Power & Price: An Analysis of Car Attributes and Market Value

By : Zane Dalco

CSIT 275 – Introduction to R Programming – Final Project

Introduction

Problem:

Showing the links between 1) Average price, and 2) Engine size, Horsepower and 0-60 time among car makes.

Data Used:

2021-2022 Sports Car Prices Dataset

https://www.kaggle.com/datasets/rkiattisak/sports-car-prices-dataset

The Plan

Technique Utilized

1) Import the necessary data and packages

2) Clean the Data and keep the important / useful information

3) Use the psych package to describe and view basic statistical observations

4) Identify car makes with the most and least expensive average price

5) Plot Engine size, horsepower, 0-60, and price against each car make

6) Compare graphs / plot to show whether the car attributions correlate to higher pricing.

Step 1: Importing Data and Packages

```
#Importing necessary Packages:
library(ggplot2)
library(dplyr)
library(psych)


# Import the Data set:
cars_data <- read.csv("SpCarPrice.csv")
```

- “cars_data” is the main data frame I will be doing my work with, taken from the Sports Car Prices dataset. It includes various car attribution data about each car make.
- Dplyr library will be used in order to manipulate the data easier
- Psych library will be used to describe and view basic statistical observations

Step 1: Importing Data and Packages

- We want to convert character columns to numeric

- Sneak peak inside “SpCarPrice.csv”, this is the data set I will be using

	Make	Model	Year	EngineSize	Horsepower	Torque	ZeroSixty	Price
1	Porsche	911	2022	3	379	331	4	101,200
2	Lamborghini	Huracan	2021	5.2	630	443	2.8	274,390
3	Ferrari	488 GTB	2022	3.9	661	561	3	333,750
4	Audi	R8	2022	5.2	562	406	3.2	142,700
5	McLaren	720S	2021	4	710	568	2.7	298,000
6	BMW	M8	2022	4.4	617	553	3.1	130,000
7	Mercedes-Benz	AMG GT	2021	4	523	494	3.8	118,500
8	Chevrolet	Corvette	2021	6.2	490	465	2.8	59,900
9	Ford	Mustang Shelby GT500	2022	5.2	760	625	3.5	81,000
10	Nissan	GT-R Nismo	2021	3.8	600	481	2.5	212,000
11	Aston Martin	DB11	2021	5.2	630	516	3.5	201,495
12	Bugatti	Chiron	2021	8	1500	1180	2.4	3,000,000
13	Dodge	Challenger SRT Hellcat	2022	6.2	717	656	3.5	61,000

Step 2: Cleaning Data

```r
# Fill missing values with appropriate defaults
cars_data[is.na(cars_data)] <- 0

# Define the character columns to be converted to numeric
char_cols_to_numeric <- c("Price", "ZeroSixty", "Year", "EngineSize", "Horsepower", "Torque")

# Loop through the character columns and apply conversion
for (col in char_cols_to_numeric) {
  # Check if the column exists in the data frame
  if (col %in% names(cars_data)) {
    # Apply condition to handle missing values
    cars_data[[col]][is.na(cars_data[[col]])] <- 0

    # Convert the column to numeric
    cars_data[[col]] <- as.numeric(gsub(",", "", as.character(cars_data[[col]])))
  } else {
    warning(paste("Column", col, "not found in the data frame. Skipping conversion."))
  }
}
```

- Fill missing values in a dataset with zeros
- Go through a list of columns, replaces missing values with zeros, and converts each column from text to numbers in a dataset. (for loop)

Step 2 Cont: Finding Averages

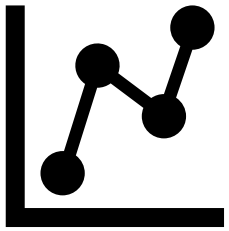
```r
# Function For Finding The Average Horsepower
average_horsepower_function <- function(data) {
  average_horsepower <- data %>%
    #Groups the data by the "Make" column
    group_by(Make) %>%
    # Calculates the mean(average) of the Horsepower column within each group
    # Also creates "Average_Horsepower" to store these values.
    summarise(Average_Horsepower = mean(Horsepower, na.rm = TRUE))
  return(average_horsepower)
}
```

```r
#Function for Finding the Average 0-60 Time
average_0_60_function <- function(data) {
  average_0_60 <- data %>%
    group_by(Make) %>%
    summarise(Average_0_60 = mean(ZeroSixty, na.rm = TRUE))
  return(average_0_60)
}
```

```r
# Function for Finding the Average Torque
average_torque_function <- function(data) {
  average_torque <- data %>%
    group_by(Make) %>%
    summarise(Average_Torque = mean(Torque, na.rm = TRUE))
  return(average_torque)
}
```

