

INTRODUCTION

This report presents a comprehensive analysis of a car repairs shop's operations, focusing on the distribution of vehicle ages, trends in service requirements, and the identification of jobs with the highest and lowest average costs. The dataset comprises detailed records of vehicle information, parts used, and job descriptions, providing valuable insights into the operational dynamics of the service center. By analyzing data from [time period, e.g., the past twelve months], the report aims to uncover actionable insights that will drive operational efficiency, enhance customer satisfaction, and boost the company's bottom line. This analysis will serve as a roadmap for targeted marketing campaigns, optimized service offerings, and improved resource allocation."

Using data from the invoicing system, customer database, and job records, the analysis identifies top-spending customers, evaluates the frequency and revenue of various job types, and assesses the overall profitability of the services. The findings are supported by visualizations and actionable recommendations to enhance service delivery, improve customer satisfaction, and boost profitability.

METHODOLOGY

The analysis involved the following steps:

DATA PREPARATION:

The vehicle, parts, invoice, customer and job data were loaded into a relational database. The data was cleaned, properly formatted, and indexed for efficient querying and analysis.

SQL scripts used to create tables and import data for customers

```
3 CREATE TABLE customer (  
4     CustomerID INT PRIMARY KEY AUTO_INCREMENT,  
5     Name VARCHAR(100),  
6     Address VARCHAR(255),  
7     Phone VARCHAR(20)  
8 );  
9  
10 LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/customer.csv"  
11 INTO TABLE customer  
12 FIELDS TERMINATED BY ','  
13 ENCLOSED BY ''''  
14 LINES TERMINATED BY '\n'  
15 IGNORE 1 ROWS  
16 (Name, Address, Phone);  
17
```

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	CustomerID	Name	Address	Phone
1		Jennifer Robinson	126 Nairn Ave, Winnipeg, MB, R3J 3C4	204-771-0784
2		Michael Smith	250 Broadway, Winnipeg, MB, R3C 0R5	204-555-1234
3		Sarah Johnson	789 Main St, Winnipeg, MB, R2W 3N2	204-666-5678
4		Emily Brown	456 Elm St, Winnipeg, MB, R3M 2S5	204-777-9101
5		David Wilson	123 Oak St, Winnipeg, MB, R2J 3C4	204-888-1112

SQL scripts used to create tables and import data for vehicle

- ```
CREATE TABLE vehicle (
 VehicleID INT PRIMARY KEY AUTO_INCREMENT,
 Make VARCHAR(50),
 Model VARCHAR(50),
 Year INT,
 Color VARCHAR(50),
 VIN VARCHAR(25),
 Reg_num VARCHAR(20),
 Mileage INT,
 OwnerName VARCHAR(100)
);
```
- ```
LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/vehicle.csv.csv"  
INTO TABLE vehicle  
FIELDS TERMINATED BY ','  
ENCLOSED BY ''  
LINES TERMINATED BY '\n'  
IGNORE 1 ROWS  
(Make, Model, Year, Color, VIN, Reg_num, Mileage, OwnerName);
```

Result Grid Filter Rows: <input type="text"/> Edit: Export/Import: Wrap Cell Content:									
	VehideID	Make	Model	Year	Color	VIN	Reg_num	Mileage	OwnerName
▶	1	BMW	X5	2012	Black	CVS123456789123-115Z	BMW 123	16495	Jennifer Robinson
	2	Toyota	Corolla	2015	White	TYS678901234567-876Z	TOY 456	45000	Michael Smith
	3	Honda	Civic	2018	Blue	HCS345678901234-123X	HON 789	30000	Sarah Johnson
	4	Ford	Escape	2020	Red	FES234567890123-456Y	FOR 987	15000	Emily Brown
	5	Chevrolet	Malibu	2016	Silver	CMS456789012345-789Z	CHE 321	60000	David Wilson

SQL scripts used to create tables and import data for invoice

- ```
CREATE TABLE invoice (
 InvoiceID INT PRIMARY KEY,
 InvoiceDate DATE,
 SubtotalParts DECIMAL(10,2),
 SubtotalLabour DECIMAL(10,2),
 SalesTaxRate DECIMAL(10,2),
 SalesTax DECIMAL(10,5),
 TotalLabour DECIMAL(10,2),
 TotalParts DECIMAL(10,2),
 Total DECIMAL(10,5),
 CustomerID INT,
 VehicleID INT,
 FOREIGN KEY (CustomerID) REFERENCES customer(CustomerID),
 FOREIGN KEY (VehicleID) REFERENCES vehicle(VehicleID)
);
```
- ```
LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/invoices.csv.csv"  
    INTO TABLE invoice  
    FIELDS TERMINATED BY ','  
    ENCLOSED BY ''''  
    LINES TERMINATED BY '\n'  
    IGNORE 1 ROWS;
```

