Exploratory Data Analysis of Used Cars

#Intro: In this notebook, we will perform an exploratory data analysis (EDA) on a used car dataset. The primary goals are to clean and preprocess the data, understand the dataset's structure, and visualize critical relationships among variables. To uncover insights into the used car market, we will analyze various attributes, including price, mileage, and model year. Additionally, we will create visualizations to highlight trends and patterns, such as the relationship between mileage and cost, and identify the most popular and expensive car brands. This analysis aims to provide a comprehensive overview of the factors affecting used car pricing and performance.

Load libraries

```r
install.packages("tidyverse")
```

```
##
## The downloaded binary packages are in
##
/var/folders/k7/3hkxc3916d94sh54b7xgkrfh0000gn/T//Rtmp7W54ur/downloaded_packa
ges
```

```r
install.packages("knitr")
```

```
##
## The downloaded binary packages are in
##
/var/folders/k7/3hkxc3916d94sh54b7xgkrfh0000gn/T//Rtmp7W54ur/downloaded_packa
ges
```

```r
install.packages("scales")
```

```
##
## The downloaded binary packages are in
##
/var/folders/k7/3hkxc3916d94sh54b7xgkrfh0000gn/T//Rtmp7W54ur/downloaded_packa
ges
```

```r
install.packages("ggplot2")
```

```
##
## The downloaded binary packages are in
##
/var/folders/k7/3hkxc3916d94sh54b7xgkrfh0000gn/T//Rtmp7W54ur/downloaded_packa
ges
```

```r
install.packages("corrplot")
```

```
##
## The downloaded binary packages are in
##
/var/folders/k7/3hkxc3916d94sh54b7xgkrfh0000gn/T//Rtmp7W54ur/downloaded_packa
ges

install.packages("reshape2")

##
## The downloaded binary packages are in
##
/var/folders/k7/3hkxc3916d94sh54b7xgkrfh0000gn/T//Rtmp7W54ur/downloaded_packa
ges

library(corrplot)

## corrplot 0.94 loaded

library(ggplot2)
library(tidyverse)
library(knitr)
library(readr)
library(scales)

##
## Attaching package: 'scales'

## The following object is masked from 'package:purrr':
##
##     discard

## The following object is masked from 'package:readr':
##
##     col_factor

library(dplyr)
library(lubridate)
library(reshape2)

##
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':
##
##     smiths
```

Import dataset

```
used_cars_dataset <- read_csv("~/Desktop/Data sets/used_cars.csv")

## Rows: 4009 Columns: 12
## ── Column specification
## Delimiter: ","
## chr (11): brand, model, milage, fuel_type, engine, transmission, ext_col,
in...
## dbl  (1): model_year
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this
message.
```

Cleaning

```
# Rename the column
colnames(used_cars_dataset)[colnames(used_cars_dataset) == "milage"] <-
"mileage"

# Remove any non-numeric characters (if necessary)
used_cars_dataset$price <- gsub("[^0-9.]", "", used_cars_dataset$price)

# Convert the cleaned character column to numeric
used_cars_dataset$price <- as.numeric(used_cars_dataset$price)

# Check the structure to confirm the change
str(used_cars_dataset)
```

```
## spc_tbl_ [4,009 × 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ brand       : chr [1:4009] "Ford" "Hyundai" "Lexus" "INFINITI" ...
##  $ model       : chr [1:4009] "Utility Police Interceptor Base" "Palisade
SEL" "RX 350 RX 350" "Q50 Hybrid Sport" ...
##  $ model_year  : num [1:4009] 2013 2021 2022 2015 2021 ...
##  $ mileage     : chr [1:4009] "51,000 mi." "34,742 mi." "22,372 mi."
"88,900 mi." ...
##  $ fuel_type   : chr [1:4009] "E85 Flex Fuel" "Gasoline" "Gasoline"
"Hybrid" ...
##  $ engine      : chr [1:4009] "300.0HP 3.7L V6 Cylinder Engine Flex Fuel
Capability" "3.8L V6 24V GDI DOHC" "3.5 Liter DOHC" "354.0HP 3.5L V6 Cylinder
Engine Gas/Electric Hybrid" ...
##  $ transmission: chr [1:4009] "6-Speed A/T" "8-Speed Automatic"
"Automatic" "7-Speed A/T" ...
##  $ ext_col     : chr [1:4009] "Black" "Moonlight Cloud" "Blue" "Black" ...
##  $ int_col     : chr [1:4009] "Black" "Gray" "Black" "Black" ...
##  $ accident    : chr [1:4009] "At least 1 accident or damage reported" "At
least 1 accident or damage reported" "None reported" "None reported" ...
##  $ clean_title : chr [1:4009] "Yes" "Yes" NA "Yes" ...
##  $ price       : num [1:4009] 10300 38005 54598 15500 34999 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   brand = col_character(),
##   ..   model = col_character(),
##   ..   model_year = col_double(),
##   ..   milage = col_character(),
##   ..   fuel_type = col_character(),
##   ..   engine = col_character(),
##   ..   transmission = col_character(),
##   ..   ext_col = col_character(),
##   ..   int_col = col_character(),
##   ..   accident = col_character(),
##   ..   clean_title = col_character(),
##   ..   price = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
sum(is.na(used_cars_dataset$price))
```

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

```
# Clean and convert the mileage column
used_cars_dataset$mileage <- gsub(" mi\\.$", "", used_cars_dataset$mileage)
used_cars_dataset$mileage <- gsub(",", "", used_cars_dataset$mileage)
used_cars_dataset$mileage <- as.numeric(used_cars_dataset$mileage)
```