- Functions Designed to find the averages for horsepower, 0-60 time, and torque.
- data %>%: Takes the data object and passes it as the first argument to the next expression.
  - %>% allows us to chain operations together, makes it easier to read the code and expressing sequences from left to right

Step 2: Cleaning Data



```
# Remove exact duplicates
unique_cars_data <- distinct(cars_data)
```

- Name of cleaned data frame: “unique_cars_data”
- This data frame has averages calculated for car attributes by grouping them based on their make.
- All NA’s and Empty fields have been removed / altered

Variables:

- Car Make
- Car Model
- Year
- EngineSize
- Horsepower
- Torque
- ZeroSixty
- Price

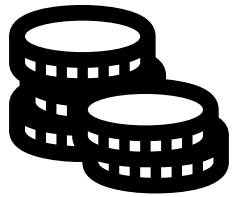
Step 3: Basic Statistical Observations

```
# Basic exploratory analysis
basic_stats <- describe(unique_cars_data)
```

	vars	n	median	min	max
Make*	1	714	17.0	1.0	38.0
Model*	2	714	81.0	1.0	176.0
Year	3	714	2021.0	1965.0	2023.0
EngineSize	4	664	4.0	0.0	8.4
Horsepower	5	710	591.0	181.0	10000.0
Torque	6	711	505.0	0.0	7376.0
ZeroSixty	7	713	3.5	1.8	6.5
Price	8	714	132597.5	25000.0	5200000.0

- This is some basic exploratory analysis using the psych package that shows the median, min, and max for each variable
- For example, we can see that the most expensive car costs $5,200,000 and the least expensive car is $25,000

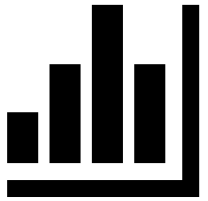
Step 4: Car make with Highest and Lowest average price



```r
# Find the car make with the highest average price
highest_price_make <- average_prices[which.max(average_prices$Average_Price), ]
#Answer: Bugatti

# Find the car make with the lowest average price
lowest_price_make <- average_prices[which.min(average_prices$Average_Price), ]
#Answer: Mazda
```

- These functions help identify the specific row indices where the maximum and minimum average prices occur in the data frame.
- Bugatti is the car make with the most expensive average price.
- Mazda is the car make with the least expensive average price.

