German_Car_Analysis.R

gagan

2025-02-19

```r
# Load necessary libraries
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(ggplot2)
library(dplyr)
library(readr)

# Loading the dataset
df <- read_csv("~/CSV_Data_Sets/germany.csv")
```

```
## Rows: 46405 Columns: 9
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (5): make, model, fuel, gear, offerType
## dbl (4): mileage, price, hp, 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.
```

```r
# Checking the structure and first few rows
str(df)
```

```
## spc_tbl_ [46,405 x 9] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ mileage  : num [1:46405] 235000 92800 149300 96200 156000 ...
##  $ make     : chr [1:46405] "BMW" "Volkswagen" "SEAT" "Renault" ...
##  $ model    : chr [1:46405] "316" "Golf" "Exeo" "Megane" ...
##  $ fuel     : chr [1:46405] "Diesel" "Gasoline" "Gasoline" "Gasoline" ...
##  $ gear     : chr [1:46405] "Manual" "Manual" "Manual" "Manual" ...
##  $ offerType: chr [1:46405] "Used" "Used" "Used" "Used" ...
```

```
##  $ price    : num [1:46405] 6800 6877 6900 6950 6950 ...
##  $ hp       : num [1:46405] 116 122 160 110 156 99 131 116 150 86 ...
##  $ year     : num [1:46405] 2011 2011 2011 2011 2011 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   mileage = col_double(),
##   ..   make = col_character(),
##   ..   model = col_character(),
##   ..   fuel = col_character(),
##   ..   gear = col_character(),
##   ..   offerType = col_character(),
##   ..   price = col_double(),
##   ..   hp = col_double(),
##   ..   year = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
head(df)
```

```
## # A tibble: 6 x 9
##   mileage make       model  fuel             gear   offerType price    hp  year
##     <dbl> <chr>      <chr>  <chr>            <chr>  <chr>     <dbl> <dbl> <dbl>
## 1  235000 BMW        316    Diesel           Manual Used       6800   116  2011
## 2   92800 Volkswagen Golf   Gasoline         Manual Used       6877   122  2011
## 3  149300 SEAT       Exeo   Gasoline         Manual Used       6900   160  2011
## 4   96200 Renault    Megane Gasoline         Manual Used       6950   110  2011
## 5  156000 Peugeot    308    Gasoline         Manual Used       6950   156  2011
## 6  147000 Toyota     Auris  Electric/Gasoline Autom~ Used       6950    99  2011
```

```