Overview of dataset

```
colnames(used_cars_dataset) #List of column names
```

```
##  [1] "brand"        "model"        "model_year"   "mileage"
"fuel_type"
##  [6] "engine"       "transmission" "ext_col"      "int_col"
"accident"
## [11] "clean_title"  "price"
```

```
ncol(used_cars_dataset) #How many columns are in data frame?
```

```
## [1] 12
```

```
nrow(used_cars_dataset) #How many rows are in data frame?
```

```
## [1] 4009
```

```
dim(used_cars_dataset)  #Dimensions of the data frame?
```

```
## [1] 4009   12
```

```
head(used_cars_dataset)  #See the first 6 rows of data frame.
```

```
## # A tibble: 6 × 12
##   brand   model model_year mileage fuel_type engine transmission ext_col
int_col
##   <chr>   <chr>      <dbl>   <dbl> <chr>     <chr>  <chr>        <chr>
<chr>
## 1 Ford    Util…        2013   51000 E85 Flex… 300.0… 6-Speed A/T   Black
```

```
Black
## 2 Hyundai Pali…       2021   34742 Gasoline  3.8L … 8-Speed Aut…
Moonli… Gray
## 3 Lexus   RX 3…       2022   22372 Gasoline  3.5 L… Automatic    Blue
Black
## 4 INFINI… Q50 …        2015   88900 Hybrid    354.0… 7-Speed A/T  Black
Black
## 5 Audi    Q3 4…       2021    9835 Gasoline  2.0L … 8-Speed Aut…
Glacie… Black
## 6 Acura   ILX …       2016  136397 Gasoline  2.4 L… F            Silver
Ebony.
## # ℹ 3 more variables: accident <chr>, clean_title <chr>, price <dbl>
```

```
str(used_cars_dataset)  #See list of columns and data types (numeric,
character, etc)
```

```
## spc_tbl_ [4,009 × 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ brand       : chr [1:4009] "Ford" "Hyundai" "Lexus" "INFINITI" ...
##  $ model       : chr [1:4009] "Utility Police Interceptor Base" "Palisade
SEL" "RX 350 RX 350" "Q50 Hybrid Sport" ...
##  $ model_year  : num [1:4009] 2013 2021 2022 2015 2021 ...
##  $ mileage     : num [1:4009] 51000 34742 22372 88900 9835 ...
##  $ fuel_type   : chr [1:4009] "E85 Flex Fuel" "Gasoline" "Gasoline"
"Hybrid" ...
##  $ engine      : chr [1:4009] "300.0HP 3.7L V6 Cylinder Engine Flex Fuel
Capability" "3.8L V6 24V GDI DOHC" "3.5 Liter DOHC" "354.0HP 3.5L V6 Cylinder
Engine Gas/Electric Hybrid" ...
##  $ transmission: chr [1:4009] "6-Speed A/T" "8-Speed Automatic"
"Automatic" "7-Speed A/T" ...
##  $ ext_col     : chr [1:4009] "Black" "Moonlight Cloud" "Blue" "Black" ...
##  $ int_col     : chr [1:4009] "Black" "Gray" "Black" "Black" ...
##  $ accident    : chr [1:4009] "At least 1 accident or damage reported" "At
least 1 accident or damage reported" "None reported" "None reported" ...
##  $ clean_title : chr [1:4009] "Yes" "Yes" NA "Yes" ...
##  $ price       : num [1:4009] 10300 38005 54598 15500 34999 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   brand = col_character(),
##   ..   model = col_character(),
##   ..   model_year = col_double(),
##   ..   milage = col_character(),
##   ..   fuel_type = col_character(),
##   ..   engine = col_character(),
##   ..   transmission = col_character(),
##   ..   ext_col = col_character(),
##   ..   int_col = col_character(),
##   ..   accident = col_character(),
##   ..   clean_title = col_character(),
##   ..   price = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
summary(used_cars_dataset)  #Statistical summary of data. Mainly for numerics
```

```
##     brand              model             model_year      mileage      
##  Length:4009        Length:4009        Min.   :1974   Min.   :   100  
##  Class :character   Class :character   1st Qu.:2012   1st Qu.: 23044  
##  Mode  :character   Mode  :character   Median :2017   Median : 52775  
##                                        Mean   :2016   Mean   : 64718  
##                                        3rd Qu.:2020   3rd Qu.: 94100  
##                                        Max.   :2024   Max.   :405000  
##   fuel_type            engine          transmission         ext_col         
##  Length:4009        Length:4009        Length:4009        Length:4009       
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##                                                                             
##                                                                             
##                                                                             
##    int_col            accident         clean_title            price        
##  Length:4009        Length:4009        Length:4009        Min.   :   2000  
##  Class :character   Class :character   Class :character   1st Qu.:  17200  
##  Mode  :character   Mode  :character   Mode  :character   Median :  31000  
##                                                           Mean   :  44553  
##                                                           3rd Qu.:  49990  
##                                                           Max.   :2954083
```

```
names(used_cars_dataset)
```

```
##  [1] "brand"        "model"        "model_year"   "mileage"      
"fuel_type"   
##  [6] "engine"       "transmission" "ext_col"      "int_col"      
"accident"    
## [11] "clean_title"  "price"
```

Quick glance summary

```
# Summary statistics for price
summary(used_cars_dataset$price)
```