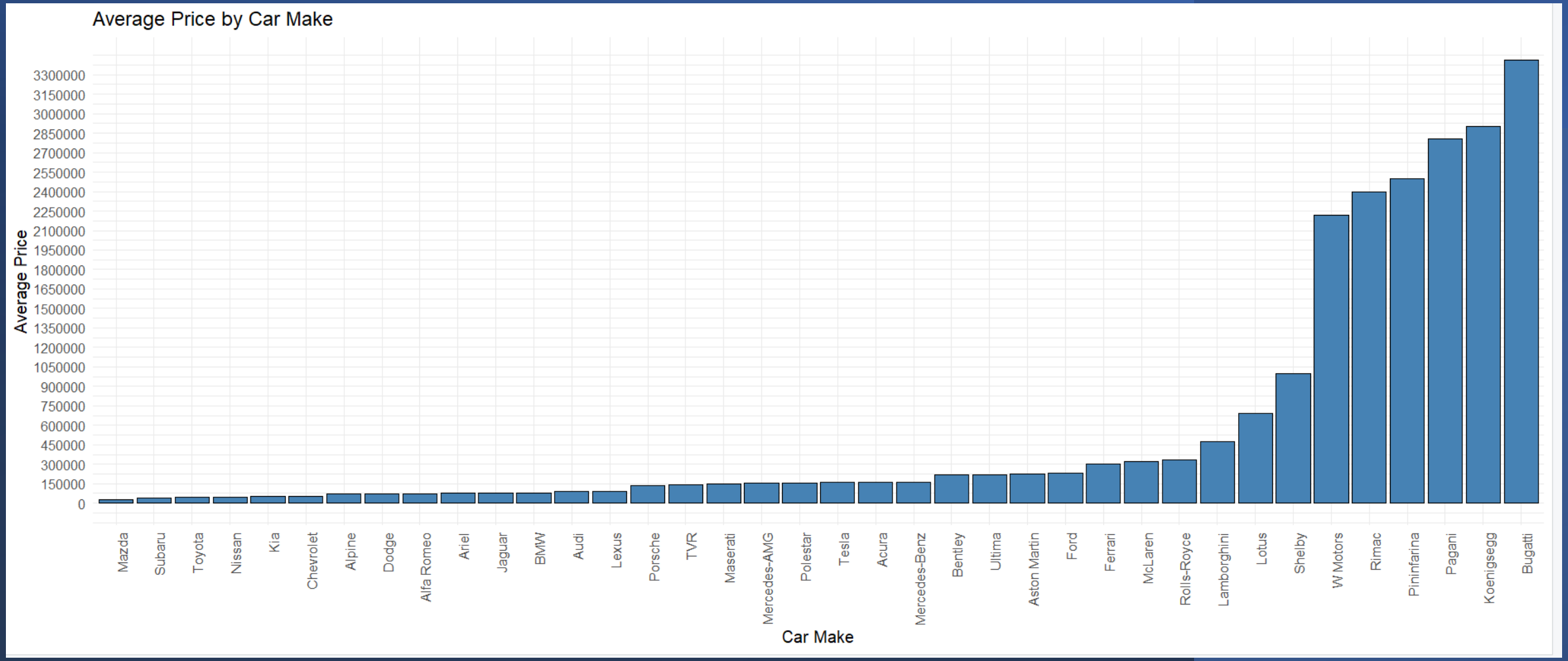
Step 5: Plot Each Car Make by Average Price & Engine Size