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	InvoiceID	InvoiceDate	SubtotalParts	SubtotalLabour	SalesTaxRate	SalesTax	TotalLabour	TotalParts	Total	CustomerID	VehicleID
▶	12345	2023-09-10	969.87	625.00	13.00	207.33310	625.00	969.87	1802.20310	1	1
	12346	2023-09-15	200.00	325.00	13.00	68.25000	325.00	200.00	593.25000	2	2
	12347	2023-09-20	150.00	200.00	13.00	45.50000	200.00	150.00	395.50000	3	3
	12348	2023-09-25	125.00	325.00	13.00	58.50000	325.00	125.00	508.50000	4	4
	12349	2023-09-30	140.00	440.00	13.00	75.40000	440.00	140.00	655.40000	5	5

SQL scripts used to create tables and import data for job

- ```
CREATE TABLE job (
 JobID INT PRIMARY KEY,
 VehicleID INT,
 Description VARCHAR(255),
 Hours DECIMAL(5,2),
 Rate DECIMAL(10,2),
 Amount DECIMAL(10, 2),
 InvoiceID INT,
 FOREIGN KEY (VehicleID) REFERENCES vehicle(VehicleID),
 FOREIGN KEY (InvoiceID) REFERENCES invoice(InvoiceID)
);
```
- ```
LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/job.csv"  
INTO TABLE job  
FIELDS TERMINATED BY ','  
ENCLOSED BY '"'  
LINES TERMINATED BY '\n'  
IGNORE 1 ROWS;
```

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	JobID	VehicleID	Description	Hours	Rate	Amount	InvoiceID
	1	1	Diagnose front wheel vibration	0.50	125.00	62.50	12345
	2	1	Replace front CV Axel	3.50	125.00	437.50	12345
	3	1	Balance tires	1.00	125.00	125.00	12345
	4	2	Oil change	1.00	75.00	75.00	12346
	5	2	Replace brake pads	2.00	125.00	250.00	12346
	6	3	Replace battery	1.50	100.00	150.00	12347
	7	3	Tire rotation	1.00	50.00	50.00	12347
	8	4	Transmission check	2.00	150.00	300.00	12348
	9	4	Replace air filter	0.50	50.00	25.00	12348
	10	5	Coolant flush	1.50	120.00	180.00	12349
	11	5	Replace spark plugs	2.00	130.00	260.00	12349

SQL scripts used to create tables and import data for parts

```
CREATE TABLE parts (  
    PartID INT,  
    JobID INT,  
    Part_num VARCHAR(50),  
    PartName VARCHAR(100),  
    Quantity INT,  
    UnitPrice DECIMAL(10, 2),  
    Amount DECIMAL(10, 2),  
    InvoiceID INT,  
    FOREIGN KEY (InvoiceID) REFERENCES invoice(InvoiceID),  
    FOREIGN KEY (JobID) REFERENCES job(JobID)  
);  
  
LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/parts.csv.csv"  
INTO TABLE parts  
FIELDS TERMINATED BY ','  
ENCLOSED BY ''  
LINES TERMINATED BY '\n'  
IGNORE 1 ROWS;
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:




	PartID	JobID	Part_num	PartName	Quantity	UnitPrice	Amount	InvoiceID
▶	1	2	23435	CV Axel	1	876.87	876.87	12345
	2	2	7777	Shop Materials	1	45.00	45.00	12345
	3	3	W187	Wheel Weights	4	12.00	48.00	12345
	4	5	54321	Brake Pads	1	200.00	200.00	12346
	5	6	67890	Battery	1	120.00	120.00	12347
	6	7	11223	Tire Rotation Kit	1	30.00	30.00	12347
	7	8	33445	Transmission Fluid	1	100.00	100.00	12348
	8	9	99887	Air Filter	1	25.00	25.00	12348
	9	10	77654	Coolant	1	60.00	60.00	12349
	10	11	99876	Spark Plugs	4	20.00	80.00	12349

ANALYSIS:

CUSTOMER ANALYSIS:



Identify the top 5 customers who have spent the most on vehicle repairs and parts.