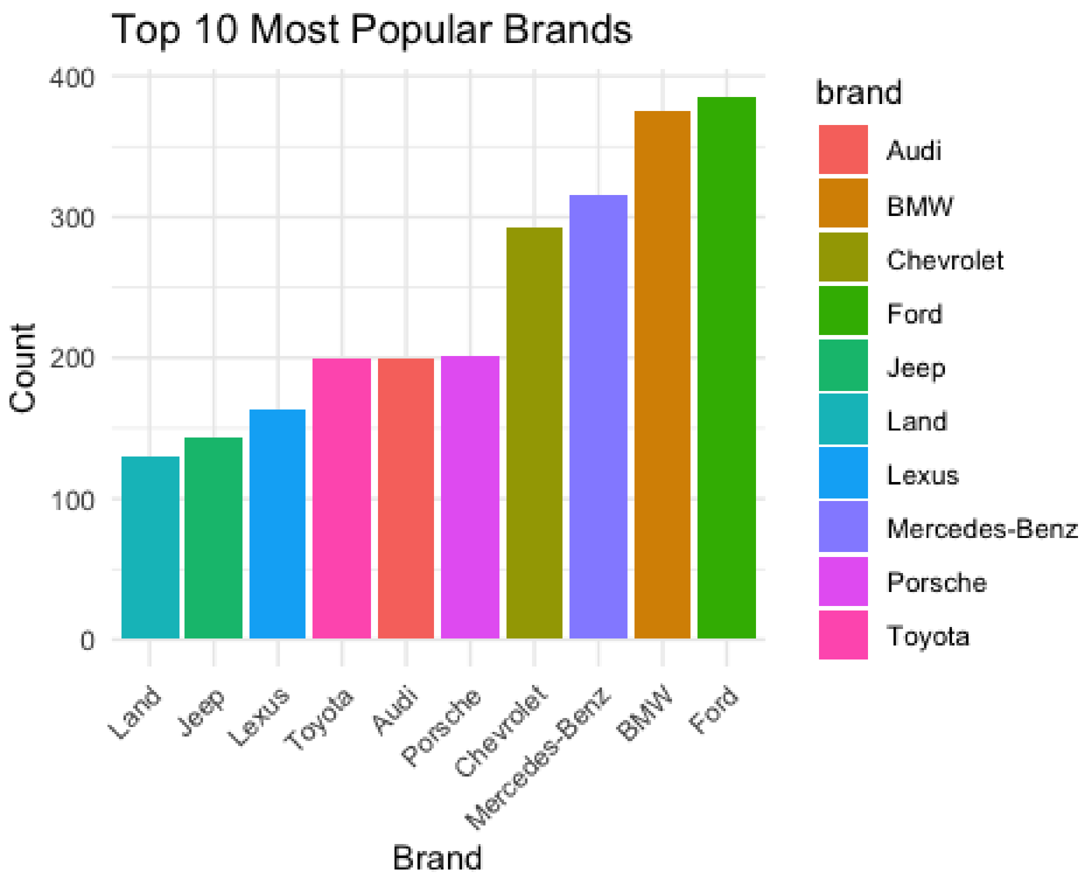
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    2000   17200   31000   44553   49990 2954083
```

```
# Summary statistics for other relevant numeric columns
summary(used_cars_dataset[, c("mileage", "model_year", "price")])
```

```
##     mileage         model_year       price        
##  Min.   :   100   Min.   :1974   Min.   :   2000  
##  1st Qu.: 23044   1st Qu.:2012   1st Qu.:  17200  
##  Median : 52775   Median :2017   Median :  31000  
##  Mean   : 64718   Mean   :2016   Mean   :  44553
##  3rd Qu.: 94100   3rd Qu.:2020   3rd Qu.:  49990  
##  Max.   :405000   Max.   :2024   Max.   :2954083
```

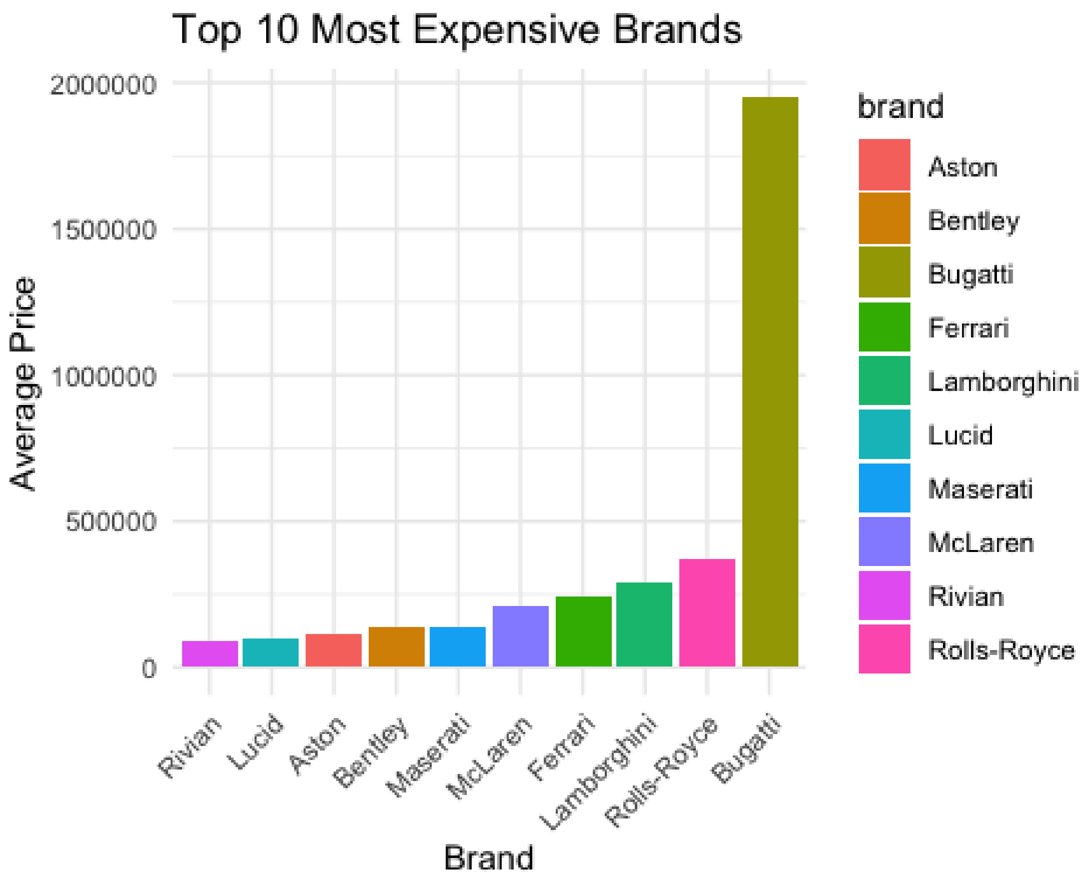
Visualizations