summary(df)
```

```
##     mileage            make              model              fuel          
##  Min.   :      0   Length:46405       Length:46405       Length:46405      
##  1st Qu.:  19800   Class :character   Class :character   Class :character  
##  Median :  60000   Mode  :character   Mode  :character   Mode  :character  
##  Mean   :  71178                                                           
##  3rd Qu.: 105000                                                           
##  Max.   :1111111                                                           
##                                                                            
##      gear            offerType             price               hp        
##  Length:46405       Length:46405       Min.   :   1100   Min.   :  1     
##  Class :character   Class :character   1st Qu.:   7490   1st Qu.: 86     
##  Mode  :character   Mode  :character   Median :  10999   Median :116     
##                                        Mean   :  16572   Mean   :133     
##                                        3rd Qu.:  19490   3rd Qu.:150     
##                                        Max.   :1199900   Max.   :850     
##                                                          NA's   :29      
##       year     
##  Min.   :2011  
##  1st Qu.:2013  
##  Median :2016  
##  Mean   :2016  
##  3rd Qu.:2019
```

```
##  Max.   :2021
##
```

```
#Data Cleaning & Preprocessing
#Handle missing values, remove duplicates, and convert categorical data

# Checking for missing values
colSums(is.na(df))
```

```
##   mileage      make     model      fuel      gear offerType     price        hp
##         0         0       143         0       182         0         0        29
##      year
##         0
```

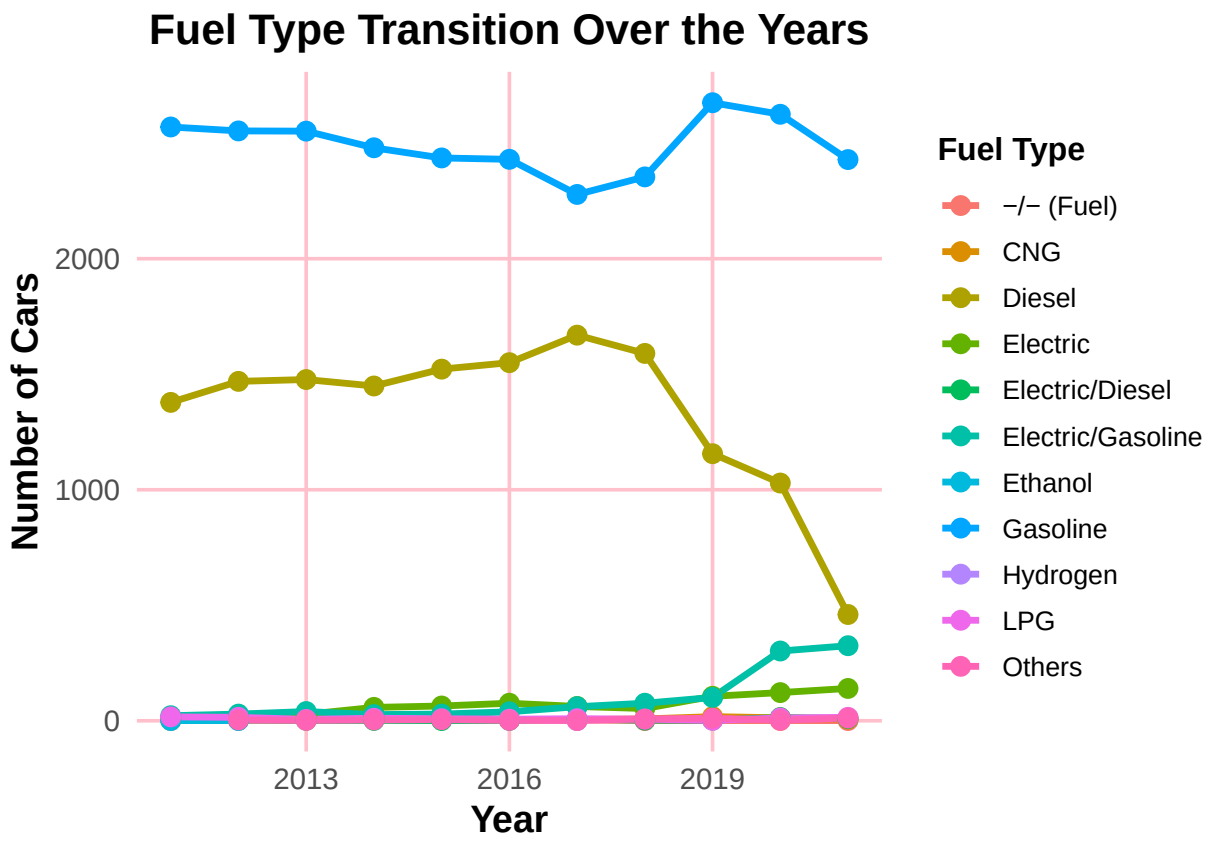
```
# Remove duplicate rows
df <- df %>% distinct()

# Convert categorical variables to factors
df$Fuel_Type <- as.factor(df$fuel)
df$Gear_Type <- as.factor(df$gear)
df$Brand <- as.factor(df$make)



# Visualizing the Gear to EV Transition
#Analyze how fuel types have changed over the years

# Counting the number of cars by Fuel Type per year
fuel_trend <- df %>%
  group_by(year, Fuel_Type) %>%
  summarise(count = n())
```

```
## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.