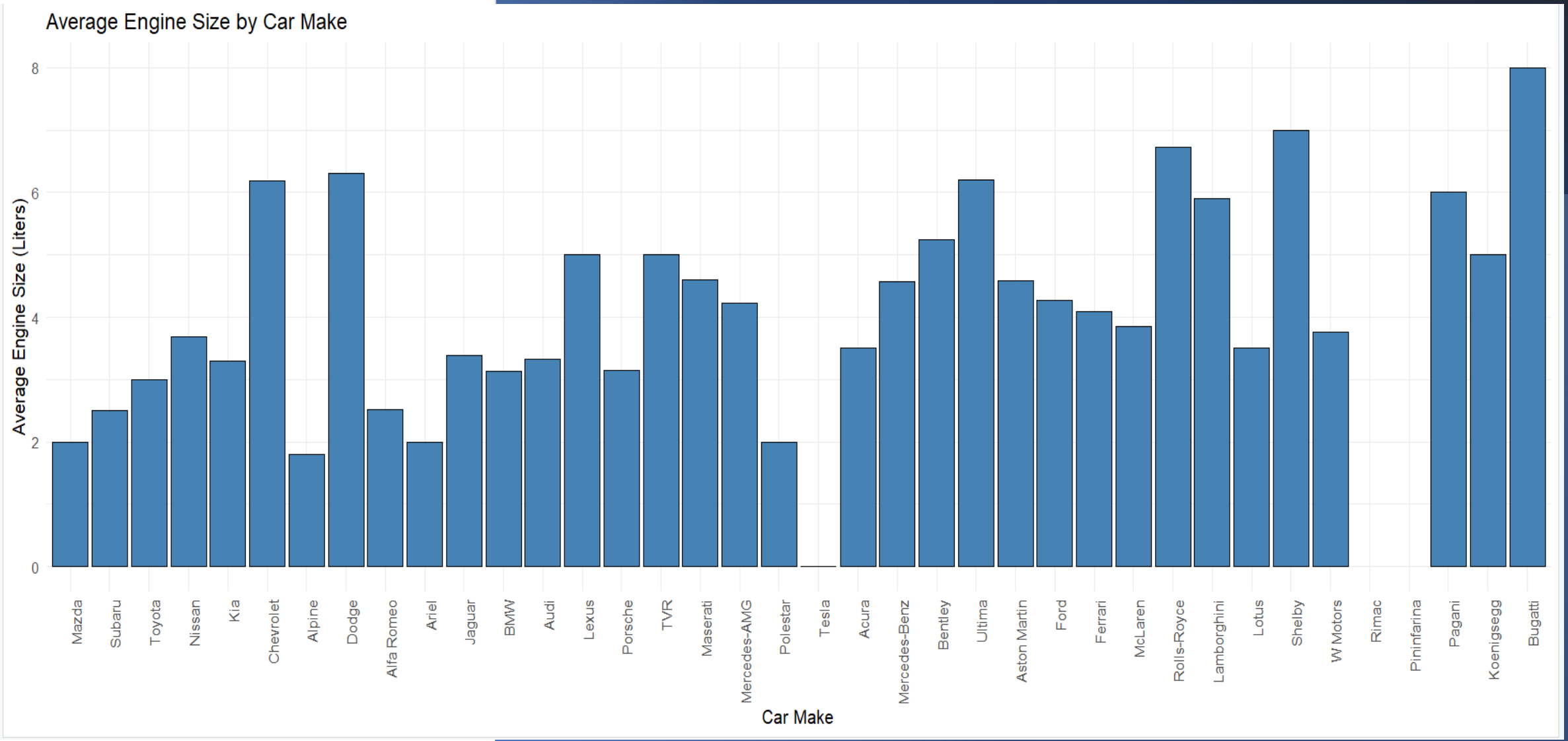
```
#Plotting Car Make and It's average price
ggplot(average_price_engine_size, aes(x=reorder(Make, Average_Price), y=Average_Price)) +
  geom_bar(stat="identity", fill="steelblue", color="black") +
  theme_minimal() +
  labs(x="Car Make", y="Average Price", title="Average Price by Car Make") +
  theme(axis.text.x = element_text(angle=90, hjust=1)) +
  scale_y_continuous(breaks=seq(0, max(average_price_engine_size$Average_Price), by=150000),
                     labels=function(x) format(x, scientific = FALSE))
```

```
# Plot Average Engine Size vs. Car Make using a bar chart
ggplot(average_price_engine_size, aes(x=reorder(Make, Average_Price), y=Average_EngineSize)) +
  geom_bar(stat="identity", fill="steelblue", color="black") +
  theme_minimal() +
  labs(x="Car Make", y="Average Engine Size (Liters)", title="Average Engine Size by Car Make") +
  theme(axis.text.x=element_text(angle=90, hjust=1))
```

- The top block of code will create a plot of Car make and its average price
- The bottom block of code will create a plot of Car make and its average engine size

