mda9159_final_project

Jinminli-100826436

2025-12-08

```
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   3.5.0     v tibble    3.2.1
## v lubridate 1.9.3     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 error
```

```
library(dplyr)
library(knitr)
library(stringr)
library(purrr)
library(rlang)
```

```
##
## Attaching package: 'rlang'
##
## The following objects are masked from 'package:purrr':
##
##     %@%, flatten, flatten_chr, flatten_dbl, flatten_int, flatten_lgl,
##     flatten_raw, invoke, splice
```

```
library(ggplot2)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
##
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
car <- read.csv("serbia_car_sales_price_2024.csv") #load dataset
# change horsepower into numeric values (unit: HP)
car$horsepower <- as.numeric(gsub(" HP.*", "", car$horsepower))
```

EDA for Raw Data

```
num_vars_0 <- c("price", "views", "favorite", "year",
  "seats_amount", "horsepower",
  "car_mileage..km", "engine_capacity..cc"
)

num_summary <- data.frame(
  variable = num_vars_0,
  n        = sapply(car[num_vars_0], function(x) sum(!is.na(x))),
  mean     = sapply(car[num_vars_0], function(x) mean(x, na.rm = TRUE)),
  sd       = sapply(car[num_vars_0], function(x) sd(x, na.rm = TRUE)),
  median   = sapply(car[num_vars_0], function(x) median(x, na.rm = TRUE)),
  min      = sapply(car[num_vars_0], function(x) min(x, na.rm = TRUE)),
  max      = sapply(car[num_vars_0], function(x) max(x, na.rm = TRUE))
)

knitr::kable(
  num_summary,
  digits  = 1,
  caption = "Summary statistics for numeric variables in the raw data"
)
```

Table 1: Summary statistics for numeric variables in the raw data

	variable	n	mean	sd	median	min	max
price	price	8413	4848.2	5631.9	3300	100	82000
views	views	8413	308.7	847.4	114	0	27770
favorite	favorite	8413	2.7	5.1	1	0	151
year	year	8413	2006.1	6.8	2006	1960	2024
seats_amount	seats_amount	8403	4.9	0.7	5	2	9
horsepower	horsepower	8403	115.4	49.3	109	1	900
car_mileage..km	car_mileage..km	8404	2851955.0	104754099.9	220000	1	4294967295
engine_capacity..cc	engine_capacity..cc	8403	1725.2	524.3	1700	100	10000

```
cat_table <- function(x, var_name) {
  tab <- table(x, useNA = "ifany")
  out <- data.frame(
    level   = names(tab),
    count   = as.vector(tab),
    percent = round(100 * tab / sum(tab), 1)
  )
  knitr::kable(
    out,
    caption = paste0("Distribution of ", var_name)
  )
}

cat_table(car$fuel,          "fuel")
```

Table 2: Distribution of fuel

level	count	percent.x	percent.Freq
	9		0.1
diesel	4797	diesel	57.0
Diesel + gas	6	Diesel + gas	0.1
electric	12	electric	0.1
gas	6	gas	0.1
hybrid	39	hybrid	0.5
methane	106	methane	1.3
petrol	2757	petrol	32.8
petrol + gas	681	petrol + gas	8.1

```
cat_table(car$car_type,       "car_type")
```

Table 3: Distribution of car_type

level	count	percent.x	percent.Freq
	8		0.1
cabriolet	123	cabriolet	1.5
caravan	1512	caravan	18.0
coupe	350	coupe	4.2
hatchback	2792	hatchback	33.2
limousine	1912	limousine	22.7
minivan (MPV)	683	minivan (MPV)	8.1
pickup	196	pickup	2.3
suv	837	suv	9.9

```
cat_table(car$type_of_drive, "type_of_drive")
```

Table 4: Distribution of type_of_drive

level	count	percent.x	percent.Freq
	9		0.1
4x4	907	4x4	10.8
back	784	back	9.3
front	6713	front	79.8

```
cat_table(car$gearbox,        "gearbox")
```

Table 5: Distribution of gearbox

level	count	percent.x	percent.Freq
	10		0.1
automatic	1164	automatic	13.8
manual, 4 speeds	168	manual, 4 speeds	2.0
manual, 5 speeds	4645	manual, 5 speeds	55.2
manual, 6 speeds	2323	manual, 6 speeds	27.6
manual, multiple speeds	18	manual, multiple speeds	0.2

level	count	percent.x	percent.Freq
semi-automatic	85	semi-automatic	1.0

```
cat_table(car$doors,          "doors")
```

Table 6: Distribution of doors

level	count	percent.x	percent.Freq
	10		0.1
2/3 doors	1556	2/3 doors	18.5
4/5 doors	6847	4/5 doors	81.4

```
cat_table(car$A.C,            "A.C")
```

Table 7: Distribution of A.C

level	count	percent.x	percent.Freq
	10		0.1
automatic A/C	3775	automatic A/C	44.9
manual A/C	3175	manual A/C	37.7
no A/C	1453	no A/C	17.3

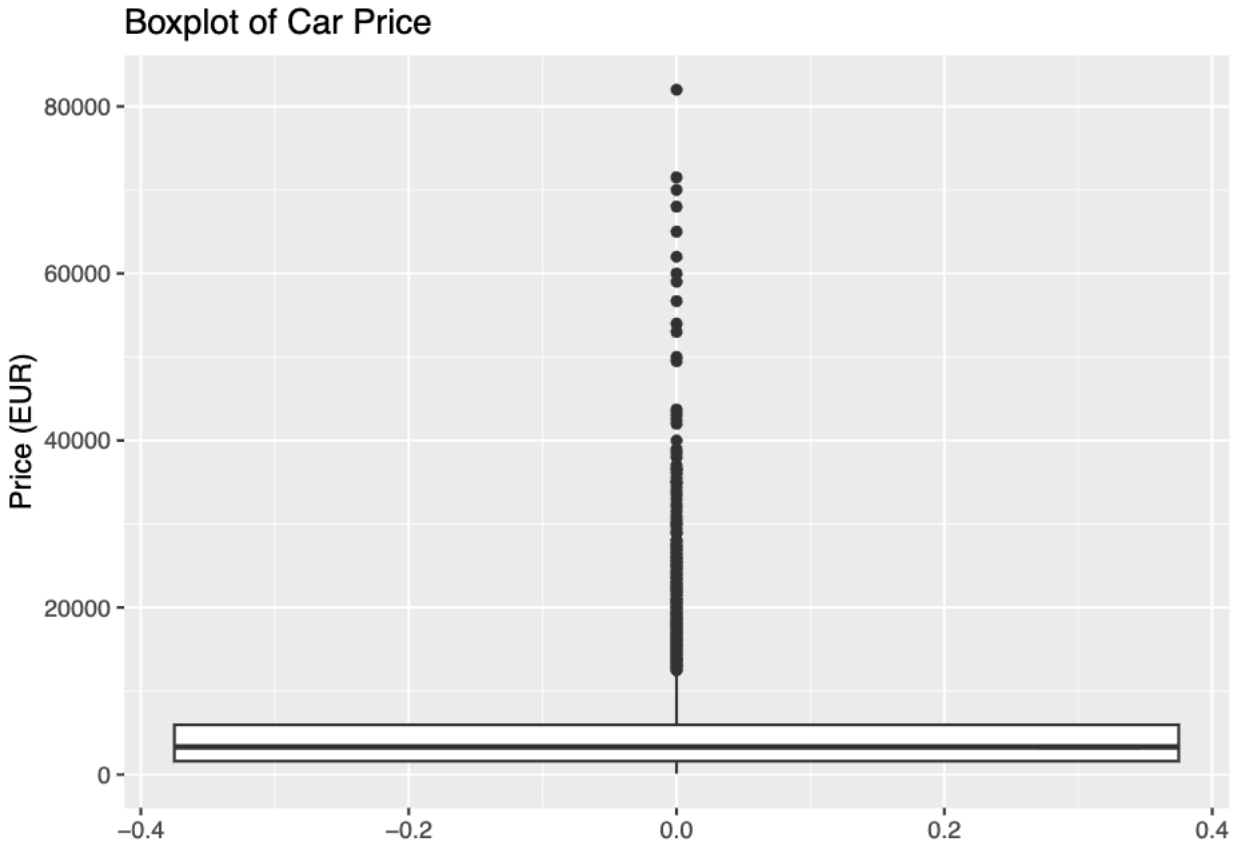
```
cat_table(car$emission_class,"emission_class")
```

Table 8: Distribution of emission_class

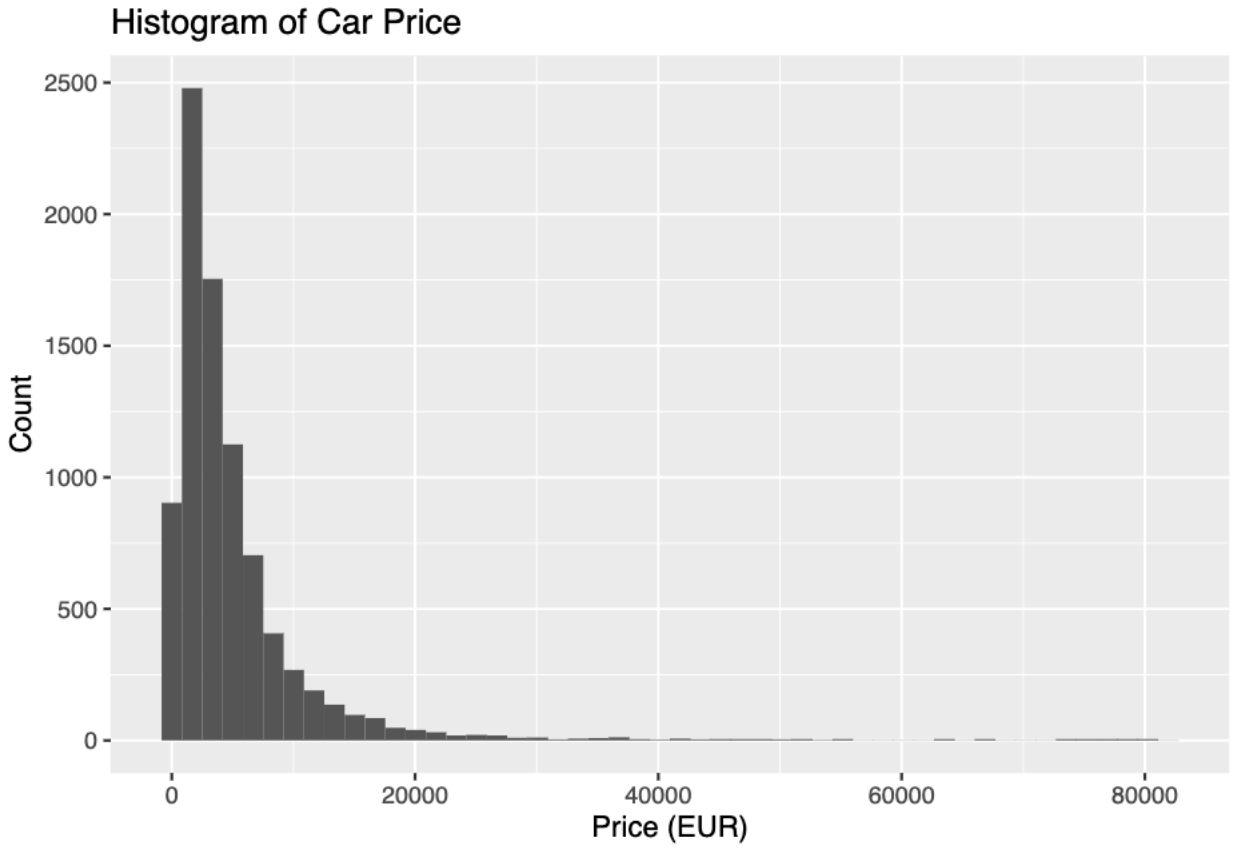
level	count	percent.x	percent.Freq
	1337		15.9
Euro 1	218	Euro 1	2.6
Euro 2	343	Euro 2	4.1
Euro 3	1829	Euro 3	21.7
Euro 4	2508	Euro 4	29.8
Euro 5	1458	Euro 5	17.3
Euro 6	720	Euro 6	8.6

Response

```
# Boxplot
p_price_box <- ggplot(car, aes(y = price)) +
  geom_boxplot() +
  labs(
    title = "Boxplot of Car Price",
    y = "Price (EUR)"
  )
p_price_box
```



