# MDA9159 Final Report

# Predicting Used Car Prices with Regression Models

Zongyu LI

Jinmin Li

Sam Guan

# 1 Executive Summary

Our project is aimed at discovering what variables contribute to the prices of used cars in Serbian automotive market. Analysis was based on a cleaned dataset which included major vehicle features including price, mileage, year, horsepower, body, fuel type, transmission, and brand type and much preparation had been done to eliminate inconsistency, as well as category structure variable.

A number of regression models were run and the specification that was chosen was the polynomial one, which included a squared horsepower term and a log of the mileage. It also produced superior AIC/BIC scores as compared to the baseline and interaction models, did not suffer the effects of multicollinearity, and the predictive performance in cross validated manner in addition to demonstrating more nonlinear trends in the data.

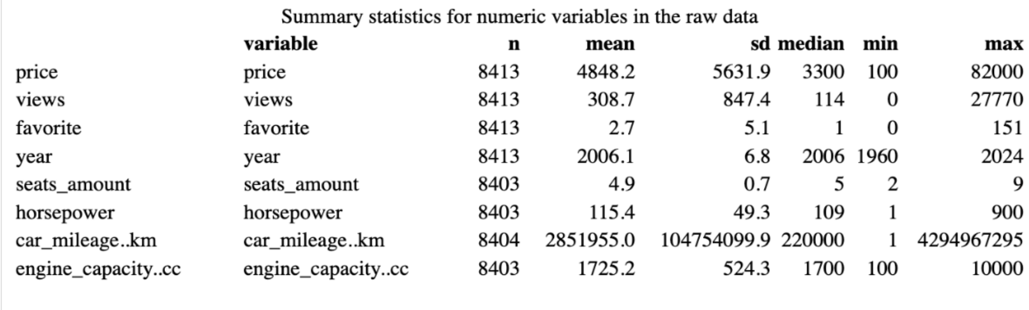
The discoveries are that newer vehicles, increased horsepower, luxurious brands and SUVs are linked to upsurge in price and thus, the segment Budget and Family incur less price. The terms of the polynomials signify a decreasing profit of engine power and the depreciation of the effect of mileage is not linear. On the whole, the last model is able to provide a transparent and understandable explanation of the majority of the variation in the used car prices in Serbia.

# 2 Data Description & Exploratory Analysis

## 2.1 Variable Description

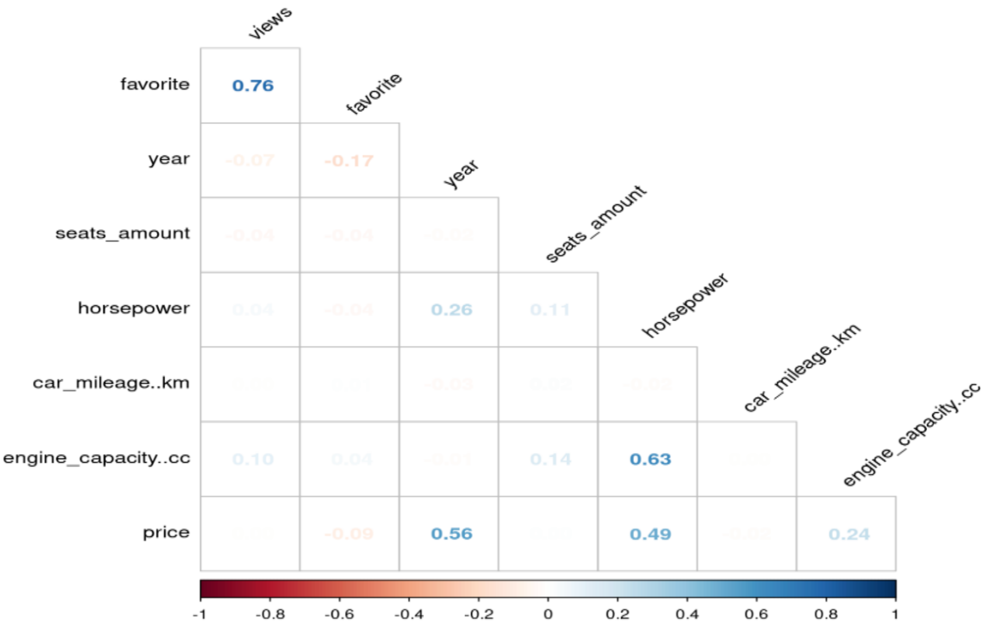
The dataset includes a range of variables that describe both the characteristics of each car and the details of its online listing. **Price** represents the listed selling price in euros and serves as the primary outcome variable. Listing engagement is captured through **views** and **favorite**, which measure how often each advertisement was seen and marked by users. **Year** identifies the vehicle's manufacturing year, while **car_mileage**, km records total distance driven, both of which are key indicators of vehicle age and wear. Engine-related attributes include **engine_capacity**, cc, which measures engine displacement, and **horsepower**, which reports engine power in text format. Structural and physical features are described by **seats amount**, **body type**, and **color**, while **A/C** indicates the presence of air conditioning and emission class reflects environmental compliance. Mechanical characteristics such as **fuel_type** and **transmission** further distinguish vehicle types. Additionally, **brand** and model categorize each car by manufacturer and specific variant, and **car_name** provides the original listing title. Finally, **post info** contains descriptive text from the advertisement. Together, these variables provide a comprehensive representation of vehicle specifications, condition indicators, and listing-level information relevant to explaining price variation.