```
116      -- Identify the top 5 customers who have spent the most on vehicle repairs and parts.
117 •    SELECT customer.CustomerID, customer.Name, SUM(invoice.Total) AS TotalSpent
118      FROM customer
119      INNER JOIN invoice
120      ON customer.CustomerID = invoice.CustomerID
121      GROUP BY customer.CustomerID, customer.Name
122      ORDER BY TotalSpent DESC
123      LIMIT 5;
```

Result Grid			
Filter Rows: <input type="text"/>			
Export:  Wrap Cell Content:  Fetch rows: 			
	CustomerID	Name	TotalSpent
▶	1	Jennifer Robinson	1802.20310
	5	David Wilson	655.40000
	2	Michael Smith	593.25000
	4	Emily Brown	508.50000
	3	Sarah Johnson	395.50000

Determine the average spending of customers on repairs and parts.

```
125      -- Determine the average spending of customers on repairs and parts.
126 •    SELECT AVG(invoice.Total) AS AverageSpending
127      FROM customer
128      INNER JOIN Invoice
129      ON customer.CustomerID = invoice.CustomerID;
130
```

Result Grid	
Filter Rows: <input type="text"/>	
Export:  Wrap Cell Content: 	
	AverageSpending
▶	790.970620000

Analyze the frequency of customer visits and identify any patterns.

```
134      -- Analyze the frequency of customer visits and identify any patterns.
135  •    SELECT customer.Name,
136      COUNT(invoice.InvoiceID) as VisitCount
137  FROM invoice
138  JOIN customer
139  ON invoice.CustomerID = customer.CustomerID
140  GROUP BY customer.Name
141  ORDER BY VisitCount DESC;
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	Name	VisitCount			
▶	Jennifer Robinson	1			
	Michael Smith	1			
	Sarah Johnson	1			
	Emily Brown	1			
	David Wilson	1			

VEHICLE ANALYSIS:

Calculate the average mileage of vehicles serviced

```
140      -- Calculate the average mileage of vehicles serviced.
141  •    SELECT AVG(Mileage) AS AverageMileage
142  FROM Vehicle;
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	AverageMileage				
▶	33299.0000				

Identify the most common vehicle makes and models brought in for service.

```
144 -- Identify the most common vehicle makes and models brought in for service.
145 • SELECT vehicle.Make, vehicle.Model, COUNT(*) AS Count
146 FROM vehicle
147 JOIN job
148 ON vehicle.VehicleID = job.VehicleID
149 GROUP BY vehicle.Make, vehicle.Model
150 ORDER BY Count DESC
151 LIMIT 5;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
Make	Model	Count		
BMW	X5	3		
Toyota	Corolla	2		
Honda	Civic	2		
Ford	Escape	2		
Chevrolet	Malibu	2		

Analyze the distribution of vehicle ages and identify any trends in service requirements based on vehicle age.

```
153 -- Analyze the distribution of vehicle ages and identify any trends
154 -- in service requirements based on vehicle age.
155 • SELECT Year, (2024 - Year) AS Age
156 FROM vehicle;
157
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
Year	Age		
2012	12		
2015	9		
2018	6		
2020	4		
2016	8		


```

158 • SELECT (2024 - Year) AS Age, COUNT(*) AS Frequency
159 FROM vehicle
160 GROUP BY Age
161 ORDER BY Age;

```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	Age	Frequency			
▶	4	1			
	6	1			
	8	1			
	9	1			
	12	1			

```

163 • SELECT (2024 - vehicle.Year) AS Age, job.Description, COUNT(*) AS Frequency
164 FROM vehicle
165 JOIN job
166 ON vehicle.VehicleID = job.VehicleID
167 GROUP BY Age, job.Description
168 ORDER BY Age, Frequency DESC;




```

Result Grid				Filter Rows:	Export:	Wrap Cell Content:
	Age	Description	Frequency			
▶	4	Transmission check	1			
	4	Replace air filter	1			
	6	Replace battery	1			
	6	Tire rotation	1			
	8	Coolant flush	1			
	8	Replace spark plugs	1			
	9	Oil change	1			
	9	Replace brake pads	1			
	12	Diagnose front wheel vibration	1			
	12	Replace front CV Axel	1			
	12	Balance tires	1			

JOB PERFORMANCE ANALYSIS




Determine the most common types of jobs performed and their frequency.