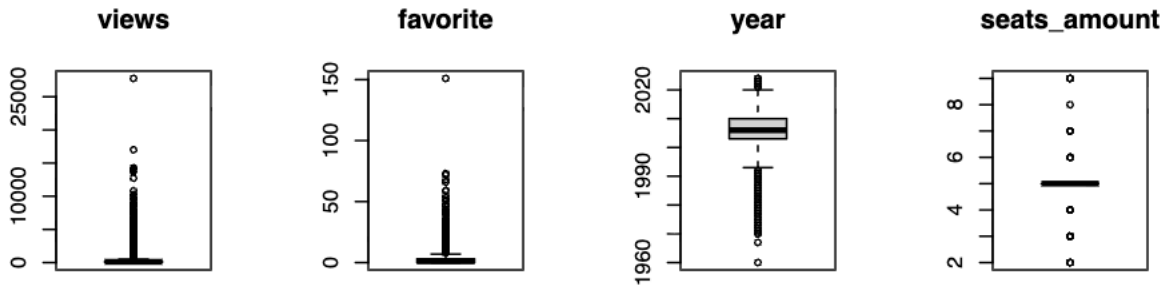

```
ggsave("mda9159_jinmin_files/price_boxplot.png", p_price_box, width = 6, height = 4)

# Histogram
p_price_hist <- ggplot(car, aes(x = price)) +
  geom_histogram(bins = 50) +
  labs(
    title = "Histogram of Car Price",
    x = "Price (EUR)",
    y = "Count"
  )
p_price_hist
```

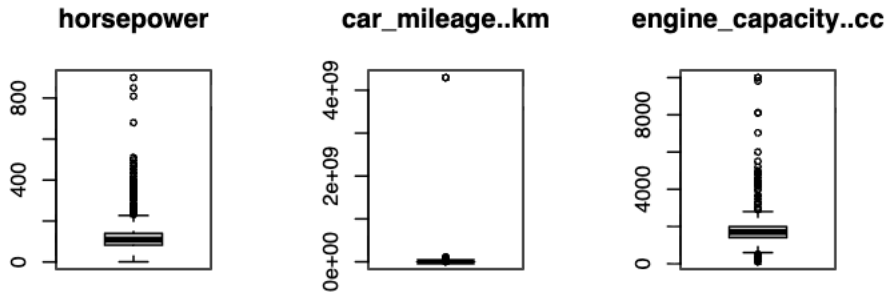



```
ggsave("mda9159_jinmin_files/price_histogram.png", p_price_hist, width = 6, height = 4)
```

Numerical

```
num_vars <- c(
  "views",
  "favorite",
  "year",
  "seats_amount",
  "horsepower",
  "car_mileage..km",
  "engine_capacity..cc"
)


par(mfrow = c(2, 4))
for (v in num_vars) {
  boxplot(car[[v]],
          main = v,
          ylab = "")
}
par(mfrow = c(1, 1))
```





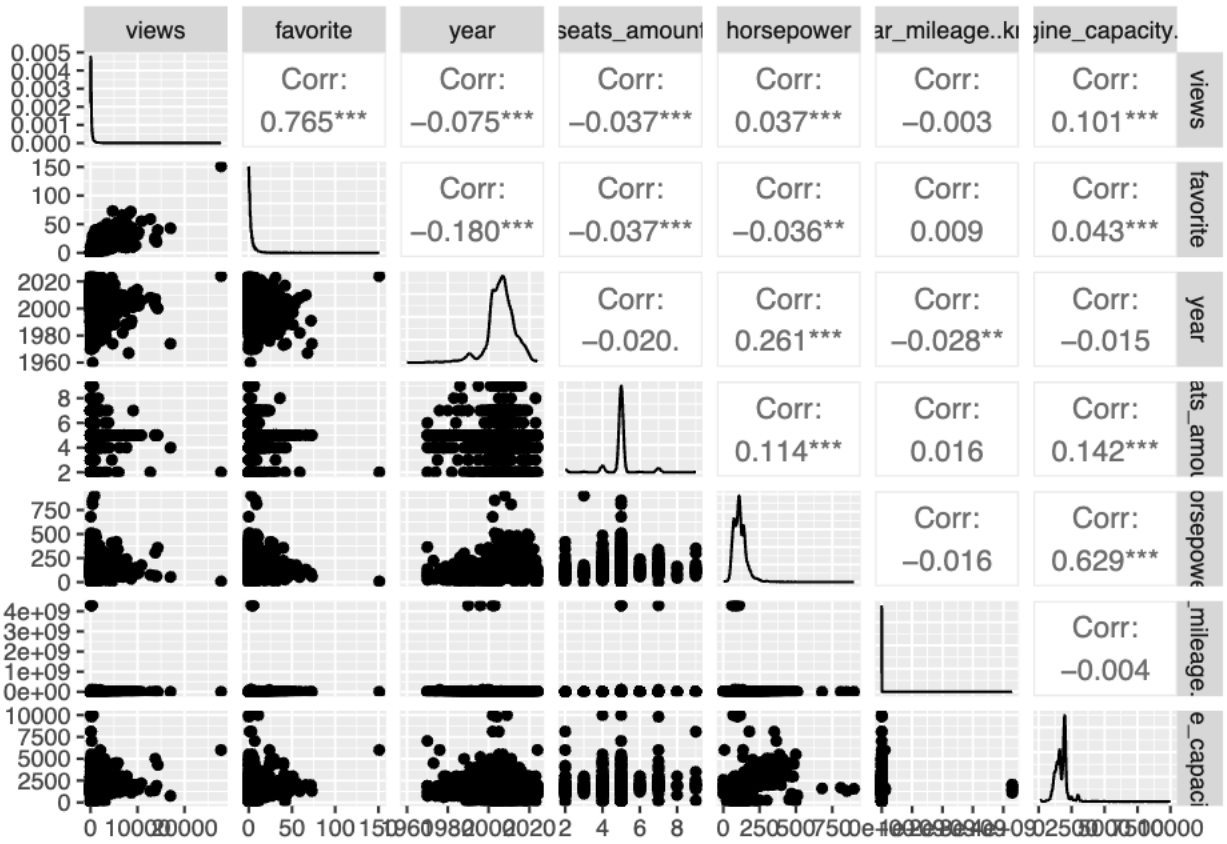

```
png("mda9159_jinmin_files/numeric_boxplots.png", width = 1200, height = 800)
par(mfrow = c(2, 4))
for (v in num_vars) {
  boxplot(car[[v]],
          main = v,
          ylab = "")
}
par(mfrow = c(1, 1))
dev.off()
```

```
## pdf
##   2
```

```
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

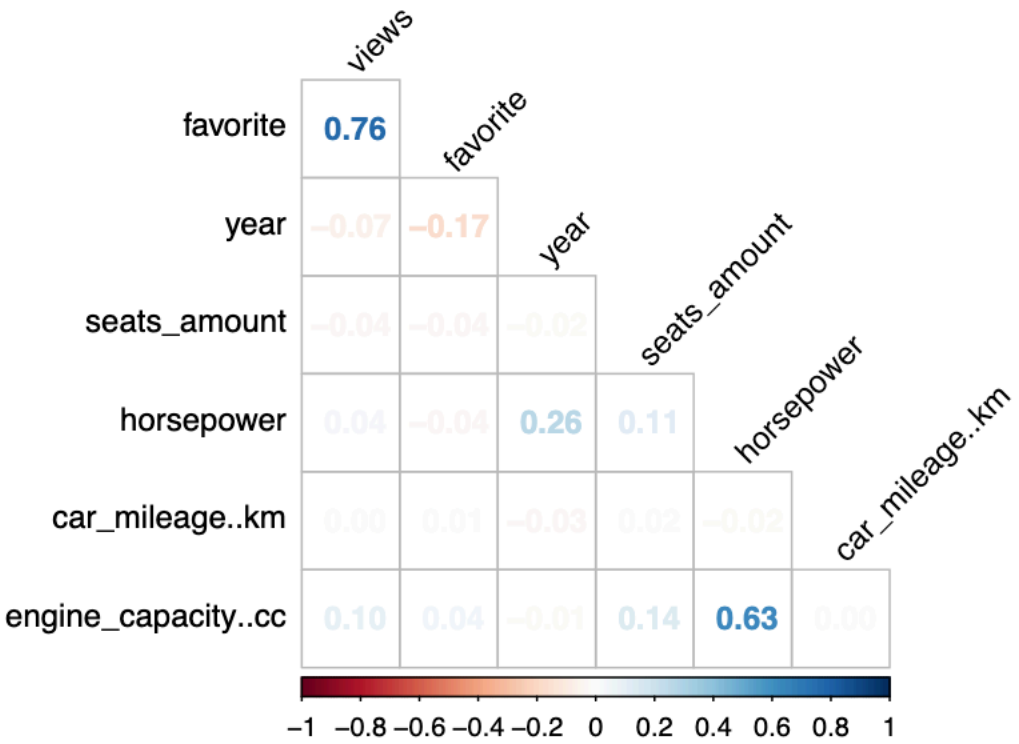
```
ggpairs(car[num_vars])
```




```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
#compute correlation matrix
cor_matrix <- cor(car[,num_vars], use = "complete.obs", method = "pearson")

#create correlation matrix plot
corrplot(cor_matrix, method = "number", type = "lower",
         diag = FALSE,  tl.col="black", tl.srt=45)
```



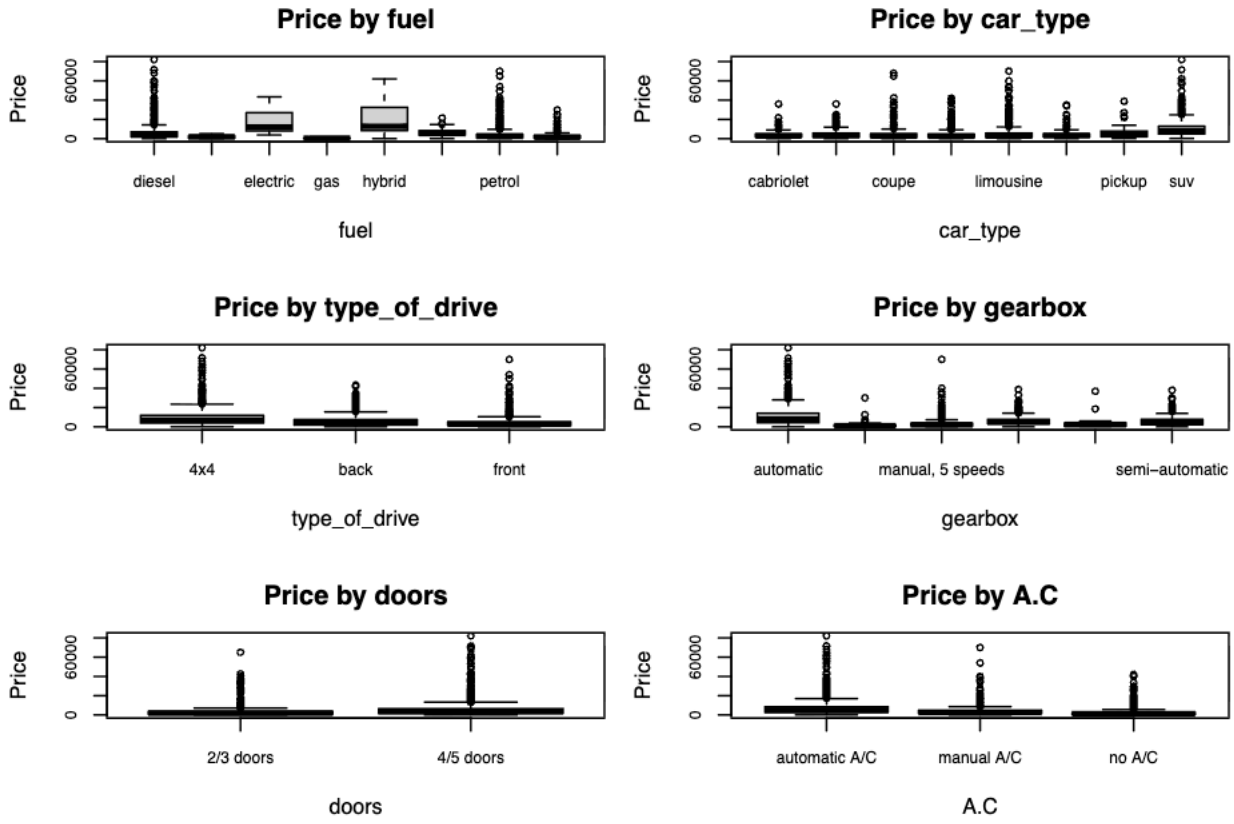

```
png("mda9159_jinmin_files/correlation_matrix.png", width = 900, height = 900)

corrplot(
  cor_matrix,
  method = "number",
  type = "lower",
  diag = FALSE,
  tl.col = "black",
  tl.srt = 45
)

dev.off()
```

```
## pdf
##   2
```

Categorical

```
cat_vars <- c("fuel", "car_type", "type_of_drive",
              "gearbox", "doors", "A.C")


par(mfrow = c(3, 2), mar = c(5, 4, 3, 1))

for (v in cat_vars) {
  x  <- car[[v]]
  ok <- x != "" & !is.na(x) & !is.na(car$price)

  boxplot(car$price[ok] ~ x[ok],
          main = paste("Price by", v),
```

```
          xlab = v,
          ylab = "Price",
          cex.axis = 0.7)
}
```




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

```
png("mda9159_jinmin_files/categorical_price_boxplots.png", width = 1200, height = 800)
par(mfrow = c(2, 3), mar = c(5, 4, 3, 1))
for (v in cat_vars) {
  x  <- car[[v]]
  ok <- x != "" & !is.na(x) & !is.na(car$price)
  boxplot(car$price[ok] ~ x[ok],
          main = paste("Price by", v),
          xlab = v,
          ylab = "Price",
          cex.axis = 0.7)
}
par(mfrow = c(1, 1))
dev.off()
```