## 2.2 Summary Statistics

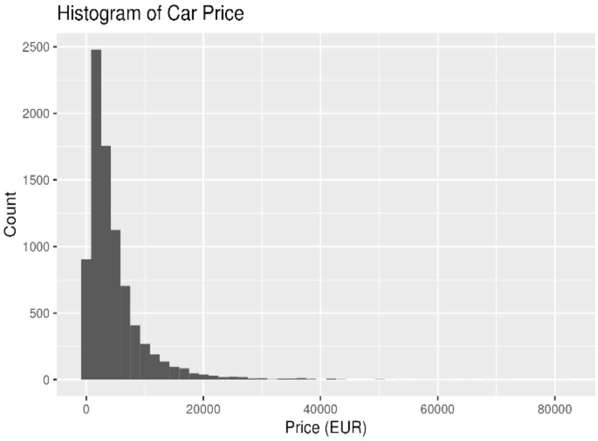


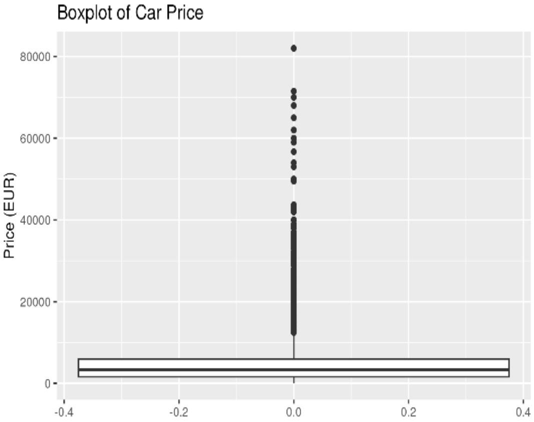
Summary statistics for numeric variables in the raw data