```
# Bar chart of the 10 most popular brands
top_brands <- used_cars_dataset %>%
  group_by(brand) %>%
  summarize(count = n()) %>%
  arrange(desc(count)) %>%
  slice_head(n = 10)

ggplot(top_brands, aes(x = reorder(brand, count), y = count, fill = brand)) +
  geom_bar(stat = "identity") +
  labs(title = "Top 10 Most Popular Brands",
       x = "Brand",
       y = "Count") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



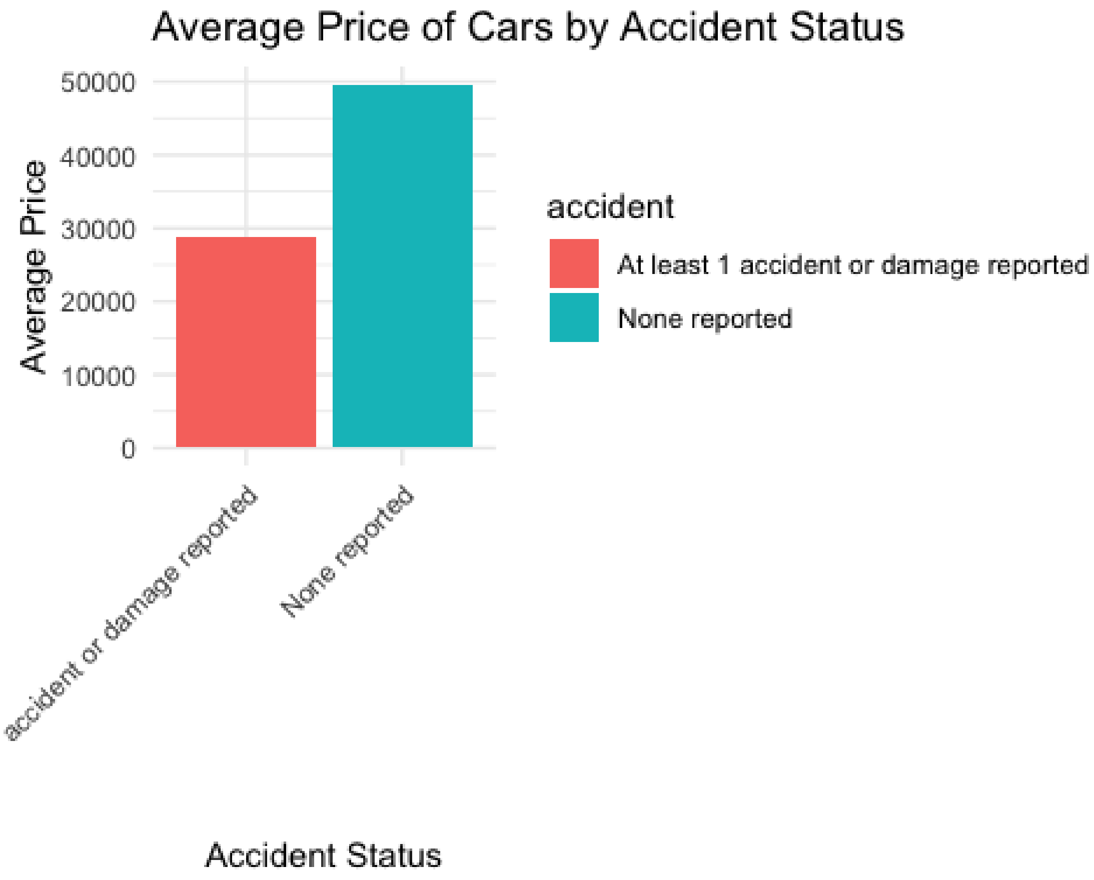
```
# Calculate the average price for each brand and select the top 10 most
expensive
```

```
top_expensive_brands <- used_cars_dataset %>%
  group_by(brand) %>%
  summarize(avg_price = mean(price, na.rm = TRUE)) %>%
  arrange(desc(avg_price)) %>%
  slice_head(n = 10)

# Bar chart of the 10 most expensive brands
ggplot(top_expensive_brands, aes(x = reorder(brand, avg_price), y =
avg_price, fill = brand)) +
  geom_bar(stat = "identity") +
  labs(title = "Top 10 Most Expensive Brands",
       x = "Brand",
       y = "Average Price") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

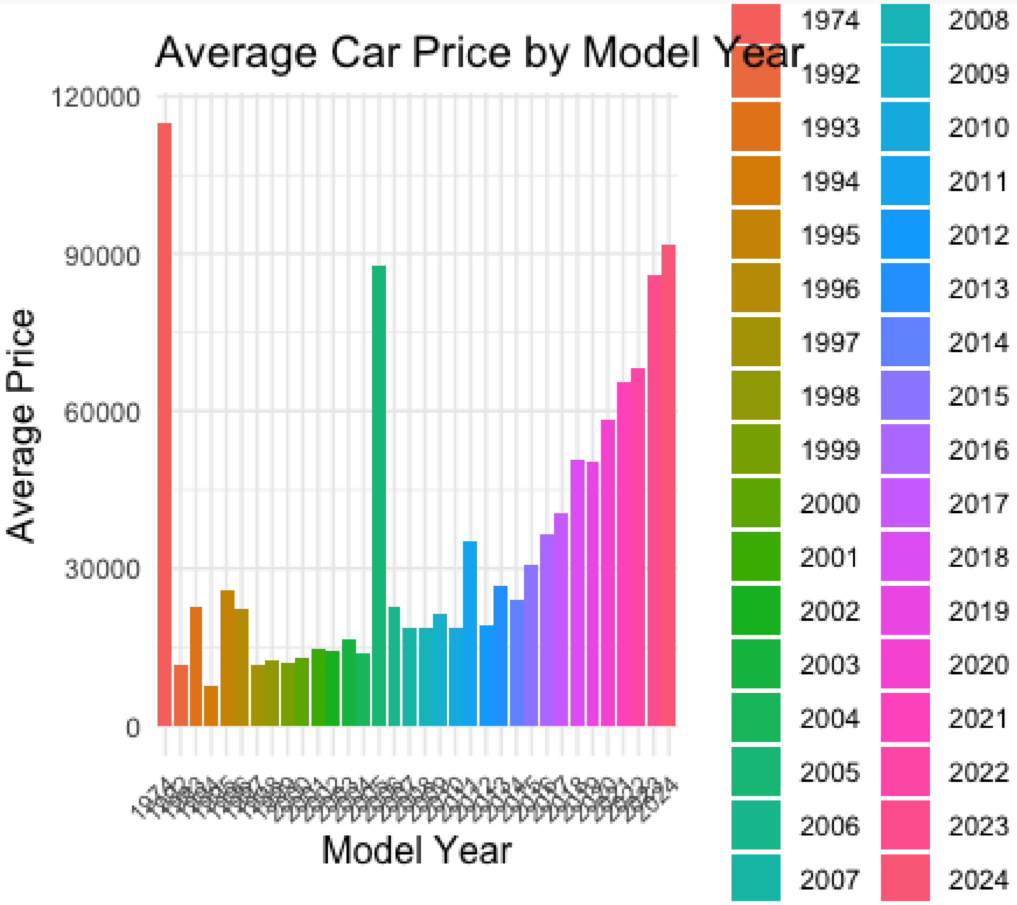