```

```
# Create the plot with enhancements
ggplot(fuel_trend, aes(x = year, y = count, color = Fuel_Type, group = Fuel_Type)) +
  geom_line(size = 1.2) +  # Thicker lines for better visibility
  geom_point(size = 3) +   # Add points for each data point
  theme_minimal(base_size = 14) +  # Use a minimal theme with larger font
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold", size = 16),
    axis.title.x = element_text(face = "bold", size = 14),
    axis.title.y = element_text(face = "bold", size = 14),
    legend.title = element_text(face = "bold", size = 12),
    legend.text = element_text(size = 10),
    panel.grid.major = element_line(color = "pink"),
    panel.grid.minor = element_blank()
  ) +
  labs(
    title = "Fuel Type Transition Over the Years",
    x = "Year",
    y = "Number of Cars",
```

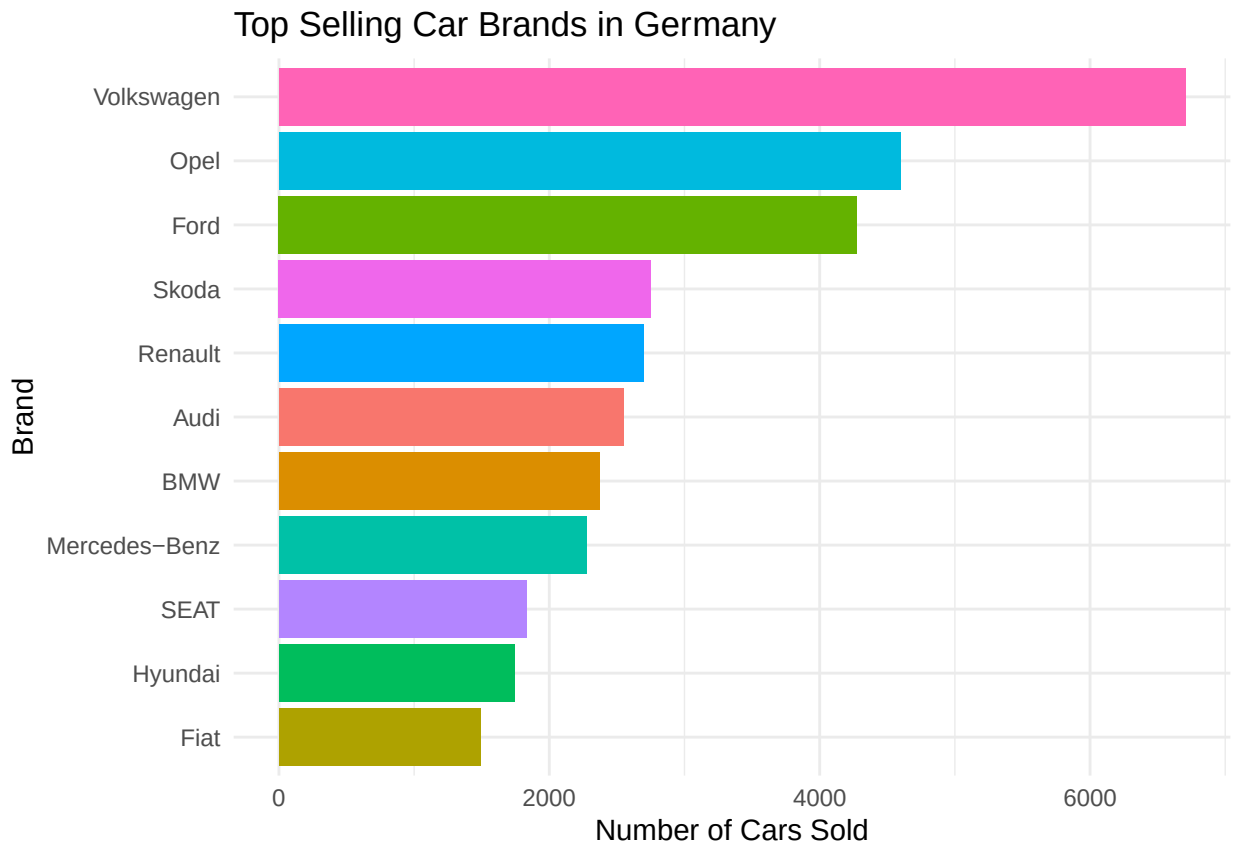
```
    color = "Fuel Type"  # Legend title
  )
```