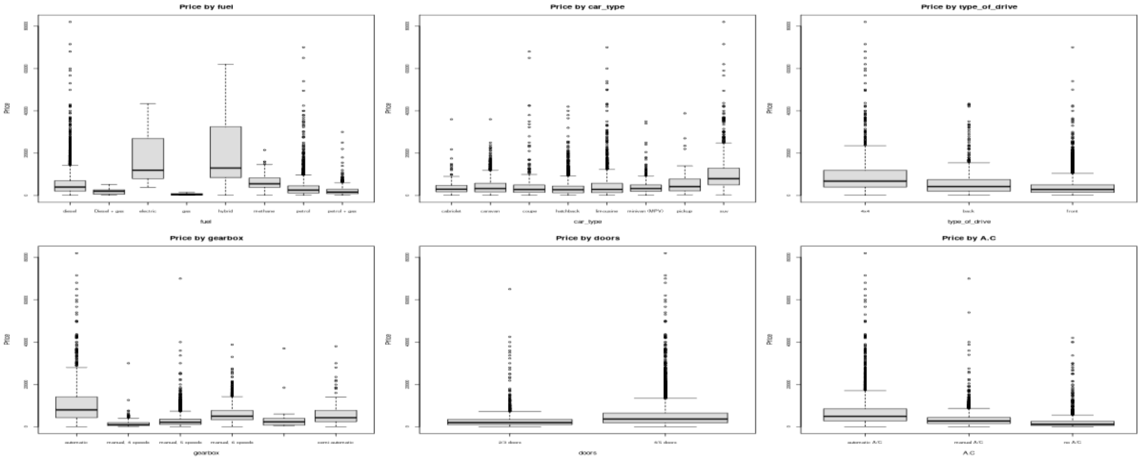
| | variable | n | mean | sd | median | min | max |
|---|---|---|---|---|---|---|---|
| price | price | 8413 | 4848.2 | 5631.9 | 3300 | 100 | 82000 |
| views | views | 8413 | 308.7 | 847.4 | 114 | 0 | 27770 |
| favorite | favorite | 8413 | 2.7 | 5.1 | 1 | 0 | 151 |
| year | year | 8413 | 2006.1 | 6.8 | 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 |

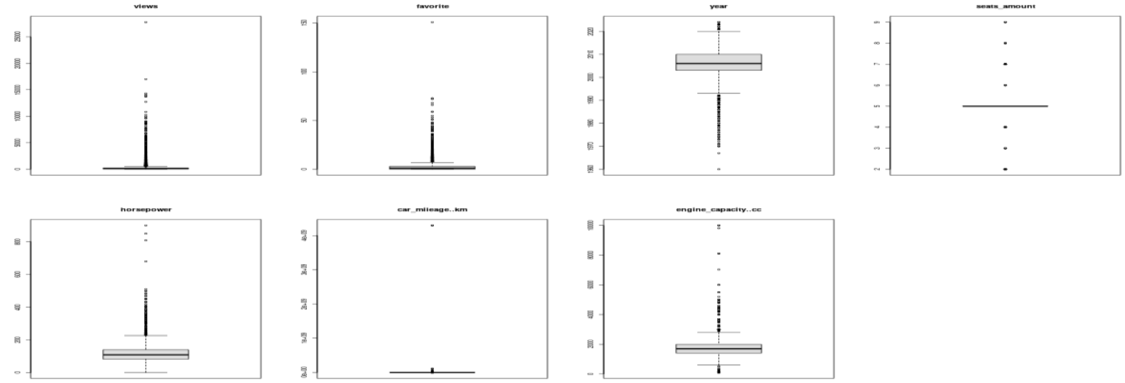


## 2.3 Exploratory Plots









## 2.4 Discussion of Observed Patterns and Potential Issues

There are evident and intuitive connections between price and car attributes as seen in the exploratory analysis. The newer vehicles, fewer miles covered, and higher engine capacities are all related to high prices. The brand also has a huge influence on the price expectations with German manufacturers taking control of the high-end segment. Such findings indicate that there are universal tendencies in the used-car market in Europe. There are however serious problems with the dataset as well. Values such as mileage and engine capacity have the most unrealistic values, some involved reaching up to 1 billion km and 10,000 ccs respectively. It is necessary to clean up these then model them. The horsepower variable is in form of text and needs to be converted into numbers. Some of the variables are the absence of the missing values such as type of emission, amount of seats, color, and the availability of A/C. The skew of the prices indicate that possible log transform to stabilize the variance and boost the model fit are required.