```
# Average price based on accident status
average_price_accident <- used_cars_dataset %>%
  group_by(accident) %>%
  summarize(avg_price = mean(price, na.rm = TRUE)) %>%
  filter(accident %in% c("At least 1 accident or damage reported", "None
reported"))
```

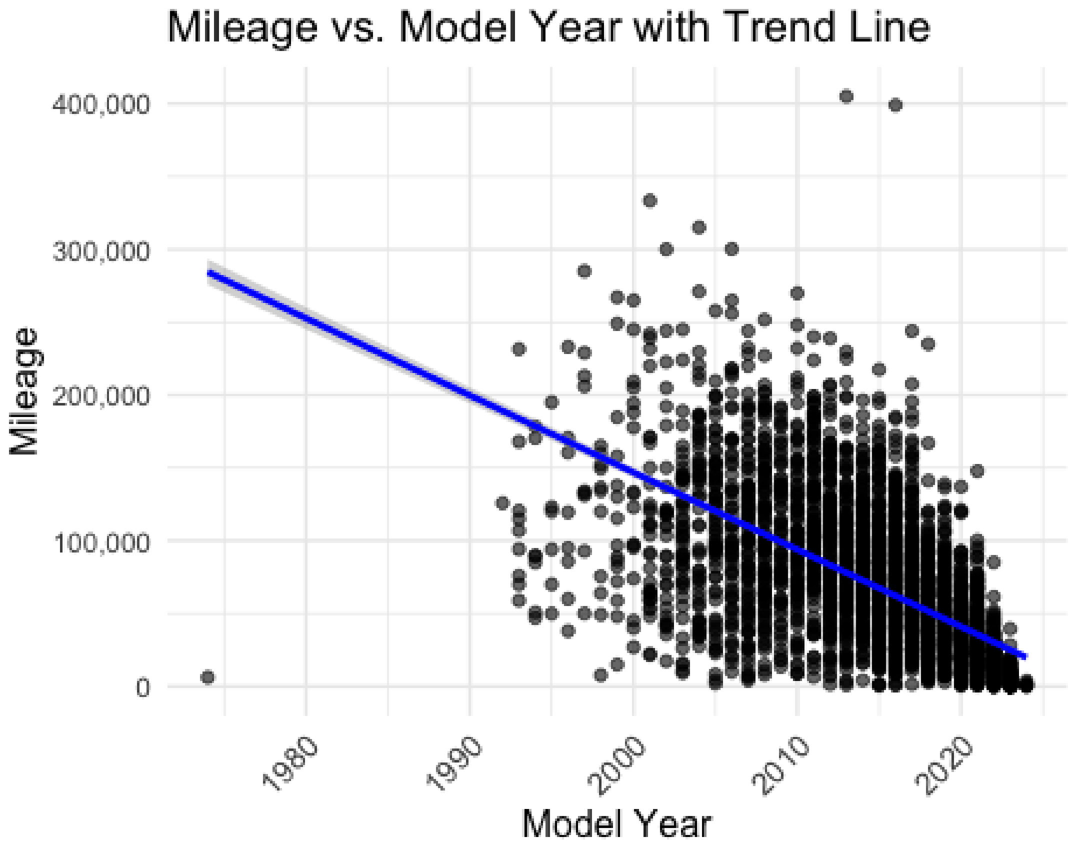
```r
ggplot(average_price_accident, aes(x = accident, y = avg_price, fill =
accident)) +
  geom_bar(stat = "identity") +
  labs(title = "Average Price of Cars by Accident Status",
       x = "Accident Status",
       y = "Average Price") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