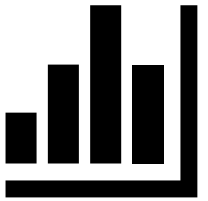


Bar Chart Showing Average Price per Car Make



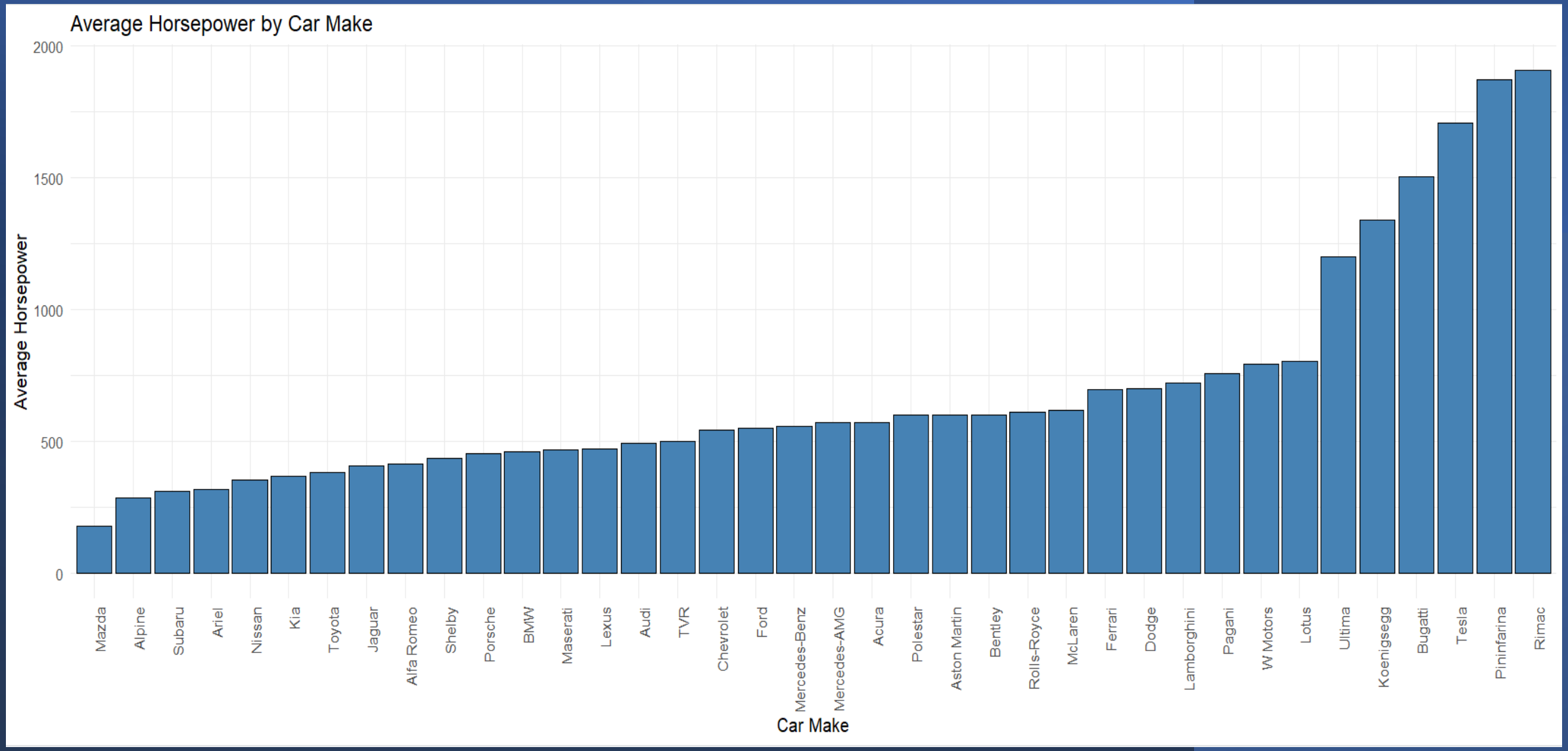
Bar Chart Showing Average Price per Car Make