```
## 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.
```




```
#####



#Identifying the Best-Selling Cars

# Finding the top-selling brands
top_brands <- df %>%
  count(Brand, sort = TRUE) %>%
  top_n(11, n)

# Bar plot of top-selling brands
ggplot(top_brands, aes(x = reorder(Brand, n), y = n, fill = Brand)) +
  geom_bar(stat = "identity",
           show.legend = FALSE) +
  coord_flip() +
```

```
  theme_minimal() +
  labs(title = "Top Selling Car Brands in Germany",
       x = "Brand",
       y = "Number of Cars Sold")
```




```
####



#Analyze the Trend of Car Prices Over Time
# Calculating average price per year for each brand
price_trend <- df %>%
  group_by(year, Brand) %>%
  summarise(Average_Price = mean(price, na.rm = TRUE))
```

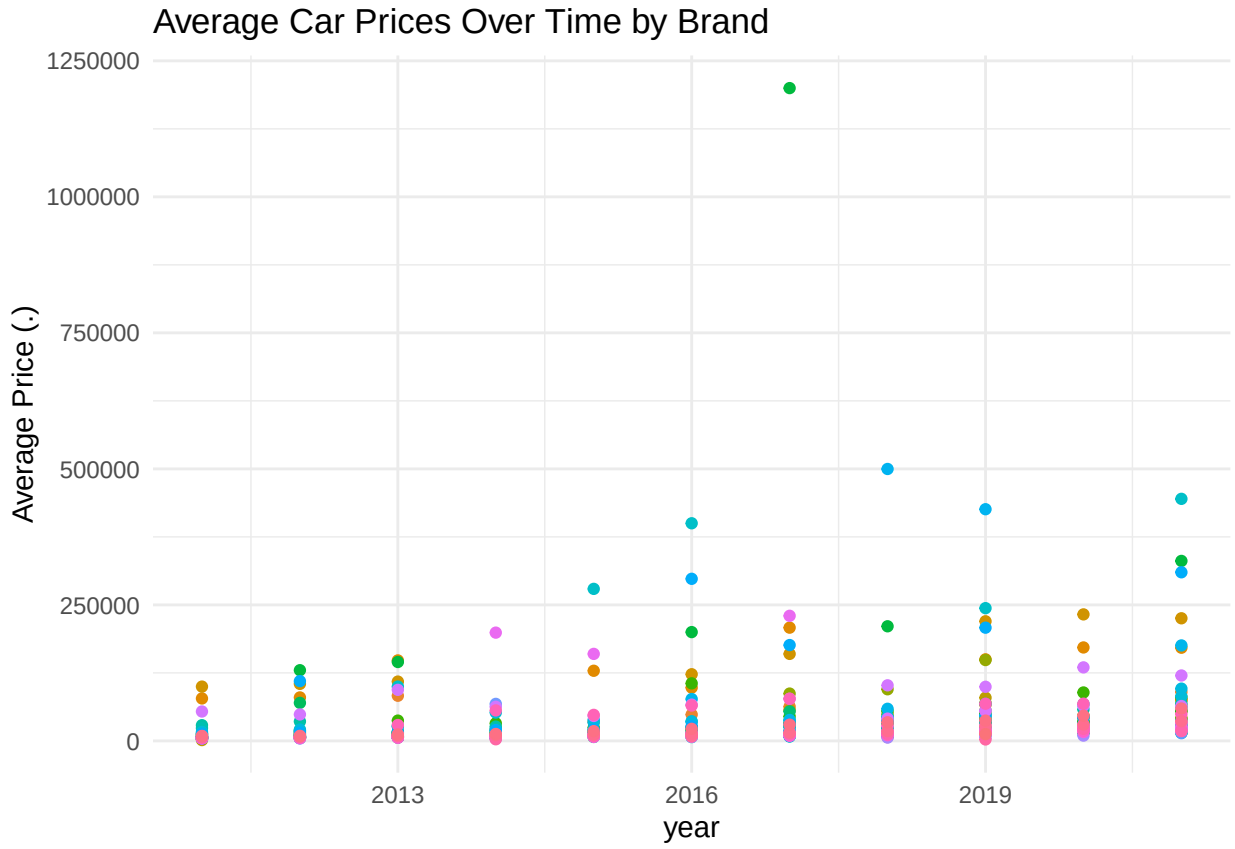
```
## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.
```

```
# View the summarized data
head(price_trend)
```

```
## # A tibble: 6 x 3
## # Groups:   year [1]
```

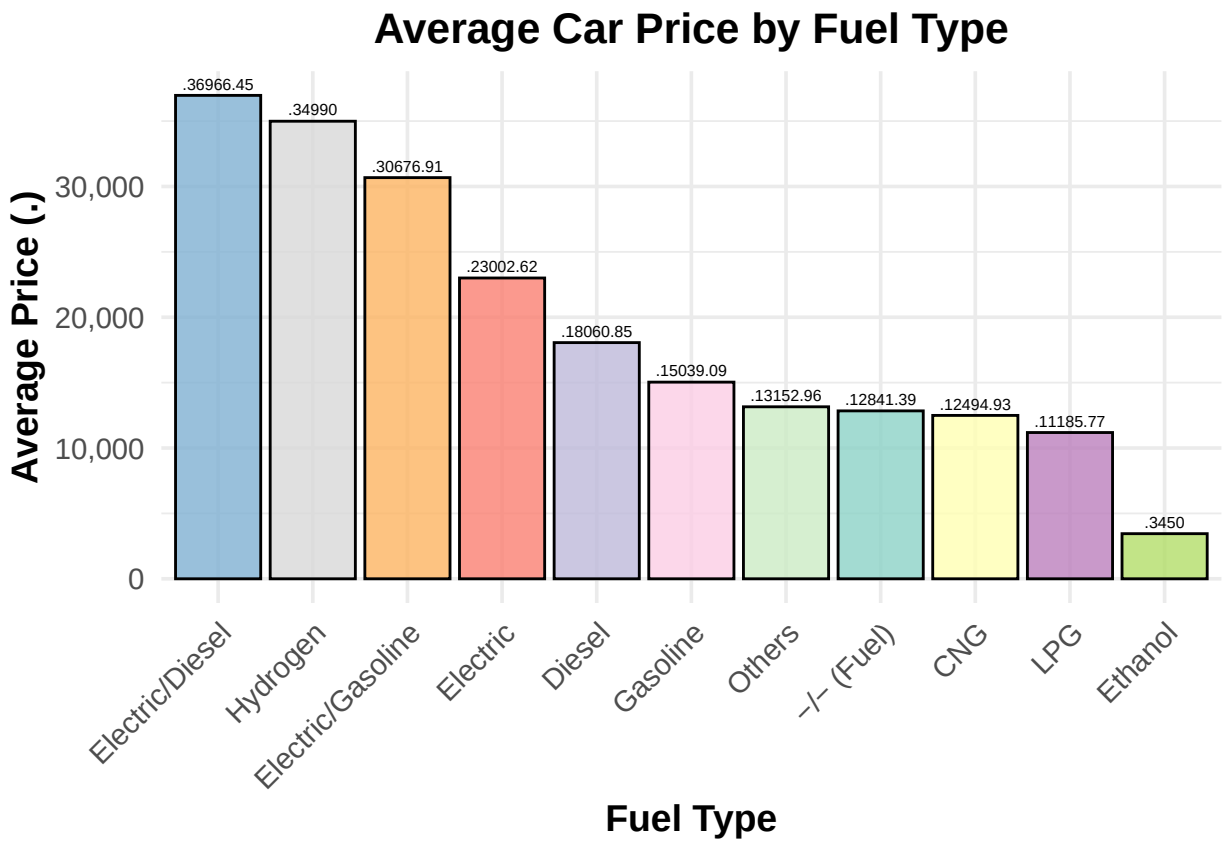
```
##    year Brand   Average_Price
##   <dbl> <fct>           <dbl>
## 1  2011 Abarth          5990
## 2  2011 Alfa            6514.
## 3  2011 Aston          78000
## 4  2011 Audi           10075.
## 5  2011 Bentley        99800
## 6  2011 BMW             9765.
```