```
170      -- Determine the most common types of jobs performed and their frequency.
171 •    SELECT Description, COUNT(JobID) as Frequency
172      FROM job
173      GROUP BY Description
174      ORDER BY Frequency DESC;
```

Result Grid  Filter Rows: <input type="text"/> Export:  Wrap Cell Content: 		
	Description	Frequency
▶	Diagnose front wheel vibration	1
	Replace front CV Axel	1
	Balance tires	1
	Oil change	1
	Replace brake pads	1
	Replace battery	1
	Tire rotation	1
	Transmission check	1
	Replace air filter	1
	Coolant flush	1
	Replace spark plugs	1

Calculate the total revenue generated from each type of job.

```
176      -- Calculate the total revenue generated from each type of job.
177 •    SELECT Description, SUM(Amount) as TotalRevenue
178      FROM job
179      GROUP BY Description
180      ORDER BY TotalRevenue DESC;
```

Result Grid  Filter Rows: <input type="text"/> Export:  Wrap Cell Content: 		
	Description	TotalRevenue
▶	Replace front CV Axel	437.50
	Transmission check	300.00
	Replace spark plugs	260.00
	Replace brake pads	250.00
	Coolant flush	180.00
	Replace battery	150.00
	Balance tires	125.00
	Oil change	75.00
	Diagnose front wheel vibration	62.50
	Tire rotation	50.00
	Replace air filter	25.00

Identify the jobs with the highest and lowest average costs.

```
182 -- Identify the jobs with the highest and lowest average costs.
183 • SELECT Description, AVG(Amount) AS AverageCost
184 FROM job
185 GROUP BY Description
186 ORDER BY AverageCost DESC;
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	Description	AverageCost			
▶	Replace front CV Axel	437.500000			
	Transmission check	300.000000			
	Replace spark plugs	260.000000			
	Replace brake pads	250.000000			
	Coolant flush	180.000000			
	Replace battery	150.000000			
	Balance tires	125.000000			
	Oil change	75.000000			
	Diagnose front wheel vibration	62.500000			
	Tire rotation	50.000000			
	Replace air filter	25.000000			

```
188 -- Job with the Highest average cost
189 • SELECT Description, AVG(Amount) AS AverageCost
190 FROM job
191 GROUP BY Description
192 ORDER BY AverageCost DESC
193 LIMIT 1;
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	Description	AverageCost			
▶	Replace front CV Axel	437.500000			





```
195 -- Job with the lowest average cost
196 • SELECT Description, AVG(Amount) AS AverageCost
197 FROM job
198 GROUP BY Description
199 ORDER BY AverageCost ASC
200 LIMIT 1;
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	Description	AverageCost			
▶	Replace air filter	25.000000			

PARTS USAGE ANALYSIS





List the top 5 most frequently used parts and their total usage.

```
202      -- List the top 5 most frequently used parts and their total usage.
203 •    SELECT PartName, SUM(Quantity) AS TotalUsage
204      FROM parts
205      GROUP BY PartName
206      ORDER BY TotalUsage DESC
207      LIMIT 5;
```

Result Grid			Filter Rows:	Export: 	Wrap Cell Content: 	Fetch rows:
	PartName	TotalUsage				
▶	Wheel Weights	4				
	Spark Plugs	4				
	CV Axel	1				
	Shop Materials	1				
	Brake Pads	1				

Calculate the average cost of parts used in repairs.

```
209      -- Calculate the average cost of parts used in repairs.
210 •    SELECT AVG(UnitPrice) AS AveragePartCost FROM parts;
```

Result Grid			Filter Rows:	Export: 	Wrap Cell Content: 
	AveragePartCost				
▶	148.887000				

Determine the total revenue generated from parts sales.

```
212      -- Determine the total revenue generated from parts sales.
213 •    SELECT SUM(Amount) AS TotalPartsRevenue FROM parts;
```

Result Grid			Filter Rows:	Export: 	Wrap Cell Content: 
	TotalPartsRevenue				
▶	1584.87				

FINANCIAL ANALYSIS:

Calculate the total revenue generated from labor and parts for each month.

```
215 -- Calculate the total revenue generated from labor and parts for each month.
216 • SELECT MONTH(invoice.InvoiceDate) AS Month,
217        SUM(invoice.TotalLabour + invoice.TotalParts) AS TotalRevenue
218 FROM invoice
219 GROUP BY MONTH(invoice.InvoiceDate);
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	Month	TotalRevenue			
▶	9	3499.87			

Determine the overall profitability of the repair shop.