```
## pdf
##   2
```

Data Cleaning

```r
car_data_cleaned <- car
# car name->car brand
car_data_cleaned$car_brand <- sub(" .*", "", car$car_name)
car_data_cleaned$car_name <- NULL
```

```r
remove_outliers <- function(data, column) {
  Q1 <- quantile(data[[column]], 0.1, na.rm = TRUE)
  Q3 <- quantile(data[[column]], 0.9, na.rm = TRUE)
  IQR <- Q3 - Q1
  lower_bound <- Q1 - 1.5 * IQR
  upper_bound <- Q3 + 1.5 * IQR
  data %>% filter(data[[column]] >= lower_bound & data[[column]] <= upper_bound)
}

car_data_cleaned <- remove_outliers(car_data_cleaned, "price")

# clean the dataset for the first time
car_data_cleaned= car_data_cleaned %>% filter(car_mileage..km < 5*10^5, favorite < 50,
                              horsepower < 400, year >= 1970)
```

```r
# car type
car_data_cleaned <- car_data_cleaned %>%
  mutate(
    car_type = case_when(
      car_type %in% c("limousine") ~ "Limousine",
      car_type %in% c("minivan (MPV)", "caravan") ~ "Family",
      car_type %in% c("pickup") ~ "Pickup",
      car_type %in% c("hatchback", "coupe") ~ "Economy",
      car_type %in% c("cabriolet") ~ "Sport",
      car_type %in% c("suv") ~ "SUV",
      TRUE ~ "Other"
    )
  )

car_data_cleaned <- car_data_cleaned[car_data_cleaned$emission_class != "", ]

# emission class
car_data_cleaned <- car_data_cleaned %>%
  mutate(
    emission_type = case_when(
      emission_class %in% c("Euro 6") ~ "Large Displacement",
      emission_class %in% c("Euro 1","Euro 2","Euro 3","Euro 4","Euro 5") ~ "Small Displacement"
    )
  )
car_data_cleaned$emission_class <- NULL

# door number
car_data_cleaned$is_2_3_doors <- ifelse(car_data_cleaned$doors == "2/3 doors", TRUE, FALSE)
car_data_cleaned$doors <- NULL

# Fuel
car_data_cleaned <- car_data_cleaned %>%
  mutate(
    fuel_type = case_when(
```

```
      fuel %in% c("petrol", "diesel", "Diesel + gas", "petrol + gas", "gas") ~ "Oil",
      fuel %in% c("electric") ~ "Electric",
      fuel %in% c("hybrid", "methane", "cng", "lpg") ~ "Mixture",
      TRUE ~ "Unknown"
    )
    )

car_data_cleaned$fuel <- NULL

# brand
car_data_cleaned <- car_data_cleaned %>%
  mutate(
    car_brand_category = case_when(
      car_brand %in% c("Audi", "BMW", "Mercedes", "Lexus", "Porsche", "Jaguar", "Land Rover", "Volvo") ~
      car_brand %in% c("Ford", "Honda", "Hyundai", "Mazda", "Nissan", "Toyota", "Volkswagen", "Skoda",
                       "Opel", "Mitsubishi", "Chevrolet", "Kia", "Subaru", "Seat") ~ "Standard",
      car_brand %in% c("Fiat", "Dacia", "Suzuki", "Citroen", "Peugeot", "Renault", "Daewoo", "Daihatsu"
                       "Lada", "Zastava") ~ "Budget",
      car_brand %in% c("Jeep", "Alfa","Dodge", "Chrysler", "Rover", "MG", "Isuzu", "Iveco", "SsangYong"
                       "Saab", "UAZ") ~ "Other",
      TRUE ~ "Unknown"
    )
  )
car_data_cleaned$car_brand <- NULL

# gearbox_group
car_data_cleaned <- car_data_cleaned %>%
  mutate(
    gearbox_group = case_when(
      grepl("automatic", gearbox) ~ "automatic",
      grepl("manual",    gearbox) ~ "manual",
      TRUE ~ "other"
    )
  )
car_data_cleaned$gearbox <- NULL
```

filter dataset

```
# clean the dataset for the second time
car_data_cleaned <- car_data_cleaned %>%
  filter(
    car_mileage..km < 5*10^5,
    favorite       < 30,
    horsepower     < 300,
    year           >= 2000,
    car_brand_category != "Unknown"
  )
```

post days

```
# post days
car_data_cleaned <- car_data_cleaned |>
  mutate(
```

```
    post_days = case_when(
      # Today = 0
      str_detect(post_info, "today") ~ 0,

      # Yesterday = 1 day ago
      str_detect(post_info, "yesterday") ~ 1,
      # a week = 7
      str_detect(post_info, "a week") ~ 7,
      # x weeks = 7x
      str_detect(post_info, "weeks") ~ as.numeric(str_extract(post_info, "\\d+")) * 7,
      # month appx= 30
      str_detect(post_info, "month") ~ 30,
      # x days = x
      str_detect(post_info, "days") ~ as.numeric(str_extract(post_info, "\\d+")),
      TRUE ~ NA_real_
    )
  )

# check null value
# sum(is.na(car_data_cleaned$post_days))
# drop post info
car_data_cleaned$post_info <- NULL
```

```
colSums(is.na(car_data_cleaned))
```

```
##               views            favorite               price                year
##                   0                   0                   0                   0
##                 A.C        seats_amount          horsepower               color
##                   0                   0                   0                   0
##     car_mileage..km engine_capacity..cc       type_of_drive            car_type
##                   0                   0                   0                   0
##       emission_type        is_2_3_doors           fuel_type  car_brand_category
##                   0                   0                   0                   0
##       gearbox_group           post_days
##                   0                   0
```

```
head(car_data_cleaned)
```

```
##   views favorite price year           A.C seats_amount horsepower color
## 1    55        0  1100 2000    manual A/C            5        106  gray
## 2   209        1   500 2000    manual A/C            5        120  gray
## 3   133        1  3500 2008 automatic A/C            5        170 black
## 4   290        6  2150 2008    manual A/C            5         90 black
## 5  2629       16  1350 2008 automatic A/C            5        120 white
## 6    89        0  2000 2007        no A/C            5        105 black
##   car_mileage..km engine_capacity..cc type_of_drive  car_type
## 1          158546               1400         front   Economy
## 2          200000               1600         front Limousine
## 3          255000               1929         front   Economy
## 4          214000               1600         front   Economy
## 5          315000               1910         front   Economy
## 6          176853               1597         front   Economy
##        emission_type is_2_3_doors fuel_type car_brand_category gearbox_group
## 1 Small Displacement         TRUE       Oil              Other        manual
## 2 Small Displacement        FALSE       Oil              Other        manual
```

```
## 3 Small Displacement        FALSE       Oil             Other        manual
## 4 Small Displacement        FALSE       Oil             Other        manual
## 5 Small Displacement        FALSE       Oil             Other        manual
## 6 Small Displacement        FALSE       Oil             Other        manual
##   post_days
## 1         2
## 2        28
## 3         5
## 4         6
## 5         7
## 6         7
```

Exploratory Data Analysis

Summary statistics & tables

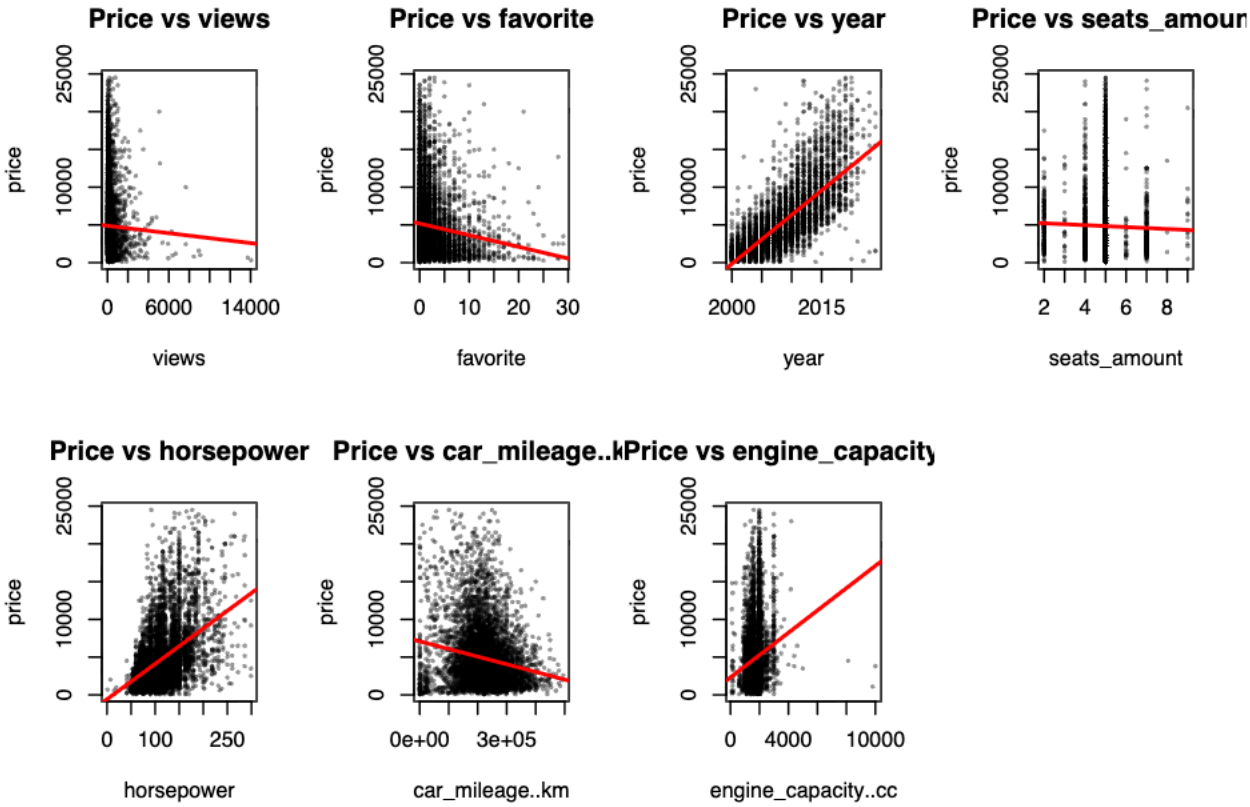
```r
num_summary <- data.frame(
  variable = num_vars,
  mean   = sapply(car_data_cleaned[num_vars], mean,   na.rm = TRUE),
  sd     = sapply(car_data_cleaned[num_vars], sd,     na.rm = TRUE),
  median = sapply(car_data_cleaned[num_vars], median, na.rm = TRUE),
  min    = sapply(car_data_cleaned[num_vars], min,    na.rm = TRUE),
  max    = sapply(car_data_cleaned[num_vars], max,    na.rm = TRUE)
)

knitr::kable(
  num_summary,
  digits  = 2,
  caption = "Summary statistics for numeric variables"
)
```

Table 9: Summary statistics for numeric variables

	variable	mean	sd	median	min	max
views	views	245.61	562.29	103	0	14040
favorite	favorite	2.16	3.30	1	0	29
year	year	2007.71	4.78	2007	2000	2024
seats_amount	seats_amount	4.95	0.73	5	2	9
horsepower	horsepower	115.73	38.38	110	1	299
car_mileage..km	car_mileage..km	221656.25	75460.15	220000	1	495000
engine_capacity..cc	engine_capacity..cc	1709.55	447.27	1698	100	10000

```r
par(mfrow = c(2, 4))
for (v in num_vars) {
  plot(car_data_cleaned[[v]], car_data_cleaned$price,
       xlab = v,
       ylab = "price",
       main = paste("Price vs", v),
       pch = 16, cex = 0.5, col = rgb(0,0,0,0.4))

  abline(lm(price ~ car_data_cleaned[[v]], data = car_data_cleaned),
         col = "red", lwd = 2)
```

```
}
par(mfrow = c(1, 1))
```




```
png("mda9159_jinmin_files/scatterplots_price.png", width = 1200, height = 800)
par(mfrow = c(2, 4))
for (v in num_vars) {
  plot(car[[v]], car$price,
       xlab = v,
       ylab = "price",
       main = paste("Price vs", v),
       pch = 16, cex = 0.5, col = rgb(0,0,0,0.4))
  abline(lm(price ~ car[[v]], data = car), col = "red", lwd = 2)
}
par(mfrow = c(1, 1))
dev.off()
```

```
## pdf
##   2
```

```
cat_vars <- c(
  "A.C", "type_of_drive",
  "gearbox_group", "car_brand_category", "car_type",
  "emission_type", "fuel_type"
)

cat_summary <- purrr::map_dfr(cat_vars, function(v) {
  car_data_cleaned %>%
    dplyr::filter(!is.na(.data[[v]]), .data[[v]] != "") %>%
    dplyr::count(level = .data[[v]]) %>%
```