```r
#Visualize the Price Trend Over Time
# Plot the price trends for different brands
ggplot(price_trend, aes(x = year, y = Average_Price, color = Brand, group = Brand)) +
  geom_point(show.legend = FALSE) +
  theme_minimal() +
  labs(title = "Average Car Prices Over Time by Brand",
       x = "year",
       y = "Average Price (€)",
       color = "Car Brand") +
  theme(legend.position = "right")
```




```r
####
```

```r
#Best Fuel Type for Cost Efficiency
#comparing fuel types based on car price and performance

# Calculating average price by fuel type
fuel_price <- df %>%
  group_by(fuel) %>%
  summarise(Average_Price = mean(price, na.rm = TRUE))

# Creating the plot with enhancements
ggplot(fuel_price, aes(x = reorder(fuel, -Average_Price), y = Average_Price, fill = fuel)) +
  geom_bar(stat = "identity", color = "black", alpha = 0.8,show.legend = FALSE) +
  geom_text(aes(label = paste0("€", round(Average_Price, 2))),
            vjust = -0.5, size = 2, color = "black") +
  scale_fill_brewer(palette = "Set3") +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold", size = 16),
    axis.title.x = element_text(face = "bold", size = 14),
    axis.title.y = element_text(face = "bold", size = 14),
    axis.text.x = element_text(angle = 45, hjust = 1),
  ) +
  labs(
    title = "Average Car Price by Fuel Type",
    x = "Fuel Type",
    y = "Average Price (€)"
  ) +
  scale_y_continuous(labels = scales::comma)
```




```
#####



#Horsepower vs. Car Price
# Check for missing values in 'hp' and 'price'
sum(is.na(df$hp))    # Check for missing values in 'hp'
```

```
## [1] 24
```

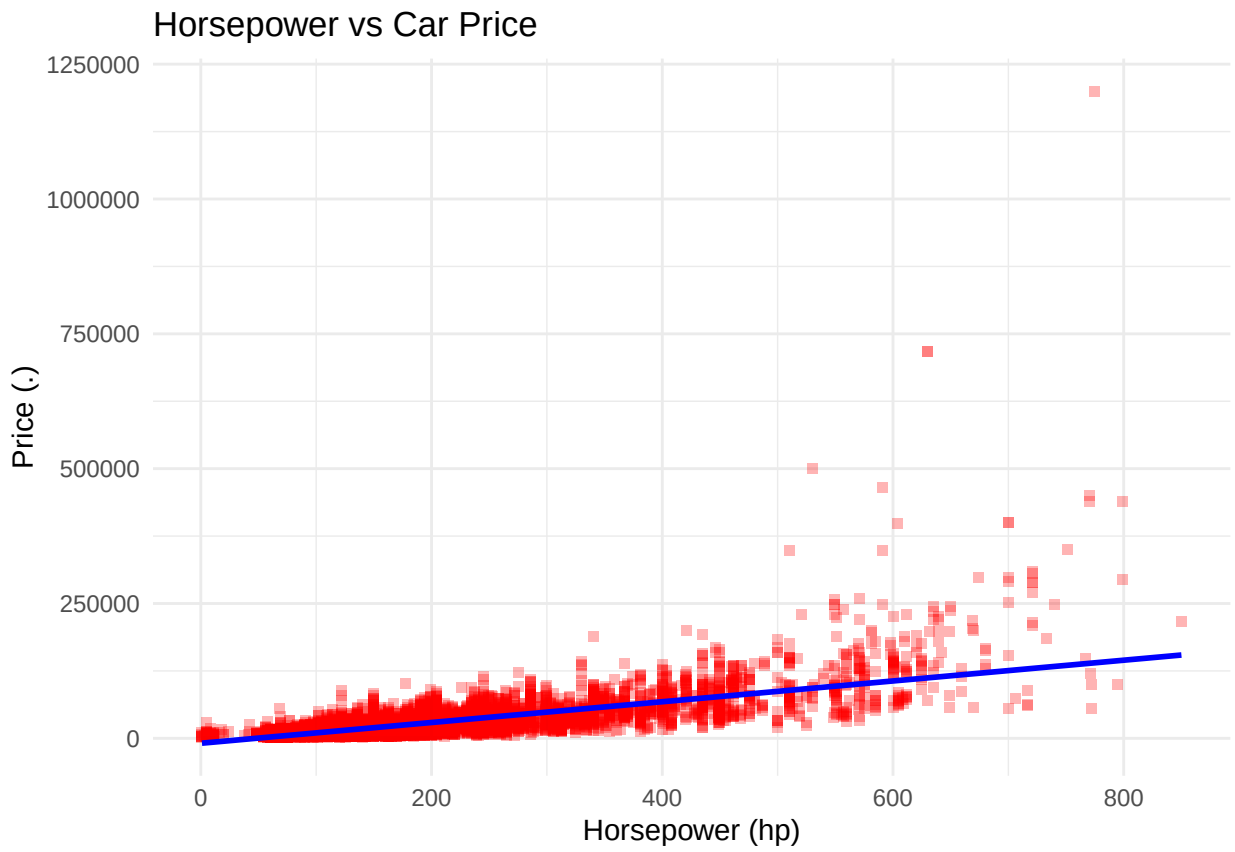
```
sum(is.na(df$price)) # Check for missing values in 'price'
```

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

```
# Remove rows with missing values in 'hp' or 'price'
df_clean <- df %>% filter(!is.na(hp)) %>% filter(!is.na(price))

# Create the scatter plot
ggplot(df_clean, aes(x = hp, y = price)) +
  geom_point(alpha = 0.3, color = "red", shape = "square") +
  geom_smooth(method = "lm", se = FALSE, color = "blue") +
  theme_minimal() +
  labs(title = "Horsepower vs Car Price",
```

```
       x = "Horsepower (hp)",
       y = "Price (€)")
```

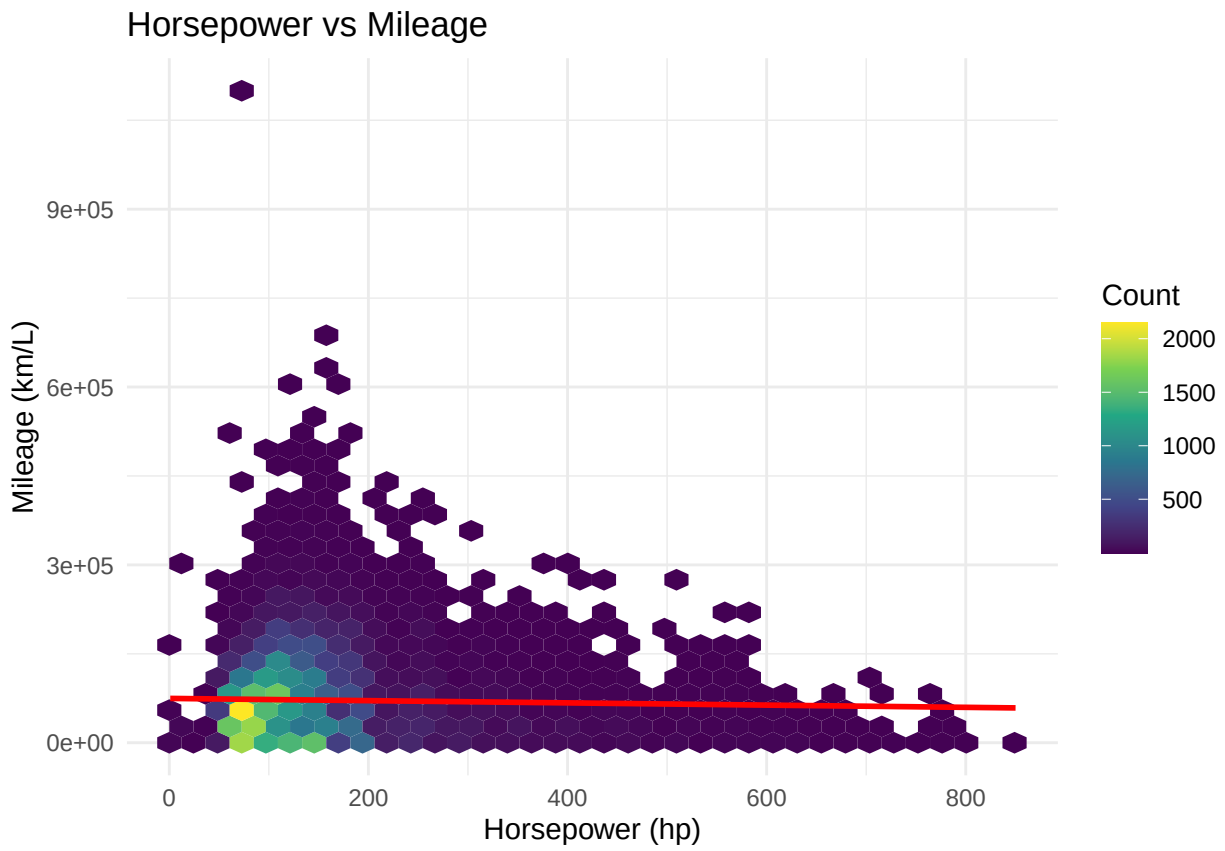
```
## `geom_smooth()` using formula = 'y ~ x'
```




```
####



#Fuel Efficiency vs. Horsepower
#Are fuel-efficient cars less powerful?