```
221 -- Determine the overall profitability of the repair shop.
222 • SELECT SUM(invoice.Total - (invoice.TotalLabour + invoice.TotalParts)) AS Profit
223 FROM invoice;
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	Profit				
▶	454.98310				

Analyze the impact of sales tax on the total revenue.

```
225 -- Analyze the impact of sales tax on the total revenue.
226 • SELECT SUM(invoice.SalesTax) AS TotalSalesTax,
227        SUM(invoice.Total) AS TotalRevenue,
228        (SUM(invoice.SalesTax) / SUM(invoice.Total)) * 100 AS SalesTaxPercentage
229 FROM invoice ;
```

Result Grid				Filter Rows:	Export:	Wrap Cell Content:
	TotalSalesTax	TotalRevenue	SalesTaxPercentage			
▶	454.98310	3954.85310	11.504424779			

ACTIONABLE RECOMMENDATIONS BASED ON ANALYSIS

Identifying Underperforming Services and Marketing Efforts

- **Analyze low-revenue services:** According to the analysis of each job description and its revenue, oil change, diagnose front wheel vibration, tire rotation and replace air filter are the services that need to be upgraded.

Improving Underperforming Services

Bundle services: Combine low-revenue services with higher-margin ones (e.g., offer a brake inspection with an oil change).

Create service packages: Develop packages tailored to specific customer segments (e.g., a "tune-up package" including oil change, air filter replacement, and tire rotation).

Adjust pricing: Consider offering discounts or tiered pricing for multiple services.

Analyze Competitors: Identify successful strategies used by competitors and adapt and innovate to differentiate your offerings.

Gather Customer Feedback: Conduct surveys or customer interviews to understand customer needs and preferences, use feedback to improve services and marketing efforts.

Improve Customer Experience: Offer rewards or loyalty programs.

While it's crucial to address underperforming services, it's equally important to capitalize on those that are already successful. By focusing on high-demand, profitable services, you can increase revenue, improve profit margins, and free up resources to tackle the underperforming areas.

Identifying High-Demand, Profitable Services

The services with consistently high revenue generation are; Replace front CV Axle, transmission check, replace spark plugs, replace brake pads.

Consider customer feedback on high-demand services to ensure quality.

Integrating with Underperforming Service Improvement

Leverage High-Demand Services to Promote Others: Use successful services as a gateway to introduce underperforming ones. For example, offer a discount on a low-demand service when customers purchase a high-demand one.

Allocate Resources Effectively: The increased revenue from high-demand services can fund improvements for underperforming ones. Invest in training, equipment, or marketing to boost their performance.

Cross-Promotion: Promote both high-demand and low-demand services together to increase overall customer satisfaction and loyalty.

Optimizing Parts Inventory

Based on the analysis, it appears that **Wheel Weights** and **Spark Plugs** have the highest usage with a count of 4 each. **CV Axle**, **Shop Materials**, and **Brake Pads** have a count of 1 each.

Recommendation

Increase stock for Wheel Weights and Spark Plugs: Since these parts have the highest usage, maintaining a higher stock level will reduce the risk of stockouts and potential lost sales or service delays.

Monitor usage for CV Axle, Shop Materials, and Brake Pads: While the current stock levels seem sufficient based on the analysis, it's crucial to monitor their usage patterns over a more extended period. If usage increases, consider adjusting stock levels accordingly.

Implement a reorder point system: Determine a specific inventory level for each part that triggers a reorder. This helps maintain optimal stock levels and prevents overstocking or stockouts.

Consider lead times: Factor in the time it takes to replenish stock when determining reorder points.

Analyze cost implications: Evaluate the costs associated with holding excess inventory versus the costs of stockouts.

Optimizing Job Scheduling

- **Identify peak hours:** Analyze job scheduling data to determine peak times and adjust staffing accordingly.
- **Prioritize urgent jobs:** Implement a system to prioritize jobs based on urgency or customer importance.
- **Skill-based scheduling:** Assign jobs to technicians with the appropriate skills to improve efficiency.
- **Reduce wait times:** Implement measures to minimize customer wait times, such as appointment scheduling.