```
    dplyr::mutate(
      variable = v,
      percent  = 100 * n / sum(n)
    ) %>%
    dplyr::relocate(variable)
})

knitr::kable(
  cat_summary,
  digits  = 1,
  caption = "Frequency and percentage for categorical variables"
)
```

Table 10: Frequency and percentage for categorical variables

variable	level	n	percent
A.C	automatic A/C	3128	50.5
A.C	manual A/C	2561	41.4
A.C	no A/C	501	8.1
type_of_drive	4x4	576	9.3
type_of_drive	back	505	8.2
type_of_drive	front	5109	82.5
gearbox_group	automatic	901	14.6
gearbox_group	manual	5289	85.4
car_brand_category	Budget	1990	32.1
car_brand_category	Luxury	1248	20.2
car_brand_category	Other	213	3.4
car_brand_category	Standard	2739	44.2
car_type	Economy	2350	38.0
car_type	Family	1786	28.9
car_type	Limousine	1241	20.0
car_type	Pickup	152	2.5
car_type	SUV	573	9.3
car_type	Sport	88	1.4
emission_type	Large Displacement	635	10.3
emission_type	Small Displacement	5555	89.7
fuel_type	Electric	5	0.1
fuel_type	Mixture	114	1.8
fuel_type	Oil	6071	98.1

```
# dummy
tibble(
  is_2_3_doors_count = sum(car_data_cleaned$is_2_3_doors, na.rm = TRUE)
)
```

```
## # A tibble: 1 x 1
##   is_2_3_doors_count
##                <int>
## 1               1017
```

Summary of Variables in `car_data_cleaned`

1. Numeric Variables

- **views**: Number of times the listing was viewed.
- **favorite**: Number of users who added the car to their favorites.
- **price**: Listing price (in euros).
- **year**: Vehicle manufacturing year.
- **seats_amount**: Number of seats in the vehicle.
- **horsepower**: Engine horsepower output.
- **car_mileage..km**: Total mileage of the car in kilometers.
- **engine_capacity..cc**: Engine displacement, measured in cubic centimeters (cc).
- **post_days**: Number of days the listing has been posted online.

2. Categorical Variables

- **A.C**: Type of air conditioning system (e.g., manual A/C, automatic A/C).
- **color**: Exterior color of the vehicle.
- **type_of_drive**: Drivetrain type (front-wheel, rear-wheel, all-wheel drive).
- **car_type**: Vehicle class (e.g., Economy, Limousine).
- **emission_type**: Emission classification based on engine design.
- **fuel_type**: Type of fuel used by the vehicle (e.g., gasoline, diesel, oil).
- **car_brand_category**: Grouped brand category assigned to each car.
- **gearbox_group**: Gearbox type (manual vs. automatic).

These variables help differentiate vehicles by design, environmental category, mechanical structure, and brand classification. ### **3. Binary / Logical Variable**

- **is_2_3_doors**: Logical indicator identifying whether the vehicle has 2 or 3 doors. # Model

```
car_model <- car_data_cleaned %>%
  dplyr::select(
    price,  # response
    views, favorite, year, seats_amount,
    horsepower, car_mileage..km, engine_capacity..cc, post_days,  # numerical
    A.C, type_of_drive,
    car_type, emission_type, fuel_type, car_brand_category, gearbox_group,  # categorical
    is_2_3_doors  # dummy
  ) %>%
  na.omit()

car_model <- car_model %>%
```

```
  mutate(
    A.C                = factor(A.C),
    type_of_drive      = factor(type_of_drive),
    car_type           = factor(car_type),
    emission_type      = factor(emission_type),
    fuel_type          = factor(fuel_type),
    car_brand_category = factor(car_brand_category),
    gearbox_group      = factor(gearbox_group)
  )
```

Baseline model

```
# Create log_price
car_model <- car_model %>%
  mutate(log_price = log(price))

# Full linear model (corrected)
full_model <- lm(
  log_price ~
    views + favorite + year + seats_amount +
    horsepower + car_mileage..km + engine_capacity..cc + post_days +
    A.C + type_of_drive +
    car_type + emission_type + fuel_type +
    car_brand_category + gearbox_group +
    is_2_3_doors,   # replacement for doors_num
  data = car_model
)

summary(full_model)
```

```
##
## Call:
## lm(formula = log_price ~ views + favorite + year + seats_amount +
##     horsepower + car_mileage..km + engine_capacity..cc + post_days +
##     A.C + type_of_drive + car_type + emission_type + fuel_type +
##     car_brand_category + gearbox_group + is_2_3_doors, data = car_model)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3830 -0.1647  0.0526  0.2399  1.9781
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  -2.413e+02  3.389e+00 -71.202  < 2e-16 ***
## views                         8.913e-06  1.286e-05   0.693 0.488339
## favorite                     -3.137e-02  2.211e-03 -14.187  < 2e-16 ***
## year                          1.243e-01  1.674e-03  74.214  < 2e-16 ***
## seats_amount                 -1.015e-02  9.292e-03  -1.092 0.274786
## horsepower                    3.746e-03  2.521e-04  14.858  < 2e-16 ***
## car_mileage..km               2.148e-07  8.497e-08   2.529 0.011476 *
## engine_capacity..cc          -1.054e-05  1.887e-05  -0.559 0.576492
## post_days                     1.194e-03  7.727e-04   1.546 0.122260
## A.Cmanual A/C                -7.472e-02  1.348e-02  -5.544 3.09e-08 ***
```

```
## A.Cno A/C                        -2.732e-01  2.406e-02 -11.354  < 2e-16 ***
## type_of_driveback                -7.226e-02  3.260e-02  -2.217 0.026684 *
## type_of_drivefront               -1.699e-01  2.615e-02  -6.496 8.87e-11 ***
## car_typeFamily                   -8.952e-02  1.589e-02  -5.633 1.85e-08 ***
## car_typeLimousine                -3.244e-03  1.729e-02  -0.188 0.851159
## car_typePickup                    1.841e-01  4.109e-02   4.479 7.62e-06 ***
## car_typeSport                     1.937e-01  5.049e-02   3.837 0.000126 ***
## car_typeSUV                       2.775e-01  2.679e-02  10.355  < 2e-16 ***
## emission_typeSmall Displacement   1.693e-02  2.338e-02   0.724 0.469180
## fuel_typeMixture                 -2.323e-01  2.031e-01  -1.144 0.252804
## fuel_typeOil                     -2.859e-01  1.993e-01  -1.434 0.151515
## car_brand_categoryLuxury          3.946e-01  2.082e-02  18.952  < 2e-16 ***
## car_brand_categoryOther          -5.906e-02  3.326e-02  -1.776 0.075786 .
## car_brand_categoryStandard        2.143e-01  1.347e-02  15.905  < 2e-16 ***
## gearbox_groupmanual              -7.825e-02  1.870e-02  -4.184 2.91e-05 ***
## is_2_3_doorsTRUE                 -3.971e-02  1.838e-02  -2.160 0.030811 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4414 on 6164 degrees of freedom
## Multiple R-squared:  0.7451, Adjusted R-squared:  0.7441
## F-statistic: 720.9 on 25 and 6164 DF,  p-value: < 2.2e-16
```

```
library(car)
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
## The following object is masked from 'package:purrr':
##
##     some
```

```
vif_values <- vif(full_model)

print(vif_values)
```

```
##                            GVIF Df GVIF^(1/(2*Df))
## views                5.04698506  1       2.2465496
## favorite             4.42198384  1       2.1028514
## year               -13.36274310  1             NaN
## seats_amount        -0.24800391  1             NaN
## horsepower          -0.08560515  1             NaN
## car_mileage..km     -2.18752780  1             NaN
## engine_capacity..cc  2.82291413  1       1.6801530
## post_days            0.48103838  1       0.6935693
## A.C                 -6.89859667  2             NaN
## type_of_drive        1.23319358  2       1.0537991
## car_type           -10.03585432  5             NaN
## emission_type      135.75479285  1      11.6513859
## fuel_type            0.06074004  2       0.4964423
## car_brand_category   0.05691272  3       0.6202045
```

```
## gearbox_group          0.99589216  1       0.9979440
## is_2_3_doors           1.43922526  1       1.1996771
```

From the full model, we observed that several variables: **views**, **seats_amount**, **engine_capacity..cc**, **post_days**, **emission_type**, **fuel_type**, **car_type (Limousine)**, and **car_brand_category (Other)** make only marginal contributions to explaining or predicting the log-transformed price. These predictors show weak statistical significance or minimal effect sizes, so it is reasonable to remove them when constructing the reduced model.

```
reduced_model <- lm(
  log_price ~
    favorite + year + horsepower + car_mileage..km +
     post_days +
    is_2_3_doors +
    A.C + type_of_drive + car_type +
    car_brand_category + gearbox_group,
  data = car_model
)

summary(reduced_model)
```

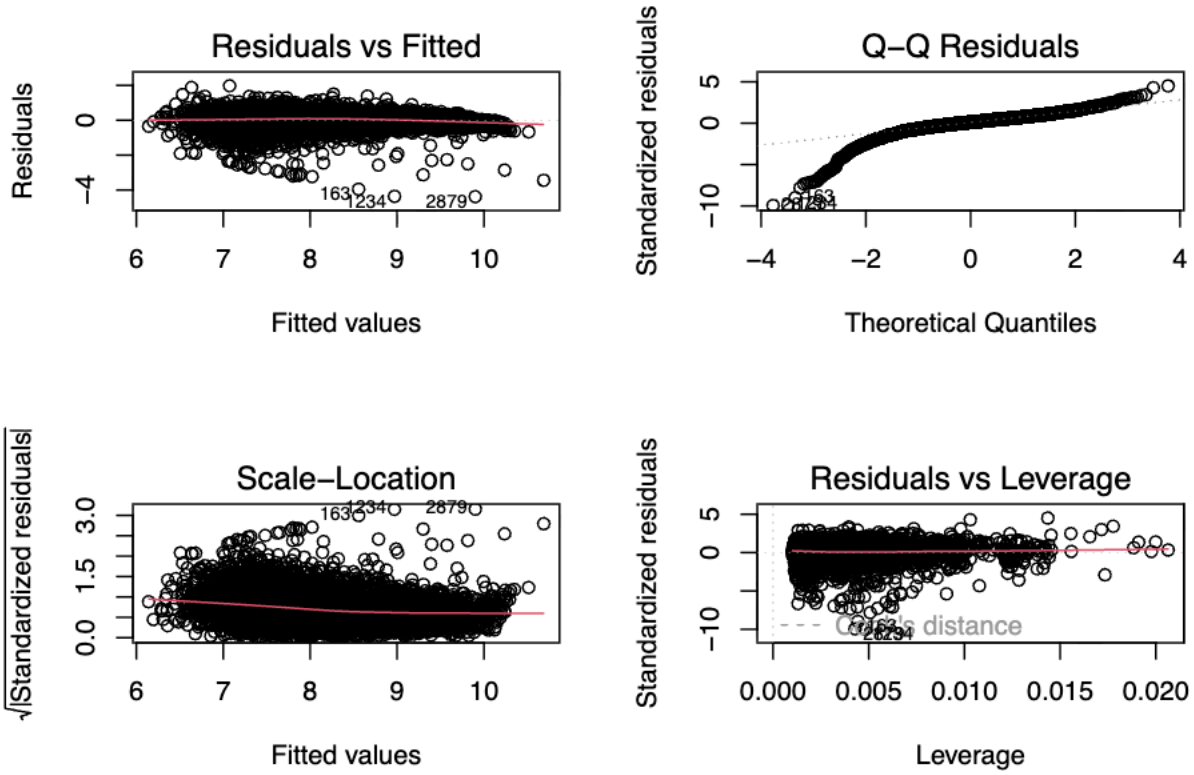
```
##
## Call:
## lm(formula = log_price ~ favorite + year + horsepower + car_mileage..km +
##     post_days + is_2_3_doors + A.C + type_of_drive + car_type +
##     car_brand_category + gearbox_group, data = car_model)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3791 -0.1652  0.0518  0.2410  1.9747
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                -2.417e+02  2.769e+00 -87.283  < 2e-16 ***
## favorite                   -3.044e-02  1.722e-03 -17.676  < 2e-16 ***
## year                        1.243e-01  1.376e-03  90.343  < 2e-16 ***
## horsepower                  3.635e-03  2.186e-04  16.628  < 2e-16 ***
## car_mileage..km             2.072e-07  8.420e-08   2.461   0.0139 *
## post_days                   1.131e-03  7.712e-04   1.467   0.1425
## is_2_3_doorsTRUE           -3.578e-02  1.798e-02  -1.990   0.0467 *
## A.Cmanual A/C              -7.468e-02  1.346e-02  -5.548 3.01e-08 ***
## A.Cno A/C                  -2.732e-01  2.404e-02 -11.366  < 2e-16 ***
## type_of_driveback          -7.194e-02  3.250e-02  -2.213   0.0269 *
## type_of_drivefront         -1.718e-01  2.600e-02  -6.608 4.22e-11 ***
## car_typeFamily             -9.257e-02  1.569e-02  -5.900 3.82e-09 ***
## car_typeLimousine          -4.633e-03  1.722e-02  -0.269   0.7879
## car_typePickup              2.010e-01  3.782e-02   5.313 1.12e-07 ***
## car_typeSport               2.004e-01  4.994e-02   4.013 6.07e-05 ***
## car_typeSUV                 2.714e-01  2.654e-02  10.229  < 2e-16 ***
## car_brand_categoryLuxury    3.947e-01  2.063e-02  19.137  < 2e-16 ***
## car_brand_categoryOther    -5.134e-02  3.281e-02  -1.565   0.1177
## car_brand_categoryStandard  2.146e-01  1.346e-02  15.946  < 2e-16 ***
## gearbox_groupmanual        -8.240e-02  1.853e-02  -4.447 8.88e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.4414 on 6170 degrees of freedom
## Multiple R-squared:  0.7449, Adjusted R-squared:  0.7441
## F-statistic: 948.2 on 19 and 6170 DF,  p-value: < 2.2e-16
```