# 3 Model Building & Justification

## 3.1 Data Cleaning

The raw dataset required extensive cleaning to ensure accuracy and suitability for modelling. First, a new variable **car_brand** was extracted from the original car_name field, after which the full name column was removed. Outliers in the response variable **price** were trimmed using a percentile-based IQR filter, and additional unrealistic values were removed by restricting mileage, favorite counts, horsepower, and year to reasonable ranges. Several categorical variables were also cleaned and recoded. Vehicle body types were consolidated into broader **car_type** categories (e.g., Limousine, Family, Economy, SUV), and empty emission class entries were removed. Emission classes were further grouped into **Large Displacement** and **Small Displacement** categories. Door information was simplified into a binary indicator for **2/3-door vehicles**, and the original variable was dropped. Fuel types were reclassified into **Oil**, **Electric**, **Mixture**, or **Unknown**, and brand names were mapped into **Luxury**, **Standard**, **Budget**, or **Other** categories, after which the original brand variable was discarded. Gearbox descriptions were standardized into **automatic**, **manual**, or **other**. A second filtering step removed extreme values and unknown brand groups to further refine the dataset. Finally, a new numerical variable **post_days** was created by converting the textual posting age (e.g., “today,” “yesterday,” “3 days,” “2 weeks,” “a month”) into consistent numeric day values. After the transformation, the unneeded post_info field was removed.

## 3.2 Base Model Construction

### Initial Multiple Linear Regression and Categorical Encoding

To explain and predict listing prices, we model the natural logarithm of price as the response variable in order to stabilise variance and interpret effects approximately as percentage changes. Let $i$ index individual listings. The initial multiple linear regression (MLR) is specified as

$$\begin{aligned}\log\left(\text{ price }_{i}\right) =&\beta_0+\beta_1 \text{ views }_{i}+\beta_2 \text{ favorite }_{i}+\beta_3 \text{ year }_{i}+\beta_4 \text{ seats }_{i}+\beta_5 \text{ hp }_{i} \\ &+\beta_6 \text{ mileage }_{i}+\beta_7 \text{ engine }_{i}+\beta_8 \text{ postdays }_{i}+\gamma^{\prime} \mathbf{Z}_i+\beta_{25} \text{ doors }_{2} 3_i+\varepsilon_i .\end{aligned}$$

Here, $\mathbf{Z}_i$ collects the dummy variables corresponding to the categorical predictors A.C, type_of_drive, car_type, emission_type, fuel_type, car_brand_category, and gearbox_group. In the dataset, the continuous components of the linear predictor correspond to the variables views, favorite, year, seats_amount, horsepower, car_mileage..km, engine_capacity..cc, post_days, and the binary indicator is_2_3_doors.

In R, categorical predictors are converted to factors via mutate(A.C = factor(A.C), ..., gearbox_group = factor(gearbox_group)), which generates standard treatment contrasts with one reference category per factor. The variable is_2_3_doors is treated as a binary indicator (FALSE/TRUE).

**Baseline Main-Effects Model**

Although the full model is useful as a starting point, it is relatively complex and includes predictors that are weak from both a statistical and an economic perspective. To obtain a more parsimonious baseline specification, we construct a reduced main-effects model that retains the core determinants of vehicle prices while excluding clearly uninformative terms.

In particular, we simplify the full model by removing predictors whose estimated effects are small and lack clear economic interpretation in the presence of more directly relevant variables (for example, keeping year and horsepower rather than additional technical details that provide little incremental information). In addition, exploratory analysis of standardised residuals identifies a small number of extreme outliers with very high mileage and very low prices (mileage $> 200{,}000 \text{ km}$ and price $< 2{,}000 \text{EUR}$ ). These observations are likely to correspond to atypical or heavily damaged vehicles, and are therefore excluded to prevent them from unduly influencing the fitted relationships.