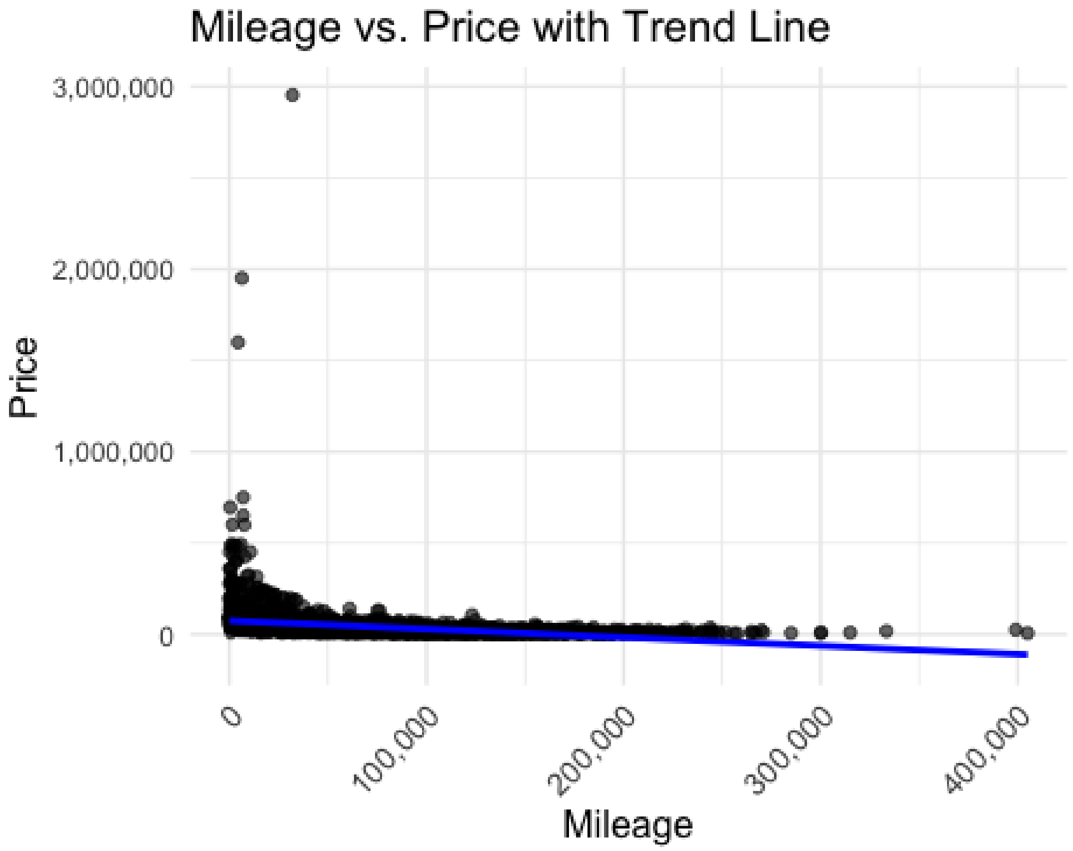
```r
# Average Car Price by Model Year
average_price_by_year <- used_cars_dataset %>%
  group_by(model_year) %>%
  summarize(avg_price = mean(price, na.rm = TRUE)) %>%
  arrange(model_year)

ggplot(average_price_by_year, aes(x = factor(model_year), y = avg_price, fill
= factor(model_year))) +
  geom_bar(stat = "identity") +
  labs(title = "Average Car Price by Model Year",
       x = "Model Year",
       y = "Average Price") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, size = 8),  #
Rotate and shrink x-axis labels
        axis.title = element_text(size = 12),
        plot.title = element_text(size = 14),
        plot.margin = margin(10, 20, 10, 10))
```



```
# Mileage vs. Model Year with Trend Line
ggplot(used_cars_dataset, aes(x = model_year, y = mileage)) +
  geom_point(alpha = 0.6) +
  geom_smooth(method = "lm", color = "blue") +
  labs(title = "Mileage vs. Model Year with Trend Line",
       x = "Model Year",
       y = "Mileage") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, size = 10),
        axis.title = element_text(size = 12),
        plot.title = element_text(size = 14),
        plot.margin = margin(10, 10, 10, 10)) +
  scale_y_continuous(labels = scales::comma)

## `geom_smooth()` using formula = 'y ~ x'
```