Step 5 Cont: Plot Each Car Make by Average Horsepower & 0-60 Time

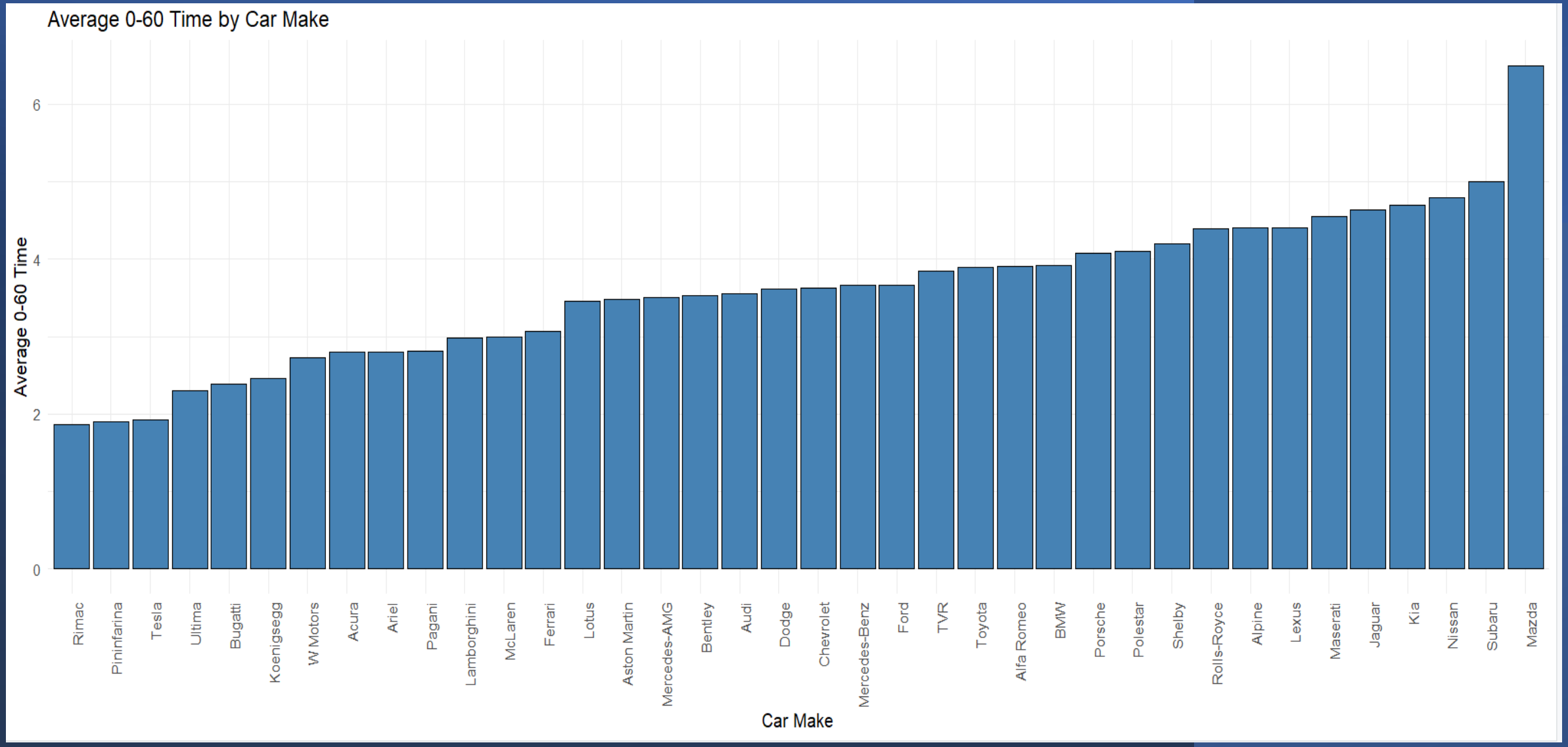


```
# Plot Average Horsepower vs. Car Make using a bar chart
ggplot(average_horsepower, aes(x = reorder(Make, Average_Horsepower), y = Average_Horsepower)) +
  geom_bar(stat = "identity", fill = "steelblue", color = "black") +
  theme_minimal() +
  labs(x = "Car Make", y = "Average Horsepower", title = "Average Horsepower by Car Make") +
  theme(axis.text.x = element_text(angle=90, hjust=1))

# Plot Average 0-60 Time by Car Make using a bar chart
ggplot(average_0_60, aes(x = reorder(Make, Average_0_60), y = Average_0_60)) +