ggplot(df, aes(x = hp, y = mileage)) +
  geom_hex(bins = 35) +  # Adjust the number of bins as needed
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  scale_fill_viridis_c() +  # Use a color scale for better visualization
  theme_minimal() +
  labs(title = "Horsepower vs Mileage",
       x = "Horsepower (hp)",
       y = "Mileage (km/L)",
       fill = "Count")
```

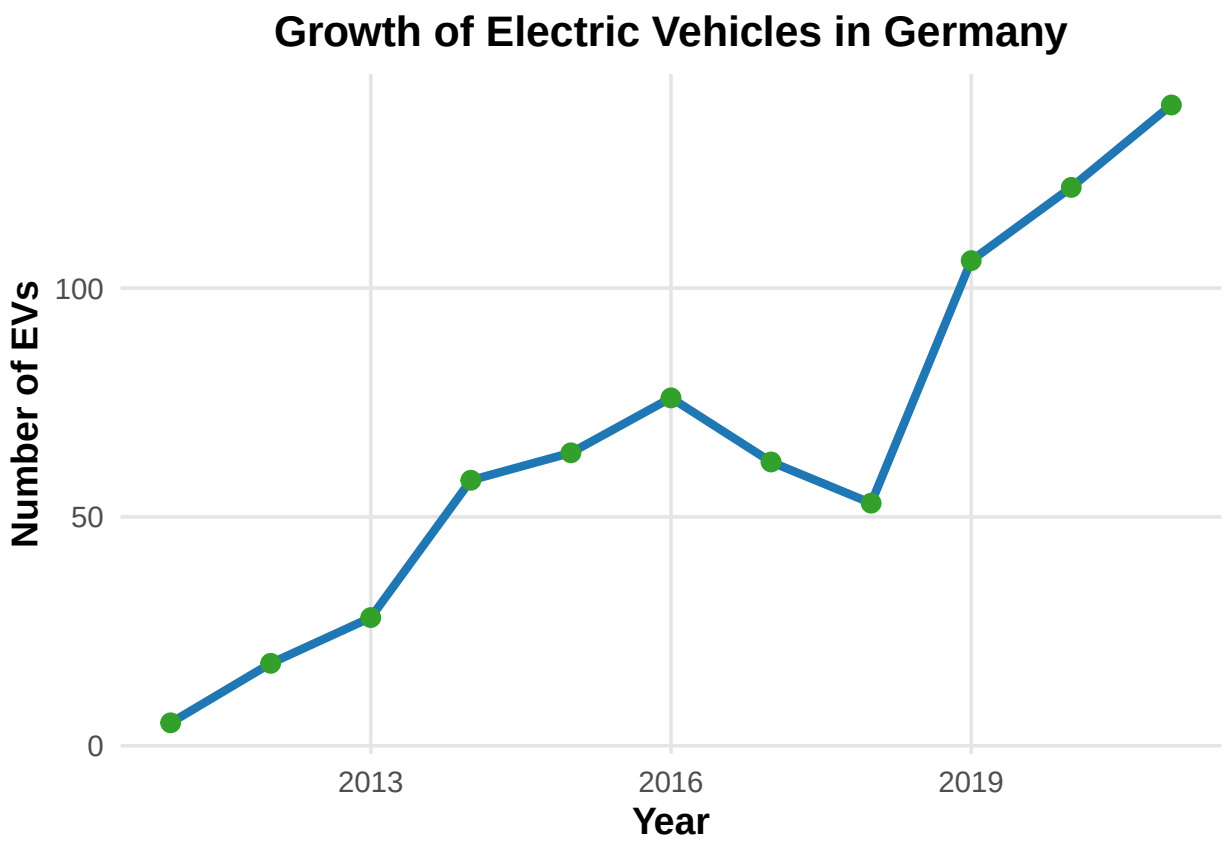
```
## Warning: Removed 24 rows containing non-finite outside the scale range
## (`stat_binhex()`).
```

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

## Warning: Removed 24 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```




```
####



#EV Adoption & Growth
#Analyze how the number of EVs has changed over time

# Filter for EVs and count per year
ev_trend <- df %>%
  filter(Fuel_Type == "Electric") %>%
  group_by(year) %>%
  summarise(EV_Count = n())

# Plot EV growth over years with enhancements
ggplot(ev_trend, aes(x = year, y = EV_Count)) +
  geom_line(color = "#1f78b4", size = 1.5) +
  geom_point(color = "#33a02c", size = 3) +
  theme_minimal(base_size = 14) +
  theme(
```

```
    plot.title = element_text(hjust = 0.5, face = "bold", size = 16),