On the cleaned dataset, the resulting baseline model is

$$\begin{aligned}\log\left(\text{ price }_i\right) =& \beta_0+\beta_1 \text{ favorite }_i+\beta_2 \text{ year }_i+\beta_3 \text{ hp }_i+\beta_4 \text{ mileage }_i+\beta_5 \text{ postdays }_i+\beta_6 \text{ doors } 23_i \\ &+\boldsymbol{\gamma}_{AC}^{\prime} \mathbf{AC}_i+\boldsymbol{\gamma}_{TD}^{\prime} \mathbf{TD}_i+\boldsymbol{\gamma}_{CT}^{\prime} \mathbf{CT}_i+\boldsymbol{\gamma}_{BC}^{\prime} \mathbf{BC}_i+\boldsymbol{\gamma}_{GB}^{\prime} \mathbf{GB}_i+\varepsilon_i\end{aligned}$$

where $\mathbf{AC}_i, \mathbf{TD}_i, \mathbf{CT}_i, \mathbf{BC}_i$, and $\mathbf{GB}_i$ are the dummy-variable vectors associated with the levels of A.C, type_of_drive, car_type, car_brand_category, and gearbox_group respectively. This baseline specification keeps demandrelated information (favorite), age and usage (year, mileage, postdays), performance (hp), body configuration (doors2_3, car_type), comfort and drivetrain (A.C, type_of_drive), as well as brand positioning and transmission (car_brand_category, gearbox_group).

**Polynomial Model**

Exploratory plots and residual patterns from the baseline model indicate that the relationships between log price and key continuous predictors are not purely linear. In particular, there appear to be diminishing returns to additional horsepower and a nonlinear (concave) dependence of price on mileage: adding horsepower is more valuable at low power levels than at already high horsepower, and the marginal price penalty associated with additional kilometres tends to diminish at very high mileage.

To capture these nonlinearities while maintaining interpretability, we extend the baseline specification in two ways:

- we introduce a centred linear and quadratic term for horsepower,

$$hp_{c,i} = \mathrm{hp}_i - \overline{\mathrm{hp}}, hp_{c,i}^2$$

so that curvature in the price-horsepower relationship can be modelled with reduced collinearity between the linear and quadratic terms; and

- we replace raw mileage with a log-transformed mileage term,

$$\log\left(1 + \text{ mileage }_i\right)$$

which reflects diminishing marginal effects of additional kilometres and mitigates skewness in the distribution of mileage.

The polynomial model is therefore specified as

$$\begin{aligned}\log\left(\text{ price }_i\right) =& \beta_0 + \beta_1 \text{ favorite }_i + \beta_2 \text{ year }_i + \beta_3 hp_{c,i} + \beta_4 hp_{c,i}^2 + \beta_5 \log\left(1 + \text{ mileage }_i\right) \\ &+ \beta_6 \text{ doors } 23_i + \boldsymbol{\gamma}_{AC}' \mathbf{AC}_i + \boldsymbol{\gamma}_{TD}' \mathbf{TD}_i + \boldsymbol{\gamma}_{CT}' \mathbf{CT}_i + \boldsymbol{\gamma}_{BC}' \mathbf{BC}_i + \boldsymbol{\gamma}_{GB}' \mathbf{GB}_i + \varepsilon_i\end{aligned}$$

This specification allows the model to accommodate economically meaningful curvature in the effects of horsepower and mileage on log price, without altering the interpretation of the remaining coefficients. The formal evaluation of its performance relative to the baseline model is presented in the model selection section.

### Interaction Model

In addition to nonlinear effects in individual predictors, it is plausible that the impact of mileage and horsepower on price depends on vehicle age. For example, high mileage is likely to be penalised more strongly for relatively new cars than for much older vehicles, since intensive use over a short period raises concerns about wear and tear. Similarly, technological progress may have changed how buyers value horsepower for recent model years, so that an increase in horsepower does not have the same price effect across all cohorts.

To account for such age-dependent effects, we consider an interaction model that augments the main-effects specification with interactions between year and the two key continuous characteristics, mileage and horsepower:

$$\begin{aligned}\log\left(\text{price}_i\right) =&\beta_0 + \beta_1 \text{ favorite}_i + \beta_2 \text{ year}_i + \beta_3 \mathrm{hp}_i + \beta_4 \text{ mileage}_i + \beta_5 \text{ door}\\ &+\boldsymbol{\gamma}'_{AC}\mathbf{AC}_i + \boldsymbol{\gamma}'_{TD}\mathbf{TD}_i + \boldsymbol{\gamma}'_{CT}\mathbf{CT}_i + \boldsymbol{\gamma}'_{BC}\mathbf{BC}_i + \boldsymbol{\gamma}'_{GB}\mathbf{GB}_i\\ &+\delta_1\left(\text{year}_i \times \text{ mileage}_i\right) + \delta_2\left(\text{year}_i \times \text{ hp}_i\right) + \varepsilon_i\end{aligned}$$

The interaction terms $\delta_1$ and $\delta_2$ allow the slope of log price with respect to mileage and horsepower to vary with model year, capturing cohort-specific valuation of usage and performance. The extent to which these interactions improve the model is examined in the model selection section.