```
reduced_model_2<- lm(
  log_price ~ favorite + year + horsepower + car_mileage..km +
    is_2_3_doors +
    A.C + type_of_drive + car_type +
    car_brand_category + gearbox_group,
  data = car_model
)
summary(reduced_model_2)
```

```
##
## Call:
## lm(formula = log_price ~ favorite + year + horsepower + car_mileage..km +
##     is_2_3_doors + A.C + type_of_drive + car_type + car_brand_category +
##     gearbox_group, data = car_model)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3857 -0.1650  0.0518  0.2397  1.9871
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                -2.417e+02  2.770e+00 -87.280  < 2e-16 ***
## favorite                   -3.016e-02  1.712e-03 -17.619  < 2e-16 ***
## year                        1.243e-01  1.376e-03  90.344  < 2e-16 ***
## horsepower                  3.635e-03  2.186e-04  16.627  < 2e-16 ***
## car_mileage..km             2.095e-07  8.420e-08   2.488   0.0129 *
## is_2_3_doorsTRUE           -3.543e-02  1.798e-02  -1.970   0.0489 *
## A.Cmanual A/C              -7.470e-02  1.346e-02  -5.549 3.00e-08 ***
## A.Cno A/C                  -2.733e-01  2.404e-02 -11.370  < 2e-16 ***
## type_of_driveback          -7.197e-02  3.250e-02  -2.214   0.0268 *
## type_of_drivefront         -1.717e-01  2.600e-02  -6.604 4.32e-11 ***
## car_typeFamily             -9.188e-02  1.568e-02  -5.858 4.92e-09 ***
## car_typeLimousine          -3.955e-03  1.721e-02  -0.230   0.8183
## car_typePickup              2.009e-01  3.783e-02   5.311 1.13e-07 ***
## car_typeSport               1.994e-01  4.994e-02   3.992 6.63e-05 ***
## car_typeSUV                 2.713e-01  2.654e-02  10.224  < 2e-16 ***
## car_brand_categoryLuxury    3.942e-01  2.063e-02  19.113  < 2e-16 ***
## car_brand_categoryOther    -5.212e-02  3.281e-02  -1.588   0.1122
## car_brand_categoryStandard  2.145e-01  1.346e-02  15.936  < 2e-16 ***
## gearbox_groupmanual        -8.251e-02  1.853e-02  -4.452 8.65e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4414 on 6171 degrees of freedom
## Multiple R-squared:  0.7448, Adjusted R-squared:  0.7441
## F-statistic:  1001 on 18 and 6171 DF,  p-value: < 2.2e-16
```

```
par(mfrow = c(2, 2)) # Set up a 2x2 plotting layout
plot(reduced_model) # Automatically generates the 4 diagnostic plots
```



- Residuals are mostly centered around zero with no strong non-linear pattern in the Residuals vs Fitted plot.
- The Q–Q plot shows heavier-than-normal tails and a few extreme outliers.
- The Scale–Location plot indicates mild heteroskedasticity, with slightly greater residual spread for certain fitted values.
- A small number of points exhibit relatively high leverage and large residuals.

These issues are not severe given the large sample size, but they suggest prediction errors may increase for atypical vehicles.

```
r_std <- rstandard(reduced_model)
which(abs(r_std) > 3)
```

```
##   18   43   97  143  163  197  198  223  511  548  716  819 1189 1232 1234 1277
##   18   43   97  143  163  197  198  223  511  548  716  819 1189 1232 1234 1277
## 1368 1406 1431 1528 1685 1855 1874 1879 1891 1899 2119 2120 2249 2264 2299 2306
## 1368 1406 1431 1528 1685 1855 1874 1879 1891 1899 2119 2120 2249 2264 2299 2306
## 2387 2398 2441 2442 2475 2476 2481 2482 2494 2549 2558 2578 2631 2736 2841 2845
## 2387 2398 2441 2442 2475 2476 2481 2482 2494 2549 2558 2578 2631 2736 2841 2845
## 2879 2942 2946 2982 3065 3068 3102 3112 3119 3189 3278 3528 3531 3596 3653 3922
## 2879 2942 2946 2982 3065 3068 3102 3112 3119 3189 3278 3528 3531 3596 3653 3922
## 4038 4346 4357 4361 4615 4620 4623 4627 4629 4794 4817 5004 5113 5132 5135 5352
## 4038 4346 4357 4361 4615 4620 4623 4627 4629 4794 4817 5004 5113 5132 5135 5352
## 5373 5593 5602 5963 6148 6149 6150 6151 6158 6160 6165 6166 6176 6178 6180 6186
## 5373 5593 5602 5963 6148 6149 6150 6151 6158 6160 6165 6166 6176 6178 6180 6186
## 6187
## 6187
```

A few listings had extremely high mileage combined with very low prices (e.g., above 200,000 km but below

500 EUR), which generated large negative standardized residuals and strongly influenced the tails of the residual distribution. We treated these as atypical outliers representing severely damaged or non-standard vehicles and removed them using the rule "mileage > 200,000 km and price < 500 EUR". Re-estimating the model on the cleaned sample yielded very similar coefficients and R^2, but slightly improved residual diagnostics.

```
car_model2 <- car_model %>%
  filter(!(car_mileage..km > 200000 & price < 2000))
```

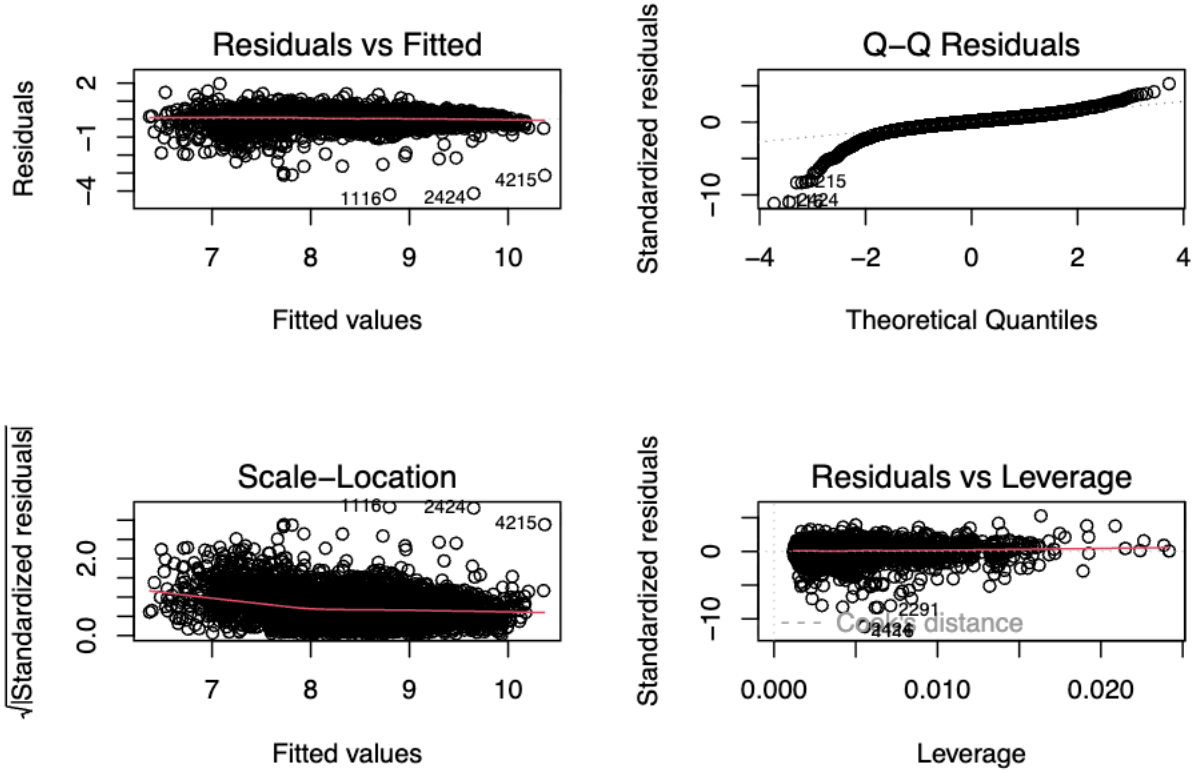
Best Baseline Model

```
reduced_model_3 <- lm(
  log_price ~ favorite + year + horsepower + car_mileage..km +
    post_days + is_2_3_doors +
    A.C + type_of_drive + car_type +
    car_brand_category + gearbox_group,
  data = car_model2
)

summary(reduced_model_3)
```

```
##
## Call:
## lm(formula = log_price ~ favorite + year + horsepower + car_mileage..km +
##     post_days + is_2_3_doors + A.C + type_of_drive + car_type +
##     car_brand_category + gearbox_group, data = car_model2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1929 -0.1523  0.0279  0.2003  1.9688
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 -2.233e+02  2.584e+00 -86.407  < 2e-16 ***
## favorite                    -2.198e-02  1.718e-03 -12.797  < 2e-16 ***
## year                         1.150e-01  1.284e-03  89.587  < 2e-16 ***
## horsepower                   3.565e-03  1.985e-04  17.961  < 2e-16 ***
## car_mileage..km              1.014e-06  7.947e-08  12.763  < 2e-16 ***
## post_days                    2.273e-04  7.221e-04   0.315  0.75297
## is_2_3_doorsTRUE            -4.378e-02  1.739e-02  -2.518  0.01183 *
## A.Cmanual A/C               -5.955e-02  1.264e-02  -4.713 2.51e-06 ***
## A.Cno A/C                   -2.962e-01  2.365e-02 -12.523  < 2e-16 ***
## type_of_driveback           -5.246e-02  2.865e-02  -1.831  0.06711 .
## type_of_drivefront          -1.009e-01  2.268e-02  -4.450 8.78e-06 ***
## car_typeFamily              -8.058e-02  1.477e-02  -5.457 5.07e-08 ***
## car_typeLimousine           -2.568e-03  1.624e-02  -0.158  0.87439
## car_typePickup               2.013e-01  3.476e-02   5.791 7.41e-09 ***
## car_typeSport                1.476e-01  4.486e-02   3.290  0.00101 **
## car_typeSUV                  2.790e-01  2.314e-02  12.057  < 2e-16 ***
## car_brand_categoryLuxury     3.577e-01  1.912e-02  18.706  < 2e-16 ***
## car_brand_categoryOther      6.025e-02  3.275e-02   1.840  0.06587 .
## car_brand_categoryStandard   1.880e-01  1.296e-02  14.504  < 2e-16 ***
## gearbox_groupmanual         -5.994e-02  1.640e-02  -3.655  0.00026 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.3768 on 5143 degrees of freedom
## Multiple R-squared:  0.7506, Adjusted R-squared:  0.7497
## F-statistic: 814.7 on 19 and 5143 DF,  p-value: < 2.2e-16
```

```
par(mfrow = c(2, 2))
plot(reduced_model_3)
```




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

```
library(sandwich)
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
coeftest(reduced_model_3, vcov = vcovHC(reduced_model_3, type = "HC1"))
```

```
##
## t test of coefficients:
##
##                              Estimate  Std. Error  t value  Pr(>|t|)
## (Intercept)                -2.2330e+02  3.2501e+00 -68.7064 < 2.2e-16 ***
## favorite                   -2.1983e-02  2.3233e-03  -9.4619 < 2.2e-16 ***
## year                        1.1504e-01  1.7532e-03  65.6200 < 2.2e-16 ***
## horsepower                  3.5654e-03  2.4917e-04  14.3092 < 2.2e-16 ***
```