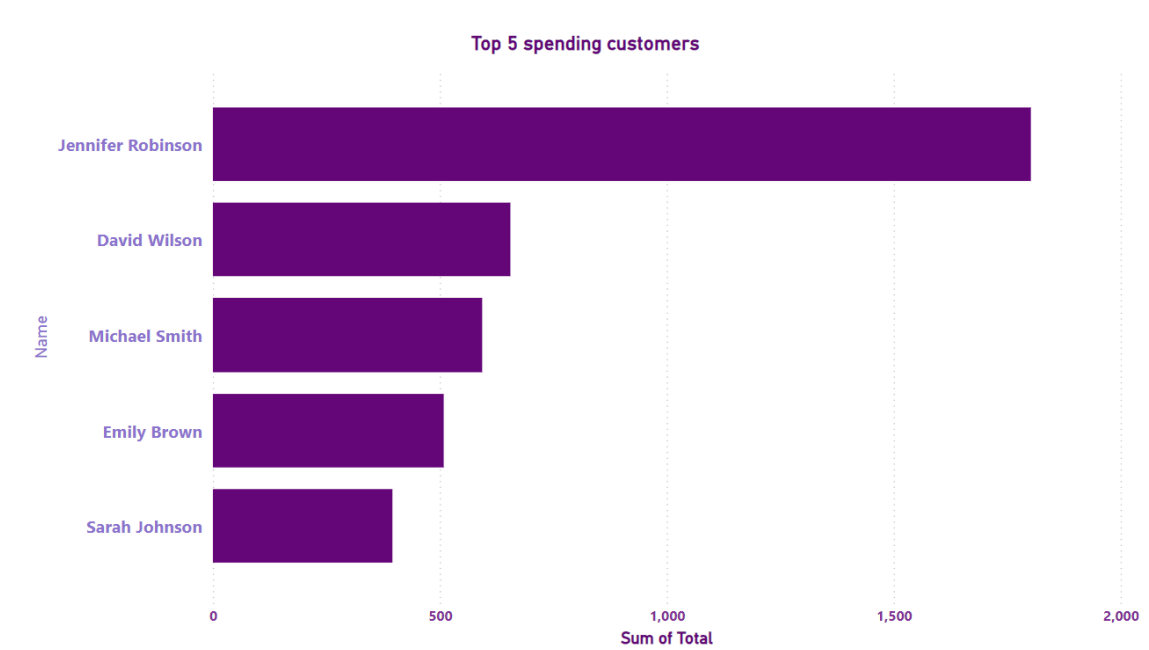
Additional Considerations

- **Employee performance:** Analyze technician productivity and identify areas for improvement through training or incentives.
- **Pricing strategy:** Review pricing for services and parts to ensure profitability and competitiveness.
- **Financial analysis:** Monitor key financial metrics (e.g., revenue, expenses, profit margins) to track business performance.
- **Customer satisfaction:** Implement customer satisfaction surveys to gather feedback and identify areas for improvement.