## 3.3 Model Selection

**Model Comparison Using Information Criteria**

Three candidate specifications were evaluated: the baseline linear model, a polynomial model incorporating $hp_c$ , $hp_c^2$ and $\log(\text{mileage})$ , and an interaction model with year–mileage and year–horsepower interactions. Model performance was compared using AIC, BIC and, where applicable, nested F-tests. The polynomial model achieves the lowest AIC (4450.1) and BIC (4587.6), representing a substantial improvement over the baseline model, while the interaction model reduces AIC only slightly relative to the baseline and remains worse than the polynomial model. F-test results (not shown) similarly support the inclusion of polynomial terms but provide limited evidence for interaction effects.

**Multicollinearity Assessment**

Variance inflation factors (VIFs) were computed for all candidate models. In both the baseline and polynomial models, VIF values for all continuous predictors remain close to $1.1 - 2.0$ , indicating no problematic multicollinearity. The centring of horsepower effectively reduces collinearity between $hp_c$ and $hp_c^2$. In contrast, the interaction model exhibits inflated GVIF values for several factor variables due to the higher-order interaction structure, decreasing coefficient stability. Because the polynomial model offers better fit with lower collinearity, it is preferred on diagnostic grounds.

**Addressing Nonlinearity**

The polynomial model explicitly addresses the nonlinearities detected in exploratory analysis. The quadratic horsepower term captures diminishing returns to additional horsepower, while $\log(1 + \text{mileage})$ reflects the concave price–mileage relationship. Both transformations yield more homogeneous residual patterns and significantly improve information criteria relative to the baseline model. Although interaction terms allow usage and performance effects to vary with vehicle age, they introduce interpretational complexity and do not materially improve model fit.

### Final Model Choice and Validation

Overall, the polynomial model shows the best balance between goodness-of-fit, interpretability and diagnostic quality. It provides the lowest AIC/BIC, no multicollinearity issues, and incorporates meaningful nonlinear effects. A 30-fold cross-validation confirms strong generalization performance, with training and validation MSE values of $0.1375$ and $0.1395$, respectively. Residual plots from both training and validation folds display no remaining systematic patterns. Trimming extreme residuals produces a nearly identical coefficient structure, verifying robustness. We therefore select the polynomial model as the final specification.