```
## car_mileage..km             1.0143e-06  1.2499e-07   8.1155 6.001e-16 ***
## post_days                   2.2728e-04  7.2732e-04   0.3125 0.7546775
## is_2_3_doorsTRUE           -4.3782e-02  2.3491e-02  -1.8638 0.0624125 .
## A.Cmanual A/C              -5.9550e-02  1.2596e-02  -4.7278 2.330e-06 ***
## A.Cno A/C                  -2.9622e-01  3.6704e-02  -8.0705 8.645e-16 ***
## type_of_driveback          -5.2458e-02  6.1548e-02  -0.8523 0.3940799
## type_of_drivefront         -1.0090e-01  7.9522e-02  -1.2688 0.2045562
## car_typeFamily             -8.0583e-02  3.3199e-02  -2.4272 0.0152480 *
## car_typeLimousine          -2.5678e-03  1.6326e+00  -0.0016 0.9987451
## car_typePickup              2.0133e-01  6.8562e-02   2.9364 0.0033351 **
## car_typeSport               1.4758e-01  6.5824e-02   2.2420 0.0250034 *
## car_typeSUV                 2.7898e-01  1.9230e-02  14.5075 < 2.2e-16 ***
## car_brand_categoryLuxury    3.5765e-01  1.8790e-02  19.0344 < 2.2e-16 ***
## car_brand_categoryOther     6.0254e-02  4.1433e-02   1.4543 0.1459367
## car_brand_categoryStandard  1.8802e-01  1.3068e-02  14.3875 < 2.2e-16 ***
## gearbox_groupmanual        -5.9942e-02  1.6193e-02  -3.7017 0.0002164 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

##WLS

```r
ols_fit <- reduced_model_3
w <- 1 / fitted(ols_fit)^2

wls_model <- lm(log_price ~ favorite + year + horsepower + car_mileage..km +
    post_days + is_2_3_doors +
    A.C + type_of_drive + car_type +
    car_brand_category + gearbox_group, data = car_model2, weights = w)
summary(wls_model)
```

```
##
## Call:
## lm(formula = log_price ~ favorite + year + horsepower + car_mileage..km +
##     post_days + is_2_3_doors + A.C + type_of_drive + car_type +
##     car_brand_category + gearbox_group, data = car_model2, weights = w)
##
## Weighted Residuals:
##      Min       1Q   Median       3Q      Max
## -0.47348 -0.01846  0.00323  0.02393  0.28324
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                -2.263e+02  2.721e+00 -83.159  < 2e-16 ***
## favorite                   -2.251e-02  1.724e-03 -13.054  < 2e-16 ***
## year                        1.165e-01  1.353e-03  86.119  < 2e-16 ***
## horsepower                  3.538e-03  2.091e-04  16.920  < 2e-16 ***
## car_mileage..km             1.175e-06  8.109e-08  14.489  < 2e-16 ***
## post_days                   4.226e-04  7.506e-04   0.563 0.573477
## is_2_3_doorsTRUE           -3.456e-02  1.742e-02  -1.984 0.047333 *
## A.Cmanual A/C              -5.489e-02  1.313e-02  -4.181 2.95e-05 ***
## A.Cno A/C                  -3.098e-01  2.329e-02 -13.301  < 2e-16 ***
## type_of_driveback          -6.941e-02  3.107e-02  -2.234 0.025539 *
## type_of_drivefront         -1.213e-01  2.468e-02  -4.918 9.03e-07 ***
## car_typeFamily             -8.177e-02  1.531e-02  -5.341 9.62e-08 ***
## car_typeLimousine          -1.431e-02  1.672e-02  -0.856 0.392035
```

```
## car_typePickup               2.161e-01  3.702e-02   5.837 5.64e-09 ***
## car_typeSport                1.468e-01  4.515e-02   3.251 0.001156 **
## car_typeSUV                  2.821e-01  2.563e-02  11.007  < 2e-16 ***
## car_brand_categoryLuxury     3.584e-01  2.008e-02  17.852  < 2e-16 ***
## car_brand_categoryOther      6.676e-02  3.299e-02   2.024 0.043064 *
## car_brand_categoryStandard   1.925e-01  1.327e-02  14.510  < 2e-16 ***
## gearbox_groupmanual         -6.298e-02  1.773e-02  -3.552 0.000386 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04727 on 5143 degrees of freedom
## Multiple R-squared:  0.7406, Adjusted R-squared:  0.7396
## F-statistic: 772.8 on 19 and 5143 DF,  p-value: < 2.2e-16
```

- **Coefficient direction is identical:**
  Key effects such as `year`, `horsepower`, `SUV`, `luxury brand`, and `automatic gearbox` remain almost unchanged.
- **Significance pattern is also unchanged:**
  `year`, `horsepower`, `A.C`, drivetrain, body type, brand category, and `gearbox_group` stay strongly significant.
  `post_days` remains clearly non-significant, confirming it adds little explanatory power.
- **Model fit is nearly the same:**
  Adjusted R2 indicating almost no difference in overall performance.

Poly

```
car_model2 <- car_model2 %>%
  mutate(
    log_mileage = log1p(car_mileage..km),
    hp_c  = horsepower - mean(horsepower, na.rm = TRUE),
    hp_c2 = hp_c^2
  )

poly_model <- lm(
  log_price ~ favorite + year + hp_c + hp_c2 + log_mileage +
    is_2_3_doors +
    A.C + type_of_drive + car_type +
    car_brand_category + gearbox_group,
  data = car_model2
)

summary(poly_model)
```

```
##
## Call:
## lm(formula = log_price ~ favorite + year + hp_c + hp_c2 + log_mileage +
##     is_2_3_doors + A.C + type_of_drive + car_type + car_brand_category +
##     gearbox_group, data = car_model2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.9222 -0.1525  0.0267  0.1968  1.9540
##
```

```
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  -2.181e+02  2.480e+00 -87.920  < 2e-16 ***
## favorite                     -2.177e-02  1.688e-03 -12.898  < 2e-16 ***
## year                          1.123e-01  1.230e-03  91.268  < 2e-16 ***
## hp_c                          4.266e-03  2.191e-04  19.473  < 2e-16 ***
## hp_c2                        -1.390e-05  2.334e-06  -5.958 2.72e-09 ***
## log_mileage                   8.444e-02  5.228e-03  16.152  < 2e-16 ***
## is_2_3_doorsTRUE             -3.019e-02  1.723e-02  -1.752   0.0799 .
## A.Cmanual A/C                -5.566e-02  1.263e-02  -4.406 1.08e-05 ***
## A.Cno A/C                    -2.835e-01  2.371e-02 -11.957  < 2e-16 ***
## type_of_driveback            -6.856e-02  2.846e-02  -2.409   0.0160 *
## type_of_drivefront           -1.181e-01  2.258e-02  -5.230 1.76e-07 ***
## car_typeFamily               -8.658e-02  1.482e-02  -5.843 5.44e-09 ***
## car_typeLimousine             2.879e-03  1.604e-02   0.179   0.8576
## car_typePickup                2.023e-01  3.426e-02   5.905 3.75e-09 ***
## car_typeSport                 9.349e-02  4.445e-02   2.104   0.0355 *
## car_typeSUV                   2.615e-01  2.316e-02  11.291  < 2e-16 ***
## car_brand_categoryLuxury      3.644e-01  1.881e-02  19.374  < 2e-16 ***
## car_brand_categoryOther       3.517e-02  3.234e-02   1.088   0.2768
## car_brand_categoryStandard    1.883e-01  1.281e-02  14.703  < 2e-16 ***
## gearbox_groupmanual          -7.694e-02  1.644e-02  -4.681 2.93e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3715 on 5143 degrees of freedom
## Multiple R-squared:  0.7575, Adjusted R-squared:  0.7566
## F-statistic: 845.5 on 19 and 5143 DF,  p-value: < 2.2e-16
```

Interaction model

From the car price data, we suspected that factors such as mileage, horsepower, and seating capacity may interact in a way that the combined effect on price is not simple. We noted that these continuous variables can interact with other predictors, such as the age or the number of seats, in order to improve the predictive power of our model.

- car_mileage..km * year : High mileage decreases the price of newer cars more significantly than older ones. When a newer car has high mileage, it indicates that it has been driven extensively in a short period, which may treat as a red flag for buyers.
- horsepower_numeric * year :Technological improvements in rencent years allow newer cars to have higher horsepower without drawbacks, altering its price impact.

```
interaction_model_1 <- lm(
  log_price ~ favorite + year + horsepower + car_mileage..km +
    is_2_3_doors +
    A.C + type_of_drive + car_type +
    car_brand_category + gearbox_group +
    year:car_mileage..km + year:horsepower,
  data = car_model2
)

summary(interaction_model_1)
```

```
##
## Call:
```

```
## lm(formula = log_price ~ favorite + year + horsepower + car_mileage..km +
##     is_2_3_doors + A.C + type_of_drive + car_type + car_brand_category +
##     gearbox_group + year:car_mileage..km + year:horsepower, data = car_model2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3309 -0.1535  0.0267  0.1980  1.9823
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 -2.378e+02  8.336e+00 -28.531  < 2e-16 ***
## favorite                    -2.145e-02  1.709e-03 -12.553  < 2e-16 ***
## year                         1.223e-01  4.152e-03  29.450  < 2e-16 ***
## horsepower                  -1.478e-01  6.603e-02  -2.238 0.025280 *
## car_mileage..km              1.778e-04  2.935e-05   6.056 1.49e-09 ***
## is_2_3_doorsTRUE            -4.112e-02  1.738e-02  -2.365 0.018044 *
## A.Cmanual A/C               -5.833e-02  1.261e-02  -4.626 3.82e-06 ***
## A.Cno A/C                   -2.922e-01  2.373e-02 -12.314  < 2e-16 ***
## type_of_driveback           -5.824e-02  2.862e-02  -2.035 0.041903 *
## type_of_drivefront          -1.046e-01  2.262e-02  -4.624 3.86e-06 ***
## car_typeFamily              -7.698e-02  1.476e-02  -5.216 1.90e-07 ***
## car_typeLimousine           -3.972e-03  1.620e-02  -0.245 0.806330
## car_typePickup               2.004e-01  3.472e-02   5.772 8.31e-09 ***
## car_typeSport                1.505e-01  4.471e-02   3.365 0.000770 ***
## car_typeSUV                  2.705e-01  2.310e-02  11.708  < 2e-16 ***
## car_brand_categoryLuxury     3.531e-01  1.906e-02  18.522  < 2e-16 ***
## car_brand_categoryOther      5.937e-02  3.265e-02   1.818 0.069079 .
## car_brand_categoryStandard   1.854e-01  1.295e-02  14.316  < 2e-16 ***
## gearbox_groupmanual         -5.812e-02  1.651e-02  -3.520 0.000435 ***
## year:car_mileage..km        -8.803e-08  1.462e-08  -6.022 1.85e-09 ***
## year:horsepower              7.537e-05  3.288e-05   2.292 0.021946 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3755 on 5142 degrees of freedom
## Multiple R-squared:  0.7524, Adjusted R-squared:  0.7514
## F-statistic: 781.1 on 20 and 5142 DF,  p-value: < 2.2e-16
```

- Backward stepwise selection systematically removes variables included that contribute minimally towards explanatory power using as criteria the AIC.

```
##
## Call:
## lm(formula = log_price ~ favorite + year + horsepower + car_mileage..km +
##     is_2_3_doors + A.C + type_of_drive + car_type + car_brand_category +
##     gearbox_group + year:car_mileage..km + year:horsepower, data = car_model2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3309 -0.1535  0.0267  0.1980  1.9823
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 -2.378e+02  8.336e+00 -28.531  < 2e-16 ***
## favorite                    -2.145e-02  1.709e-03 -12.553  < 2e-16 ***
```

```
## year                          1.223e-01  4.152e-03  29.450  < 2e-16 ***
## horsepower                   -1.478e-01  6.603e-02  -2.238 0.025280 *
## car_mileage..km               1.778e-04  2.935e-05   6.056 1.49e-09 ***
## is_2_3_doorsTRUE             -4.112e-02  1.738e-02  -2.365 0.018044 *
## A.Cmanual A/C                -5.833e-02  1.261e-02  -4.626 3.82e-06 ***
## A.Cno A/C                    -2.922e-01  2.373e-02 -12.314  < 2e-16 ***
## type_of_driveback            -5.824e-02  2.862e-02  -2.035 0.041903 *
## type_of_drivefront           -1.046e-01  2.262e-02  -4.624 3.86e-06 ***
## car_typeFamily               -7.698e-02  1.476e-02  -5.216 1.90e-07 ***
## car_typeLimousine            -3.972e-03  1.620e-02  -0.245 0.806330
## car_typePickup                2.004e-01  3.472e-02   5.772 8.31e-09 ***
## car_typeSport                 1.505e-01  4.471e-02   3.365 0.000770 ***
## car_typeSUV                   2.705e-01  2.310e-02  11.708  < 2e-16 ***
## car_brand_categoryLuxury      3.531e-01  1.906e-02  18.522  < 2e-16 ***
## car_brand_categoryOther       5.937e-02  3.265e-02   1.818 0.069079 .
## car_brand_categoryStandard    1.854e-01  1.295e-02  14.316  < 2e-16 ***
## gearbox_groupmanual          -5.812e-02  1.651e-02  -3.520 0.000435 ***
## year:car_mileage..km         -8.803e-08  1.462e-08  -6.022 1.85e-09 ***
## year:horsepower               7.537e-05  3.288e-05   2.292 0.021946 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3755 on 5142 degrees of freedom
## Multiple R-squared:  0.7524, Adjusted R-squared:  0.7514
## F-statistic: 781.1 on 20 and 5142 DF,  p-value: < 2.2e-16
```