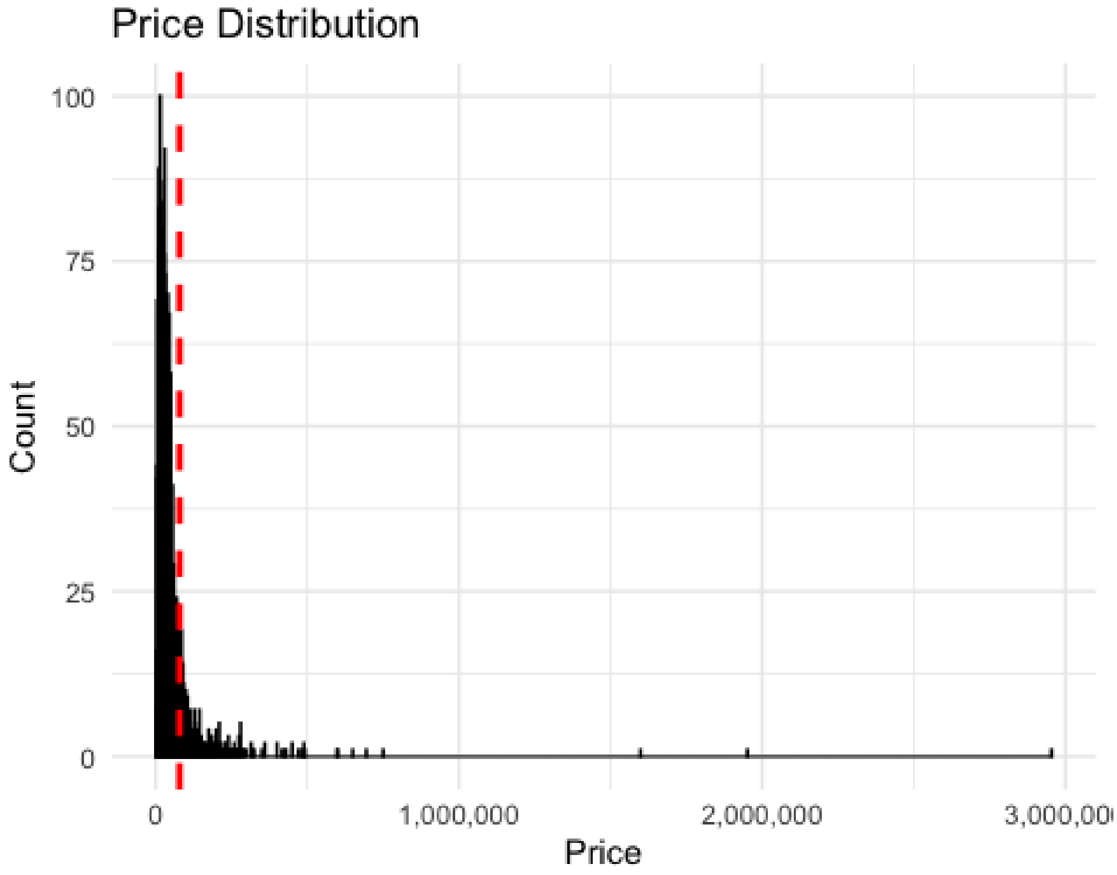


```
# Mileage vs. Price with Trend Line
ggplot(used_cars_dataset, aes(x = mileage, y = price)) +
  geom_point(alpha = 0.6) +
  geom_smooth(method = "lm", color = "blue") +
  labs(title = "Mileage vs. Price with Trend Line",
       x = "Mileage",
       y = "Price") +
  scale_x_continuous(labels = scales::comma) +
  scale_y_continuous(labels = scales::comma) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, size = 10),
        axis.title = element_text(size = 12),
        plot.title = element_text(size = 14),
        plot.margin = margin(10, 10, 10, 10))

## `geom_smooth()` using formula = 'y ~ x'
```

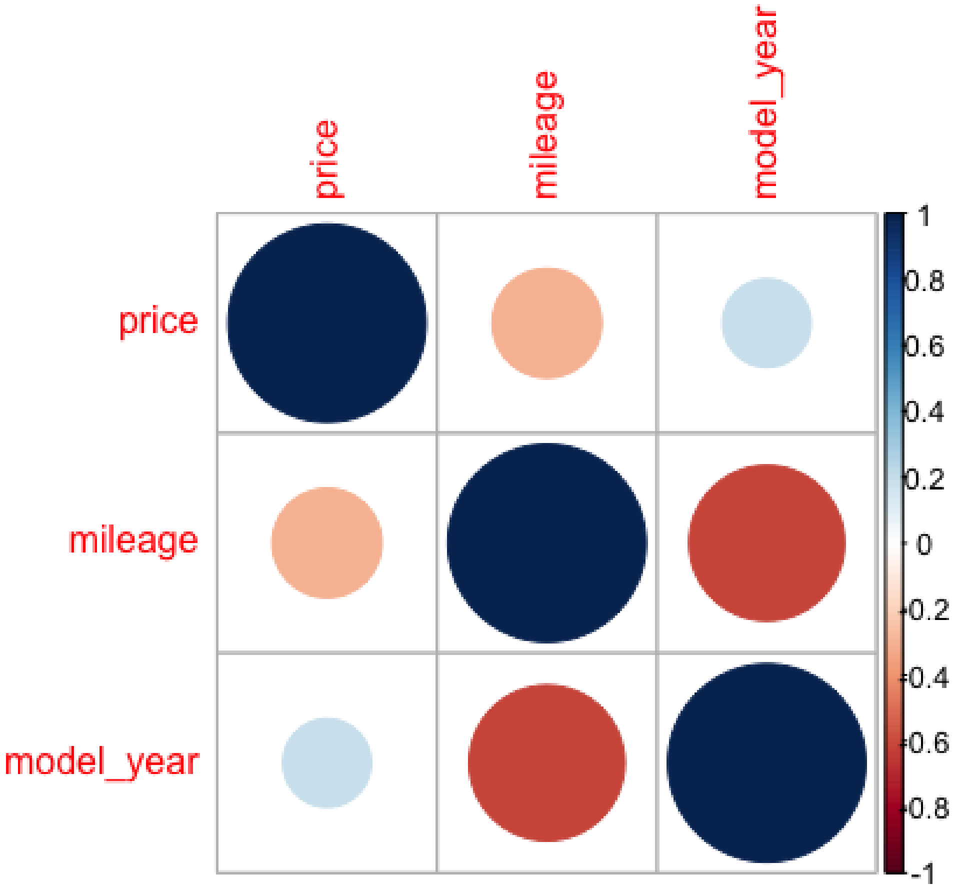


```
peak_price <-
used_cars_dataset$price[which.max(table(used_cars_dataset$price))]

# Histogram of price
ggplot(used_cars_dataset, aes(x = price)) +
  geom_histogram(binwidth = 1000, fill = "blue", color = "black") +
  labs(title = "Price Distribution", x = "Price", y = "Count") +
  theme_minimal() +
  scale_x_continuous(labels = scales::comma) +
  geom_vline(aes(xintercept = peak_price), color = "red", linetype =
"dashed", size = 1) +
  geom_text(aes(x = peak_price, y = Inf, label = paste("Peak:", peak_price)),
            vjust = -0.5, color = "red")

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

## Warning in geom_text(aes(x = peak_price, y = Inf, label = paste("Peak:", :
All aesthetics have length 1, but the data has 4009 rows.
## ℹ Please consider using `annotate()` or provide this layer with data
```