## 4 Model Interpretation

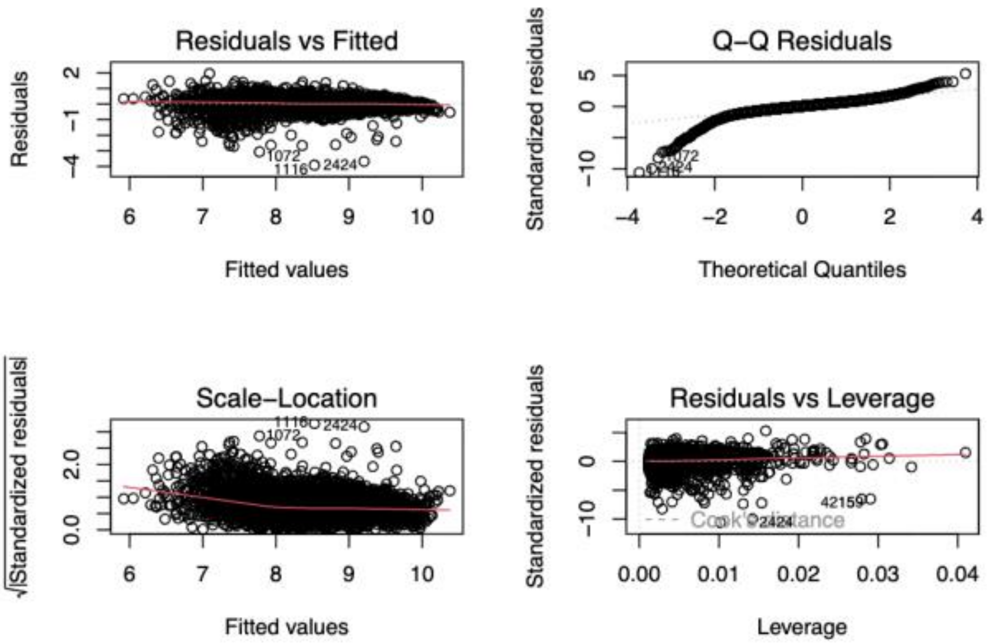
The coefficients indicate how each predictor affects the **log of price** while holding other variables constant. The positive and highly significant coefficient for **year** shows that newer cars are priced higher. **Horsepower (hp_c)** also has a strong positive effect, while its squared term is negative, indicating diminishing returns: price increases with horsepower but at a decreasing rate. The negative coefficient for **favorite** suggests that more frequently saved listings tend to be lower-priced vehicles. Categorical variables behave as expected: **Luxury brand** cars and **SUVs** have higher prices relative to the reference group, whereas **Economy** and **Family** types tend to reduce price. Variables with non-significant coefficients contribute little explanatory value.

The model's **R$^2$ of 0.7575** indicates that about **76% of the variation in car prices** is explained by the predictors, showing strong model performance. The **adjusted R$^2$ of 0.7566**, nearly identical to R$^2$, suggests that the model is not overfitted and that most predictors included in the model meaningfully improve the explanation of price variability.

The 95 percent confidence intervals indicate the possible values of each of the coefficients in the population. Most predictors do not have a zero in the interval indicating that these variables are significantly related to log-price at the 5% level. As an illustration, the confidence interval between the year (0.1099 to 0.1147) is all positive and this proves that the newer cars do raise price all along. In the same way, the intervals of the hp c, the log mileage, and the Luxury brand category are strictly positive as well, which implies consistently positive influences on the price. On the other hand, the predictors like A.C manual, type of driveback and car type Family give negative intervals implying that it substantially lowers price. Some of the predictors like car typeLimousine and car brand categoryOther contain intervals that run into zero showing that their influence are not statistically significant different than zero.

# 5 Model Diagnostics & Validation

## 5.1 Residual Plot



## 5.2 Outlier & influence analysis

The **Residuals vs Leverage** plot identifies several points with higher leverage, meaning these observations have unusual predictor combinations. A few of these points also show relatively large standardized residuals, such as those near the Cook's distance threshold, indicating potential influence on the fitted model. However, none of the points clearly exceed the Cook's distance warning boundary, suggesting that no single observation is excessively distorting the model.

## 5.3 Linearity & Constant Variance Assumption

Linearity was evaluated using the **Residuals vs Fitted** and **Q–Q plot**. The slight curvature in the residuals indicates minor violations of linearity, though the inclusion of polynomial terms has addressed much of the earlier non-linearity. The Q–Q plot shows that residuals largely follow the normality line, but deviations appear in the tails, indicating non-normality for extreme values. Constant variance is partially satisfied but not perfect, as shown by the increasing spread of residuals at higher fitted values.

## 5.4 Final Assessment of Model Adequacy

Overall, the diagnostic result suggest that the model provides a reasonable fit. The residuals are mostly well behaved, the linearity assumption is broadly satisfied, and outliers or influential points do not appear to dominate the model. While some mild heteroscedasticity and tail deviations from normality are present, these issues are not uncommon in large observational datasets and do not substantially undermine the model's interpretability or predictive usefulness. Given its high $R^2$ and stable diagnostic behavior, the model is considered adequate for explaining variation in car prices and supporting the subsequent analysis.