Model selection

```
# Store models in a named list
models <- list(
  baseline    = reduced_model_3,
  poly        = poly_model,
  interaction = interaction_model
)

# Loop through and print VIF for each model
for (m in names(models)) {
  cat("\n---- VIF for", m, "model ----\n")
  print(vif(models[[m]]))
}
```

```
##
## ---- VIF for baseline model ----
##                          GVIF Df GVIF^(1/(2*Df))
## favorite            1.2957666  1       1.1383174
## year                1.3825561  1       1.1758215
## horsepower          1.4540754  1       1.2058505
## car_mileage..km     1.7161159  1       1.3100061
## post_days           1.5237402  1       1.2343987
## is_2_3_doors        1.4902724  1       1.2207672
## A.C                 1.0129989  2       1.0032340
## type_of_drive      -0.5590018  2             NaN
## car_type           -0.4838816  5             NaN
```

```
## car_brand_category  0.3202891  3       0.8271616
## gearbox_group       1.1600970  1       1.0770780
##
## ---- VIF for poly model ----
##                          GVIF Df GVIF^(1/(2*Df))
## favorite            1.5886424  1       1.2604136
## year                1.7053350  1       1.3058848
## hp_c                1.9254884  1       1.3876197
## hp_c2               2.0042251  1       1.4157066
## log_mileage         1.2699336  1       1.1269133
## is_2_3_doors        1.2418501  1       1.1143833
## A.C                 0.9556486  2       0.9887228
## type_of_drive      -0.5040044  2             NaN
## car_type           -0.4909744  5             NaN
## car_brand_category  0.4089958  3       0.8615619
## gearbox_group       1.2099133  1       1.0999606
##
## ---- VIF for interaction model ----
```

```
## there are higher-order terms (interactions) in this model
## consider setting type = 'predictor'; see ?vif
```

```
##                              GVIF Df GVIF^(1/(2*Df))
## favorite              8.038315e-02  1       0.2835192
## year                  3.578805e-01  1       0.5982312
## horsepower           -1.014470e+03  1             NaN
## car_mileage..km      -1.397616e-02  1             NaN
## is_2_3_doors          9.207041e-02  1       0.3034311
## A.C                   2.054079e+03  2       6.7321581
## type_of_drive         1.733582e+03  2       6.4526204
## car_type             -3.535056e-06  5             NaN
## car_brand_category   -1.395058e+00  3             NaN
## gearbox_group         1.031680e+00  1       1.0157163
## year:car_mileage..km -3.790393e-01  1             NaN
## year:horsepower      -2.078180e+00  1             NaN
```

```
lapply(models, AIC)
```

```
## $baseline
## [1] 4594.631
##
## $poly
## [1] 4450.111
##
## $interaction
## [1] 4560.199
```

```
lapply(models, BIC)
```

```
## $baseline
## [1] 4732.165
##
## $poly
## [1] 4587.645
##
## $interaction
```

```
## [1] 4704.283
```

```
coef_table <- broom::tidy(poly_model)

coef_table |>
  dplyr::mutate(
    term = gsub("is_2_3_doorsTRUE", "2/3 doors (vs 4/5)", term),
    term = gsub("gearbox_groupmanual", "Manual gearbox (vs automatic)", term)
  ) |>
  gt::gt() |>
  gt::fmt_number(
    columns = c(estimate, std.error, statistic, p.value),
    decimals = 3
  )
```

term	estimate	std.error	statistic	p.value
(Intercept)	−218.054	2.480	−87.920	0.000
favorite	−0.022	0.002	−12.898	0.000
year	0.112	0.001	91.268	0.000
hp_c	0.004	0.000	19.473	0.000
hp_c2	0.000	0.000	−5.958	0.000
log_mileage	0.084	0.005	16.152	0.000
2/3 doors (vs 4/5)	−0.030	0.017	−1.752	0.080
A.Cmanual A/C	−0.056	0.013	−4.406	0.000
A.Cno A/C	−0.284	0.024	−11.957	0.000
type_of_driveback	−0.069	0.028	−2.409	0.016
type_of_drivefront	−0.118	0.023	−5.230	0.000
car_typeFamily	−0.087	0.015	−5.843	0.000
car_typeLimousine	0.003	0.016	0.179	0.858
car_typePickup	0.202	0.034	5.905	0.000
car_typeSport	0.093	0.044	2.104	0.035
car_typeSUV	0.262	0.023	11.291	0.000
car_brand_categoryLuxury	0.364	0.019	19.374	0.000
car_brand_categoryOther	0.035	0.032	1.088	0.277
car_brand_categoryStandard	0.188	0.013	14.703	0.000
Manual gearbox (vs automatic)	−0.077	0.016	−4.681	0.000

SO WE CHOSE POLYNOMIAL MODEL AS FINAL MODEL!!

```
final_model <- poly_model
final_formula <- log_price ~ favorite + year + hp_c + hp_c2 + log_mileage +
    is_2_3_doors + A.C + type_of_drive + car_type + car_brand_category +
    gearbox_group
```

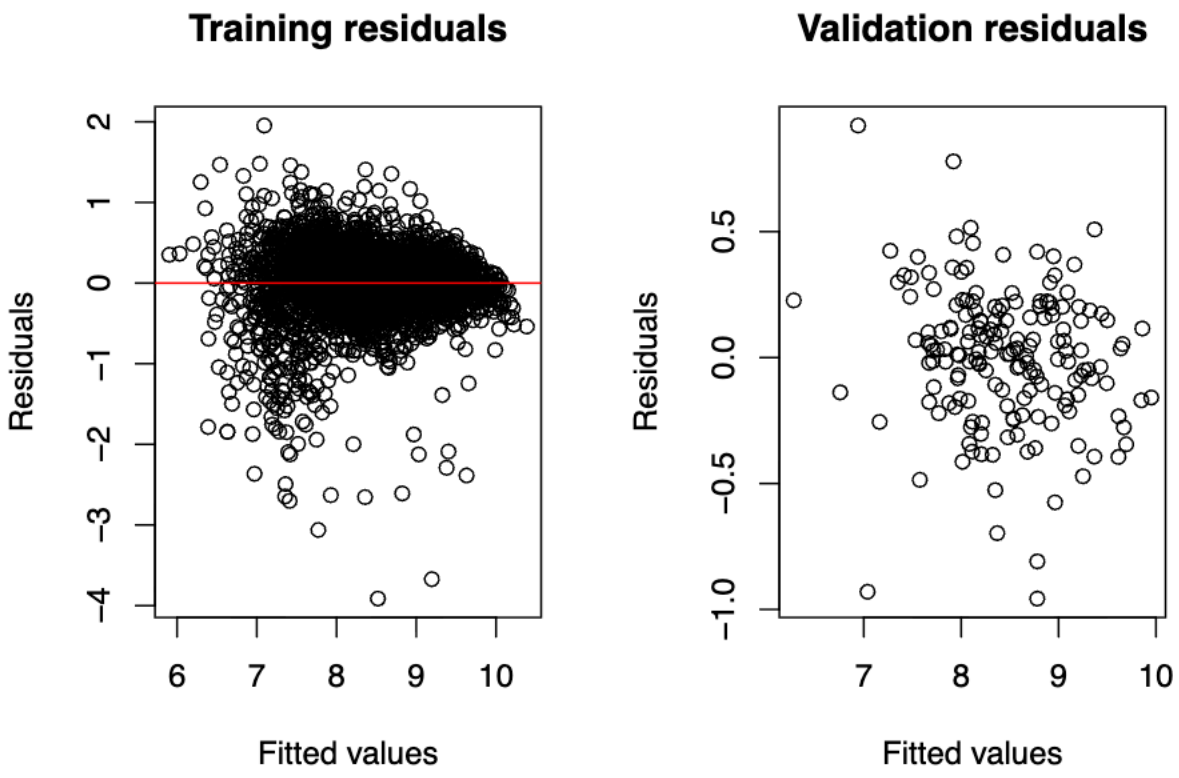
```
set.seed(11)
k <- 30

data_cv <- car_model2[complete.cases(model.matrix(final_formula, data = car_model2)), ]
data_cv <- data_cv[sample(1:nrow(data_cv)), ]

folds <- cut(seq_len(nrow(data_cv)), breaks = k, labels = FALSE)

mse_train <- numeric(k)
```

```r
mse_valid <- numeric(k)

for (i in 1:k) {
valid_idx  <- which(folds == i)
valid_data <- data_cv[ valid_idx, ]
train_data <- data_cv[-valid_idx, ]

fit_i <- lm(final_formula, data = train_data)

y_train <- model.response(model.frame(final_formula, data = train_data))
y_valid <- model.response(model.frame(final_formula, data = valid_data))

pred_train <- predict(fit_i, newdata = train_data)
pred_valid <- predict(fit_i, newdata = valid_data)

mse_train[i] <- mean((y_train - pred_train)^2)
mse_valid[i] <- mean((y_valid - pred_valid)^2)
}

cv_results <- data.frame(
Metric = c("Average MSE (training)", "Average MSE (validation)"),
Value  = c(mean(mse_train), mean(mse_valid))
)

knitr::kable(cv_results, digits = 4,
caption = "30-fold cross-validation results for the final model")
```

Table 12: 30-fold cross-validation results for the final model

Metric	Value
Average MSE (training)	0.1375
Average MSE (validation)	0.1395

```r
par(mfrow = c(1, 2))
plot(pred_train, y_train - pred_train,
main = "Training residuals",
xlab = "Fitted values", ylab = "Residuals")
abline(h = 0, col = "red")

plot(pred_valid, y_valid - pred_valid,
main = "Validation residuals",
xlab = "Fitted values", ylab = "Residuals")
```



PI and CI

```
final_fit <- lm(final_formula, data = car_model2)

set.seed(123)
new_data <- car_model2[sample(1:nrow(car_model2), 5), ]

#CI & PI at log scale

ci_log <- predict(final_fit, newdata = new_data, interval = "confidence")
pi_log <- predict(final_fit, newdata = new_data, interval = "prediction")
# back to price scale

results_pi_ci <- data.frame(
actual_price = new_data$price,
fitted_price = exp(ci_log[,"fit"]),
CI_lower     = exp(ci_log[,"lwr"]),
CI_upper     = exp(ci_log[,"upr"]),
PI_lower     = exp(pi_log[,"lwr"]),
PI_upper     = exp(pi_log[,"upr"])
)

knitr::kable(results_pi_ci, digits = 1,
caption = "Confidence and prediction intervals for selected cars (price scale)")
```

Table 13: Confidence and prediction intervals for selected cars (price scale)

	actual_price	fitted_price	CI_lower	CI_upper	PI_lower	PI_upper
2463	3999	3816.0	3720.8	3913.7	1841.2	7909.2
2511	8499	9810.2	9399.9	10238.5	4729.4	20349.3
2227	16999	10409.5	9897.7	10947.9	5015.9	21603.2
526	10500	10073.5	9561.2	10613.3	4853.4	20908.4
4291	4550	2264.2	2201.2	2328.9	1092.3	4693.2

trimmed final model

```
rstd <- rstandard(poly_model)
car_model3 <- car_model2[abs(rstd) <= 3.5, ]

poly_model_trimmed <- lm(
  log_price ~ favorite + year + hp_c + hp_c2 + log_mileage +
    is_2_3_doors + A.C + type_of_drive + car_type +
    car_brand_category + gearbox_group,
  data = car_model3
)

summary(poly_model_trimmed)
```

```
##
## Call:
## lm(formula = log_price ~ favorite + year + hp_c + hp_c2 + log_mileage +
##     is_2_3_doors + A.C + type_of_drive + car_type + car_brand_category +
##     gearbox_group, data = car_model3)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.40600 -0.15547  0.01713  0.18005  1.18854
##
## Coefficients:
##                               Estimate Std. Error  t value Pr(>|t|)
## (Intercept)                 -2.135e+02  2.033e+00 -104.990  < 2e-16 ***
## favorite                    -2.021e-02  1.391e-03  -14.531  < 2e-16 ***
## year                         1.102e-01  1.008e-03  109.347  < 2e-16 ***
## hp_c                         4.328e-03  1.797e-04   24.089  < 2e-16 ***
## hp_c2                       -1.306e-05  1.916e-06   -6.816 1.04e-11 ***
## log_mileage                  4.272e-02  4.551e-03    9.386  < 2e-16 ***
## is_2_3_doorsTRUE            -3.603e-02  1.412e-02   -2.551   0.0108 *
## A.Cmanual A/C               -5.826e-02  1.030e-02   -5.655 1.64e-08 ***
## A.Cno A/C                   -2.284e-01  1.965e-02  -11.626  < 2e-16 ***
## type_of_driveback           -5.654e-02  2.325e-02   -2.432   0.0150 *
## type_of_drivefront          -1.206e-01  1.841e-02   -6.548 6.39e-11 ***
## car_typeFamily              -7.749e-02  1.209e-02   -6.409 1.59e-10 ***
## car_typeLimousine            1.530e-02  1.314e-02    1.164   0.2443
## car_typePickup               1.929e-01  2.792e-02    6.911 5.41e-12 ***
## car_typeSport                4.411e-02  3.714e-02    1.188   0.2349
## car_typeSUV                  2.628e-01  1.885e-02   13.943  < 2e-16 ***
## car_brand_categoryLuxury     3.552e-01  1.534e-02   23.162  < 2e-16 ***
## car_brand_categoryOther      3.819e-02  2.657e-02    1.437   0.1507
```