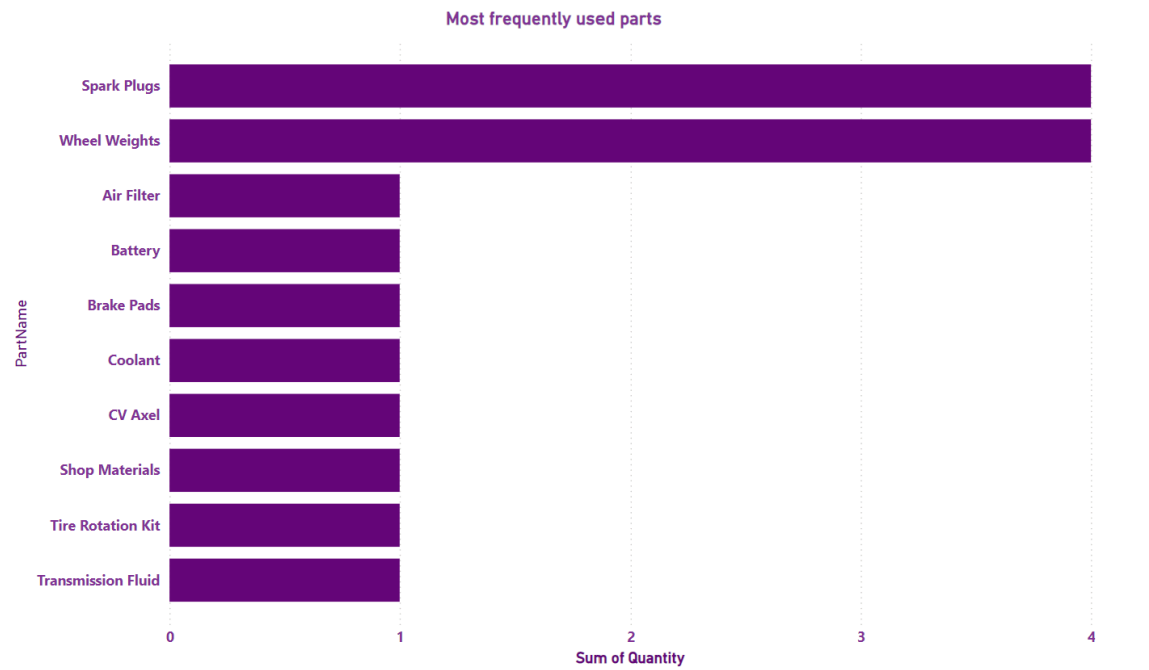
By implementing these recommendations, it can optimize operations, increase revenue, improve customer satisfaction, and enhance overall business performance.