  geom_bar(stat = "identity", fill = "steelblue", color = "black") +
  theme_minimal() +
  labs(x = "Car Make", y = "Average 0-60 Time", title = "Average 0-60 Time by Car Make") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

- The top block of code will create a plot of Car make and its average horsepower
- The bottom block of code will create a plot of Car make and its average 0-60 time



Bar Chart Showing Average Horsepower per Car Make

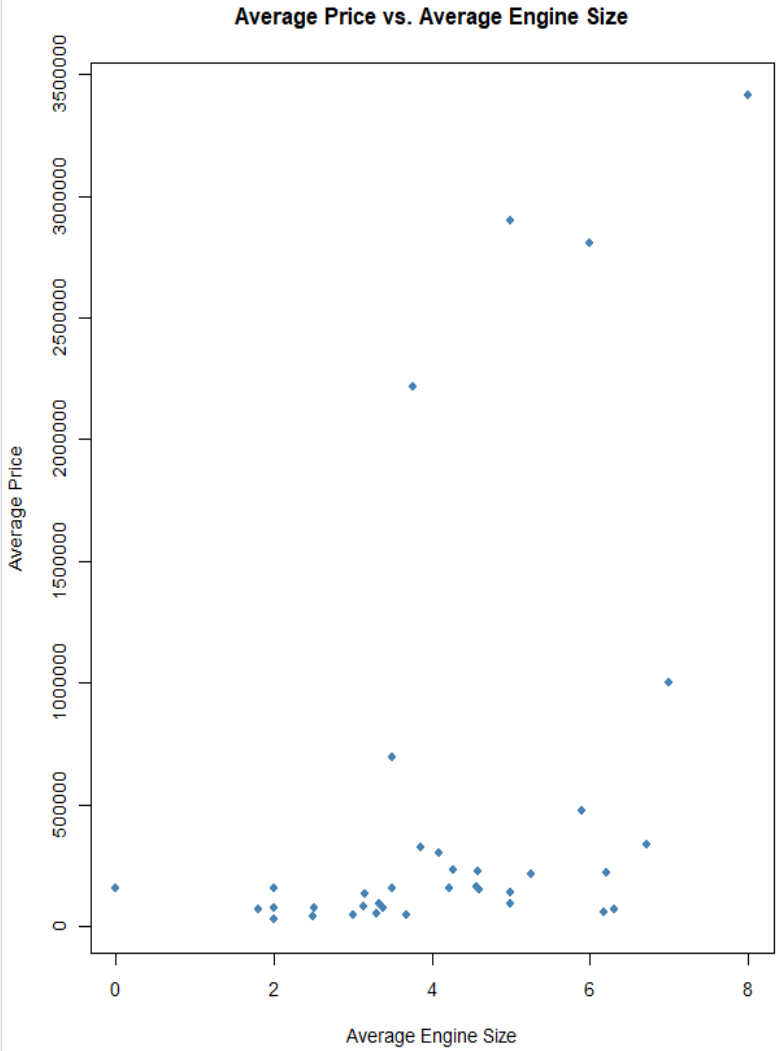


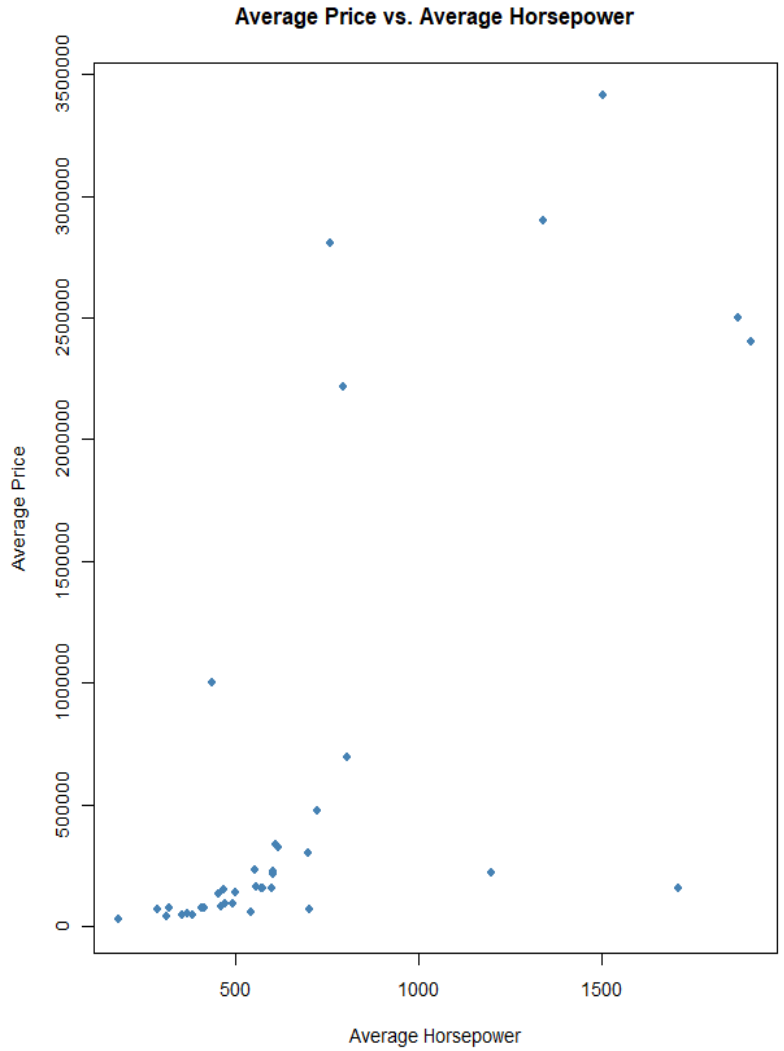
Bar Chart Showing Average 0-60 per Car Make

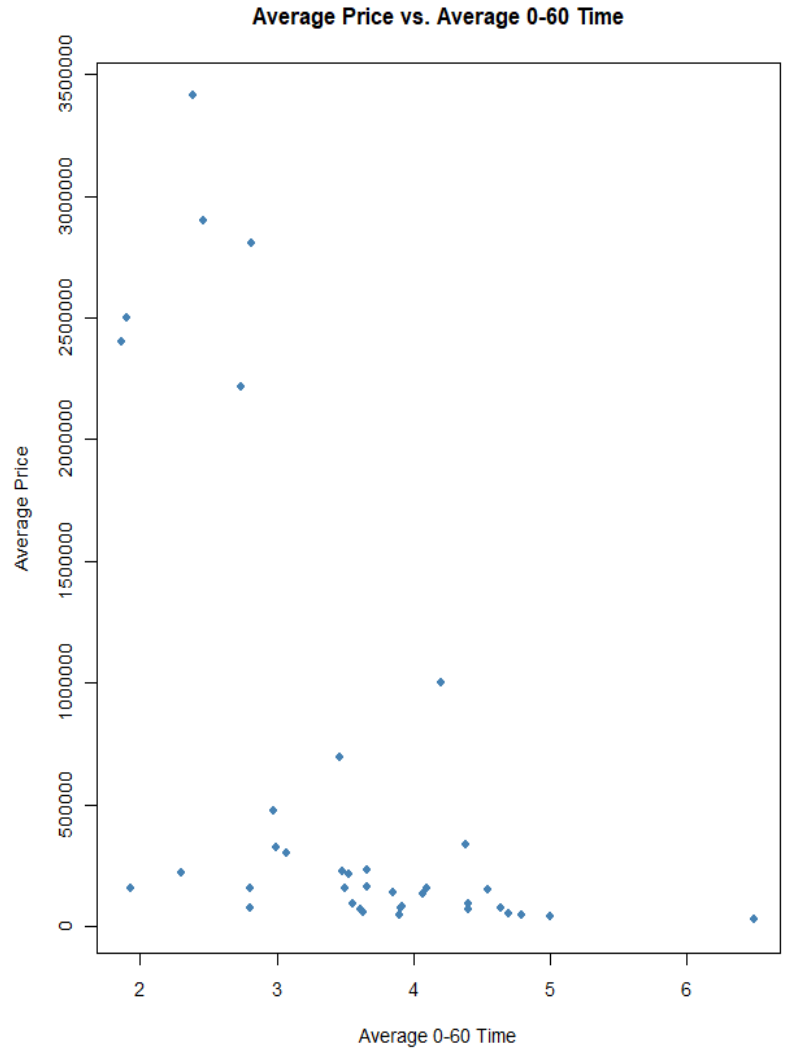
Step 6: Showing Relation between Car Attribution and Price