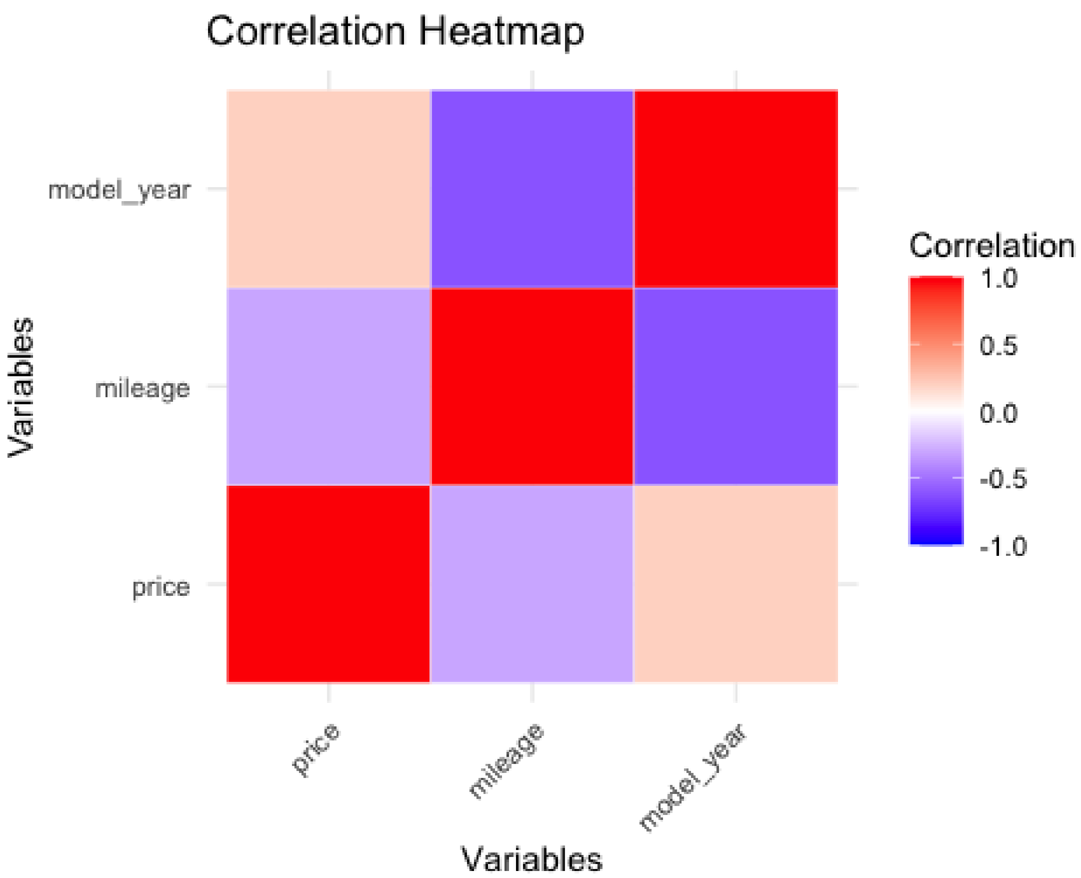
```
containing
##   a single row.
```



```
# Correlation
numeric_data <- used_cars_dataset %>%
  select(price, mileage, model_year)

correlation_matrix <- cor(numeric_data, use = "complete.obs")
print(correlation_matrix)

##                  price    mileage model_year
## price       1.0000000 -0.3055281  0.1994962
## mileage    -0.3055281  1.0000000 -0.6177204
## model_year  0.1994962 -0.6177204  1.0000000

corrplot(correlation_matrix, method = "circle")
```



```r
correlation_data <- melt(correlation_matrix)

ggplot(correlation_data, aes(Var1, Var2, fill = value)) +
  geom_tile(color = "white") +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                       midpoint = 0, limit = c(-1, 1), space = "Lab",
                       name = "Correlation") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1)) +
  labs(title = "Correlation Heatmap", x = "Variables", y = "Variables")
```



#Conclusion: Looking at the data, Ford is the most popular brand in total sales, while Bugatti is the most expensive brand. Cars with at least one accident report sell for approximately two times less than those with no reported accidents. The selling price of the model years of vehicles varies between cars made before 1974, rising in price likely due to the vintage factor and newer cars selling at higher prices likely due to their modernity. There is an outlier of vehicles sold in 2005, which is significantly higher than other years relatively close to it, which could spark further analysis. Cars with higher mileage also tend to sell less according to the trend line for the graph "Mileage vs. Price Year with Trend Line." the same is actual with model year according to "Mileage vs. Model Year with Trend Line" and as stated earlier.