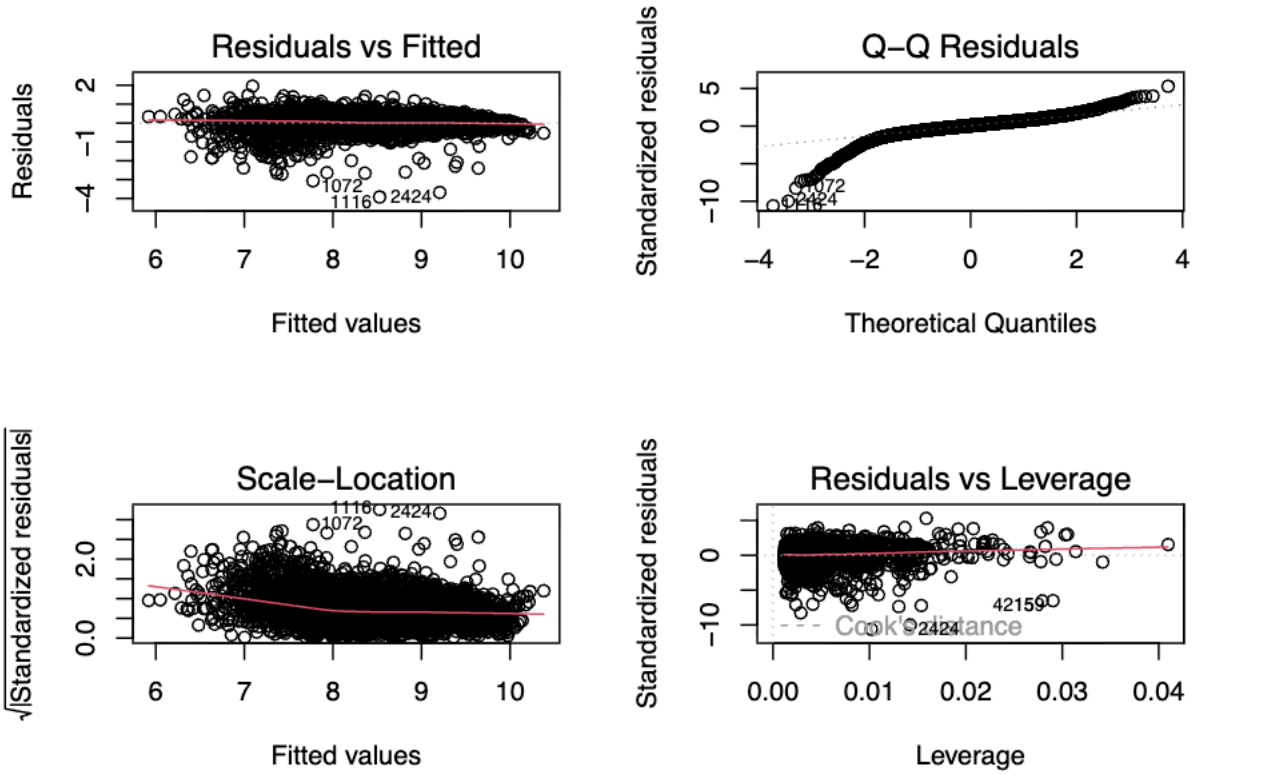
```
## car_brand_categoryStandard  1.810e-01  1.044e-02   17.326  < 2e-16 ***
## gearbox_groupmanual        -7.167e-02  1.338e-02   -5.358 8.78e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3013 on 5078 degrees of freedom
## Multiple R-squared:  0.816,  Adjusted R-squared:  0.8153
## F-statistic:  1185 on 19 and 5078 DF,  p-value: < 2.2e-16
```

Model Diagnostics

Residual plots

We first evaluated the OLS baseline model and our final polynomial model using the four standard diagnostic plots (Residuals vs Fitted, Normal Q–Q, Scale–Location and Residuals vs Leverage). For the polynomial specification, the Residuals vs Fitted plot shows a roughly symmetric cloud of points around zero, with no strong curvature, suggesting that the log transformation of price, the log transformation of mileage and the quadratic term in horsepower capture most of the non-linear patterns. The Q–Q plot still exhibits noticeable deviations in the extreme tails: very cheap and very expensive cars are harder to predict, and residuals for those listings are larger in magnitude. The Scale–Location plot indicates mild heteroskedasticity, with slightly smaller residual variance at higher fitted values.

```
par(mfrow = c(2, 2))
plot(poly_model)
```




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

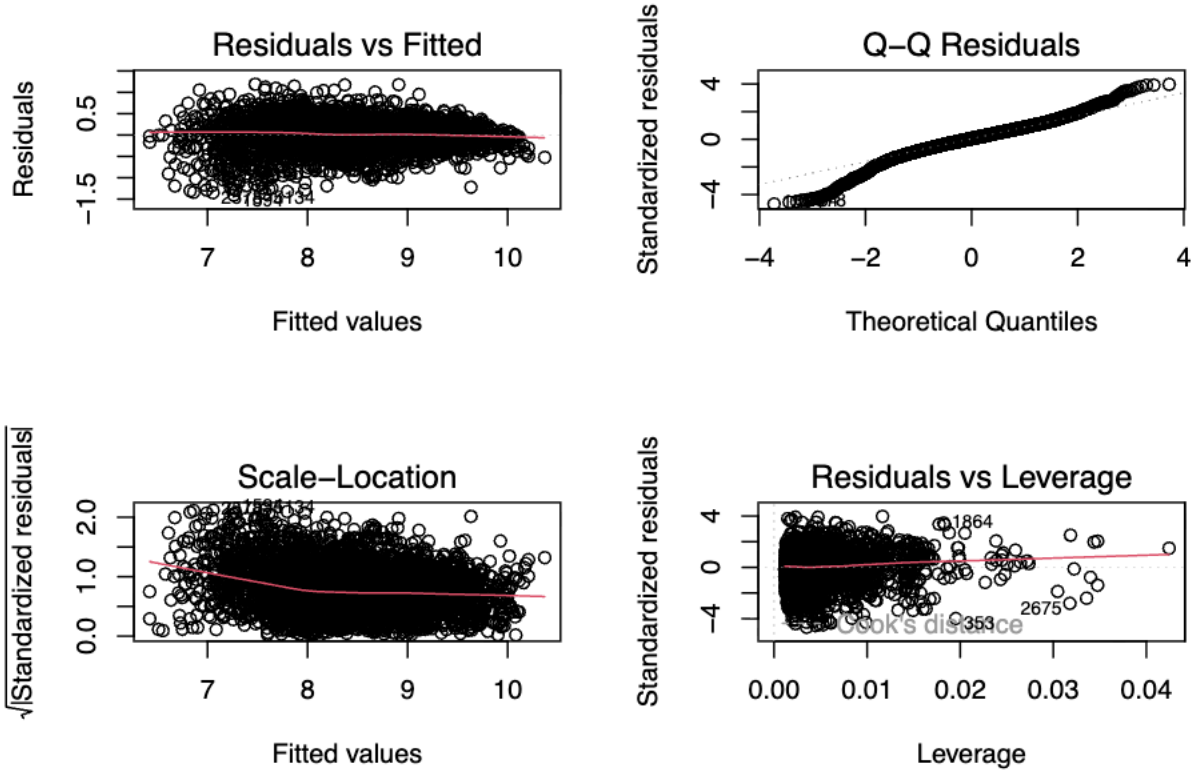
Outlier

We examined standardized residuals and influence measures to identify unusual listings. A small number of observations have |standardized residuals| above about 3.5 and appear at the ends of the Q–Q plot; these correspond to extreme combinations such as very high mileage with very low price. We also inspected leverage and Cook's distance in the Residuals vs Leverage plot and found only a handful of points with both high leverage and large residuals. As a robustness check, we trimmed observations with |standardized residual| > 3.5 and refitted the polynomial model (poly_model_trimmed). The diagnostic plots become slightly tighter, but the estimated coefficients and their significance are very similar to the untrimmed model, indicating that our conclusions are not driven by a few extreme listings.

```
rstd <- rstandard(poly_model)
table(abs(rstd) > 3.5)   # number of extreme residuals
```

```
##
## FALSE  TRUE
##  5098    65
```

```
par(mfrow = c(2, 2))
plot(poly_model_trimmed) # trimmed robustness model
```



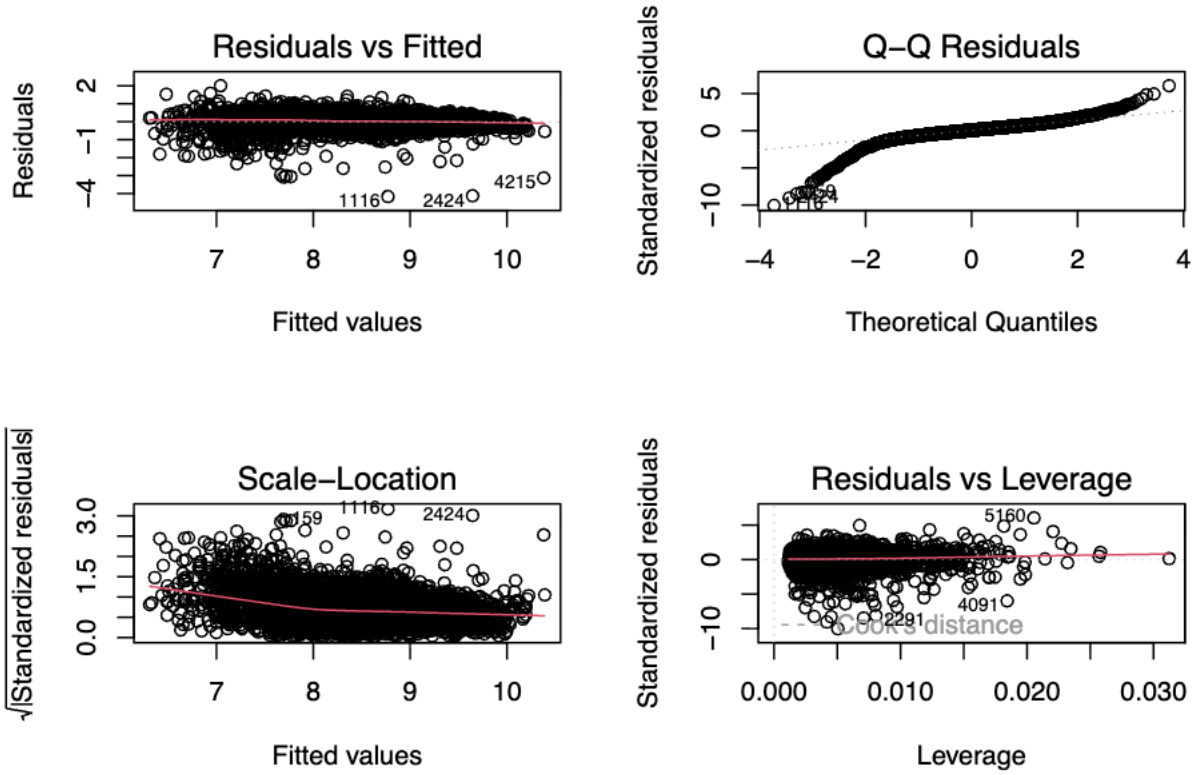

```
par(mfrow = c(1, 1))
```

Linearity

Scatterplots of price against the main continuous predictors showed strong non-linear patterns: mileage is highly right-skewed and its marginal effect on price becomes flatter at high values, while horsepower has a curved relationship with price. To address this, we modelled log(price) as the response, used log_mileage instead of raw mileage, and added a quadratic term in centred horsepower (hp_c2). After these transformations, the Residuals vs Fitted plot no longer shows pronounced curvature, suggesting that the linearity assumption is reasonable on the transformed scale. However, the Scale–Location plot still indicates mild

heteroskedasticity. Therefore we additionally estimated a weighted least squares (WLS) version of the polynomial model. The WLS fit reduces the spread of residuals for high fitted values, but the main coefficient estimates and their t-statistics remain very close to the OLS results, which supports the robustness of our findings.

```
par(mfrow = c(2, 2))
plot(wls_model)
```




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

Overall

Overall, the final polynomial model with log(price) as the response, log mileage, a quadratic horsepower term and grouped categorical predictors achieves an adjusted R2 of about 0.76 and captures the main economic relationships in the data: newer, more powerful, low-mileage cars from premium brands command higher prices. The diagnostic checks show that the linearity assumption is broadly satisfied after transformation, and that heteroskedasticity and non-normality are present mainly in the extreme tails but not severe. Outlier trimming and WLS robustness checks lead to very similar coefficient estimates. We therefore conclude that the model is adequate for explaining and predicting used-car prices in this dataset, while acknowledging that prediction error is larger for the most unusual listings.