VISUALIZATIONS

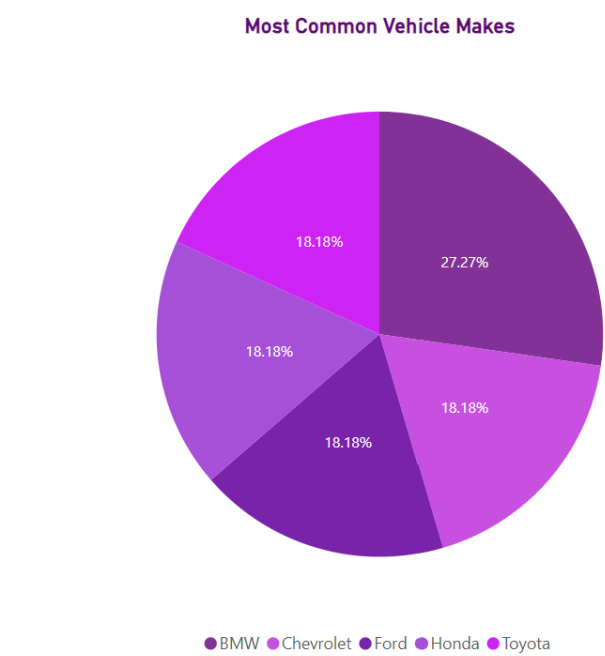
- Top 5 Customers by Spending



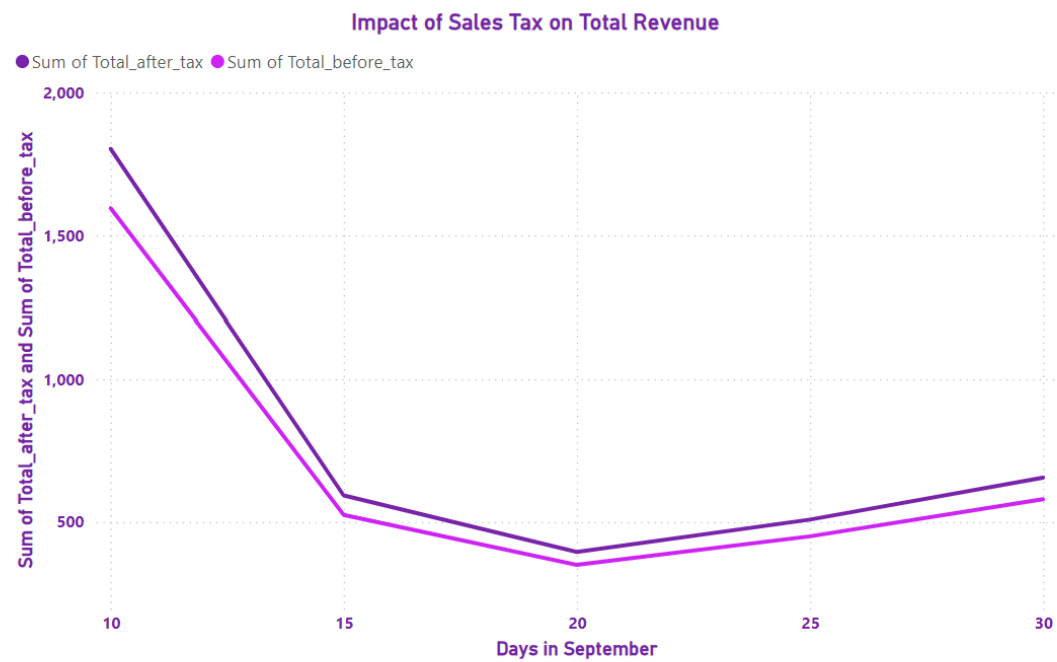
- Top 5 Most Frequently Used Parts



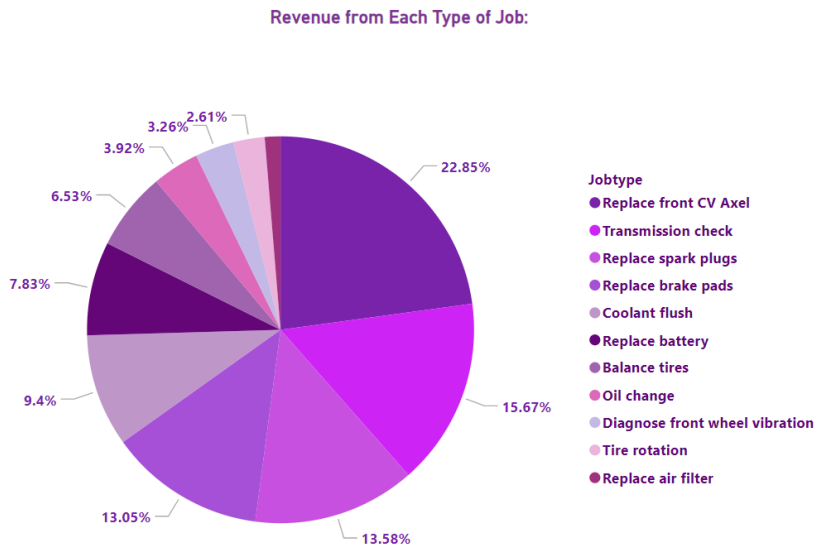
- **Most Common Vehicle Makes**



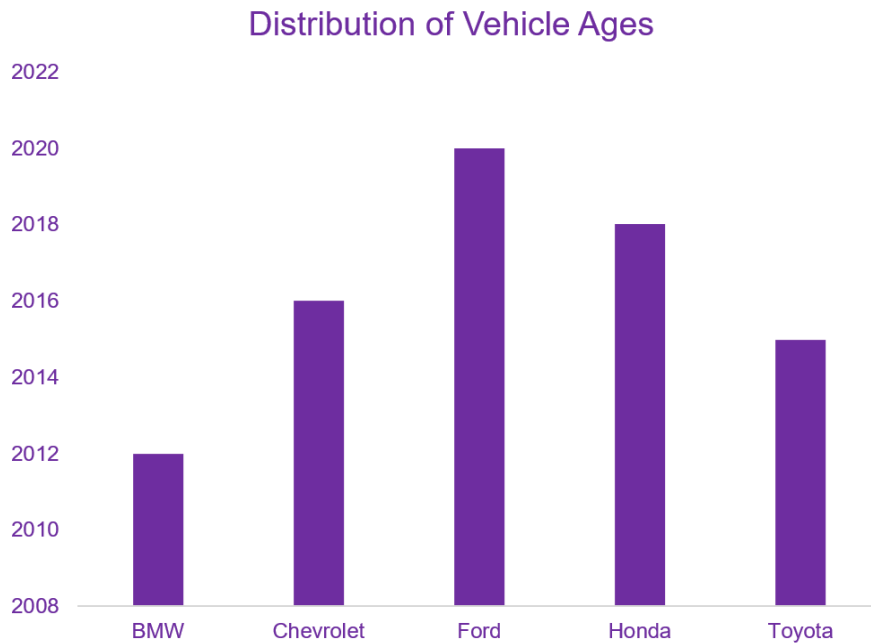
- **Impact of Sales Tax on Total Revenue**



- **Revenue from Each Type of Job**



- **Distribution of Vehicle Ages**



Below is the link to the sql file
[Database export Single Part B Script](#)

