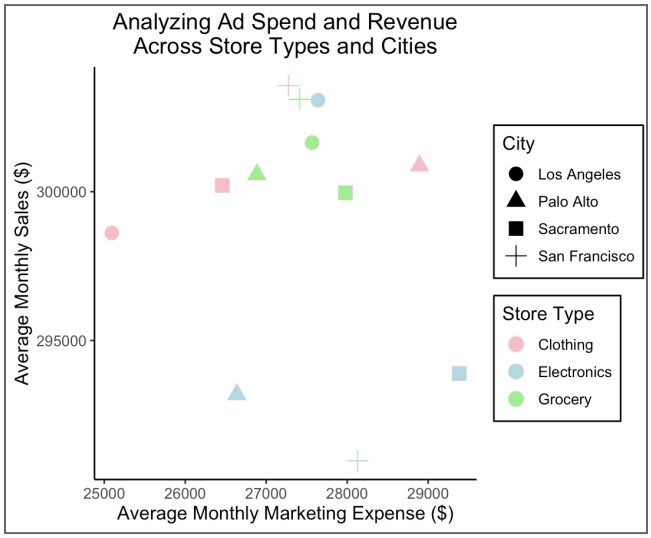
```
query13 <-"SELECT Location, Store_Type, AVG(Marketing_Expense) AS AVG_Marketing_Expense,
          AVG(Monthly_Sales) AS AVG_Monthly_Sales
          FROM final_stores_data
          GROUP BY Location, Store_type
          ORDER BY Location, Store_type"
sqldf(query13)
query13df <- sqldf(query13)
```

```
> head(query13df)
     Location  Store_Type AVG_Marketing_Expense AVG_Monthly_Sales
1 Los Angeles    Clothing              25093.75          298610.2
2 Los Angeles Electronics              27640.29          303073.4
3 Los Angeles     Grocery              27566.67          301649.9
4   Palo Alto    Clothing              28893.13          300874.0
5   Palo Alto Electronics              26637.80          293184.2
6   Palo Alto     Grocery              26885.91          300578.1
```

The purpose of this query was to evaluate the relationship between marketing expenses and monthly sales across different store types and locations. We essentially wanted to explore whether specific locations exhibited different responses to higher or lower marketing spending. For example, our initial hypothesis was that Los Angeles would have one of the highest marketing spends for clothing, which we expected to correlate with high revenue in that category.

Visualization: Shape Plot of Relationship between Monthly Revenues and Marketing Expense by Location and Store Type

```
ggplot(query13df, aes(x = AVG_Marketing_Expense,y = AVG_Monthly_Sales,color = Store_Type,
  shape = Location)) + geom_point(size = 4) +
  labs(title = "Analyzing Ad Spend and Revenue\nAcross Store Types and Cities",
       x = "Average Monthly Marketing Expense ($)",
       y = "Average Monthly Sales ($)",
       color = "Store Type",
       shape = "City") +
  scale_color_manual(values = c("Grocery" = "lightgreen", "Electronics" = "lightblue",
                                "Clothing" = "pink"), name = "Store Type") +
  theme_classic() +
  theme( text = element_text(size = 12), legend.position = "right",
    plot.title = element_text(hjust = 0.5),
    legend.background = element_rect(color = "black", size = 0.5))
```



The only significant distinction within this visualization code was our decision to outline the legend and utilize the shape aesthetic; with fewer congregated scatter points, we felt that a clearer legend would improve the graph's overall readability. Now turning to our graphical interpretations– despite fluctuations in ad spend, electronics stores consistently report lower monthly revenue, with Los Angeles being an exception. Grocery stores, on the other hand, demonstrate a balanced approach, maintaining middle-line marketing expenses while consistently achieving higher revenues compared to their counterparts. Los Angeles and San Francisco emerge as the cities with the highest revenue numbers across most store types, highlighting their strong market performance.

CONCLUSION

In conclusion, the analysis of performance trends across store types reveals that grocery stores exhibit higher foot traffic and sustained revenue growth over time, highlighting their ability to build long-term customer loyalty. On the other hand, clothing and electronics stores show more variability in performance, suggesting they are more sensitive to factors such as marketing investment and store size. Comparing performance across store types can be challenging due to management differences, but metrics like sales per customer and cost per customer provide a simplified approach for cross-store comparison. Despite the variations in store types, overall performance remains strong, with consistent staff efficiency and stable customer spending across

locations and economic conditions. These positive internal factors indicate significant growth potential, suggesting that stores are well-positioned for continued improvement in monthly revenue moving forward. Overall, our analysis confirmed our initial predictions, demonstrating the resilience and growth potential within the current operational framework.