    axis.title.x = element_text(face = "bold", size = 14),
    axis.title.y = element_text(face = "bold", size = 14),
    panel.grid.major = element_line(color = "gray90"),
    panel.grid.minor = element_blank()
  ) +
  labs(
    title = "Growth of Electric Vehicles in Germany",
    x = "Year",
    y = "Number of EVs"
  )
```




```
####



#Correlation Between Car Price and Performance
#Find if expensive cars have better fuel efficiency, horsepower, or safety ratings

# Correlation heatmap
library(corrplot)
```

```
## corrplot 0.95 loaded
```

```r
# Selecting numeric columns and calculating correlations
numeric_cols <- df %>%
  select_if(is.numeric) %>%
  cor(use = "complete.obs")  # Using pairwise complete observations

# Custom color palette
col <- colorRampPalette(c("#BB4444", "#EE9988", "#FFFFFF", "#77AADD", "#4477AA"))

# Plotting correlation heatmap with enhancements
corrplot(numeric_cols,
         method = "color",
         type = "upper",
         tl.cex = 0.9,
         tl.col = "black",
         tl.srt = 45,
         number.cex = 0.8,
         addCoef.col = "black",
         col = col(200),
         diag = FALSE,
         cl.pos = "r",
         cl.ratio = 0.2,
         mar = c(0, 0, 1, 0))
```


```r
####
```