```r
par(mfrow = c(1, 3))

# Plot 1: Average Price vs. Average Engine Size
plot(average_price_engine_size$Average_EngineSize, average_price_engine_size$Average_Price,
     xlab = "Average Engine Size", ylab = "Average Price", main = "Average Price vs. Average Engine Size",
     col = "steelblue", pch = 19)

# Plot 2: Average Price vs. Average Horsepower
plot(average_horsepower$Average_Horsepower, average_price_engine_size$Average_Price,
     xlab = "Average Horsepower", ylab = "Average Price", main = "Average Price vs. Average Horsepower",
     col = "steelblue", pch = 19)

# Plot 3: Average Price vs. Average 0-60 time
plot(average_0_60$Average_0_60, average_price_engine_size$Average_Price,
     xlab = "Average 0-60 Time", ylab = "Average Price", main = "Average Price vs. Average 0-60 Time",
     col = "steelblue", pch = 19)
```

- Shows the plot of Average engine size, horsepower, and 0-60 time with the average price side by side
- Makes it easier to compare data and see correlation







Plots of Average Engine Size, Horsepower and 0-60 time by the Average price

Conclusion

- **My findings suggest a modest correlation between engine size, horsepower, torque, 0-60 time, and car prices.**
- **However, it's important to note that this correlation is not consistently significant across all categories, with outliers such as electric cars and high-end sports cars contributing to variations in the relationship.**


Questions?

Thank You for Your Time!