# 6 Conclusion & Limitation

Our study investigate on the determinants of used car prices in Serbia using a cleaned dataset that included price, mileage, year, horsepower, body type, fuel type, transmission, and brand category. After comparing several models, the polynomial specification, with a squared horsepower term and log transformed mileage, was chosen because it outperformed the baseline and the interaction models on AIC/BIC, avoided multicollinearity, and also it showed better cross-validated accuracy.

The finding indicates that newer cars, higher horsepower, luxury brands, and SUVs command higher prices, while Budget and Family segments are cheaper. The squared horsepower term captures diminishing returns to engine power, and the log-mileage variable reflects nonlinear depreciation. Overall, the model provide a clear and interpretable explanation of most of the variation in used car prices.

Several limitations remain: there are some measurement inconsistencies likely persist despite cleaning, and key determinants such as accident history, detailed condition, service records, and regional variation are missing. Regression assumptions hold only approximately, and the dataset represent a single platform and time period, limiting generalizability.

Future work should incorporate richer vehicle level information and explore more flexible modelling techniques to capture additional complexity and improve predictive performance.

## 7 R Code

```r
library(tidyverse)
library(dplyr)
library(knitr)
library(stringr)
library(purrr)
library(rlang)
library(ggplot2)
library(gridExtra)

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

```r
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"
)

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$fuel,          "fuel")
cat_table(car$car_type,      "car_type")
cat_table(car$type_of_drive, "type_of_drive")
cat_table(car$gearbox,       "gearbox")
cat_table(car$doors,         "doors")
cat_table(car$A.C,           "A.C")
cat_table(car$emission_class,"emission_class")
```

## Response

```r
# 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

```r
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()

library(GGally)
ggpairs(car[num_vars])

library(corrplot)
num_var_w_y <- c(num_vars, "price")
#compute correlation matrix
cor_matrix <- cor(car[,num_var_w_y], 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()
```

## Categorical

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

```r
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()
```

# 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

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
```

```r
car_data_cleaned= car_data_cleaned %>% filter(car_mileage..km < 5*10^5, favorite < 50,
               horsepower < 400, year >= 1970)

# 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") ~ "Luxury",
      car_brand %in% c("Ford", "Honda", "Hyundai", "Mazda", "Nissan", "Toyota", "Volkswagen", "Skoda", "Mini",
                        "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", "Smart",
                        "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))
head(car_data_cleaned)
```

# Exploratory Data Analysis

## Summary statistics & tables

```
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"
)

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_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))
dev.off()

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"
)

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

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)

library(car)
vif_values <- vif(full_model)

print(vif_values)
```

```r
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)

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)

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

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

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

### Best Baseline Model

```r
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)

par(mfrow = c(2, 2))
plot(reduced_model_3)
par(mfrow = c(1, 1))

library(sandwich)
library(lmtest)

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

##WLS

```
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)
```

## 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)
```

## Interaction model

```
int_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)^2


interaction_full <- lm(int_formula, data = car_model2)

set.seed(123)
interaction_step <- step(interaction_full,
direction = "both",
trace = FALSE)

summary(interaction_step)

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)
```

# 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]]))
}

lapply(models, AIC)

lapply(models, BIC)

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
  )
```

## 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
```

```r
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)
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")


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)
idx <- sample(1:nrow(car_model2), 30)
new_data <- car_model2[idx, ]

#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(
  id          = seq_along(idx),
  actual_price = new_data$price,
  fit_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)")

ggplot(car_model2, aes(x = year, y = log_price)) +
  geom_point(alpha = 0.2) +
  geom_smooth(
    method = "lm",
    formula = y ~ x,
    se = TRUE,       # CI
    level = 0.95
  ) +
  labs(
    title = "Log price vs year with 95% confidence band",
    x = "Year",
    y = "log(Price)"
  ) +
  theme_minimal()

# Coefficient Confidence Intervals for the final model
coef_ci <- confint(final_model)

knitr::kable(
  coef_ci,
  digits = 4,
  caption = "95% Confidence Intervals for Regression Coefficients"
)
```

### 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)
```

# Model Diagnostics

### Residual plots

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