Priprema i istraživačka analiza skupa podataka s cijenama automobila

Vukoje, Leo

Sveučilište u Rijeci
Fakultet informatike
i digitalnih tehnologija

Sveučilišni prijediplomski studij Informatika

Leo Vukoje

Priprema i istraživačka analiza skupa podataka s cijenama automobila

Završni rad

Mentor: Prof. dr. sc. Maja Matetić

Rijeka, datum 09.07.2024.

Sažetak

Ovaj završni rad bavi se pripremom i istraživačkom analizom skupa podataka o cijenama automobila, s ciljem razumijevanja faktora koji utječu na cijenu automobila. Glavno pitanje koje rad istražuje je kako različite karakteristike automobila, kao što su tip karoserije, veličina motora, potrošnja goriva i snaga motora, utječu na cijenu automobila. Svrha rada je identificirati značajne varijable koje doprinose cijeni automobila i razumjeti njihove međusobne odnose te pripremiti podatke za oblikovanje modela strojnog učenja.

Metodologija rada uključuje postupke obrade podataka (*engl. data preprocessing*), deskriptivnu i inferencijalnu statistiku. Obrada podataka obuhvaća čišćenje i transformaciju podataka kako bi se stvorio kvalitetan skup podataka za analizu. Deskriptivna statistika koristi se za pregled osnovnih karakteristika podataka, putem primjerice histograma i tablica kontingencije. Inferencijalna statistika koristi se za testiranje hipoteza i donošenje zaključaka o populaciji na temelju uzorka podataka. U radu su primijenjeni t-test, ANOVA i Chi-kvadrat test za ispitivanje značajnih razlika između grupa.

Rezultati analize pokazali su da postoje značajne razlike u cijenama automobila između različitih tipova karoserije i vrsta goriva. ANOVA test je pokazao značajne razlike u srednjim cijenama između različitih tipova karoserije, dok je Chi-kvadrat test otkrio povezanost između tipa karoserije i vrste goriva.

Ključne riječi: Statistika; Analiza podataka; Priprema podataka; Umjetna inteligencija

SADRŽAJ

1. Uvod

Tema ovog završnog rada je priprema i istraživačka analiza podataka. Skup podataka koji će biti korišten sadrži cijene automobila uz brojne podatke o automobilu koji su potencijalni faktori koji utječu na konačnu cijenu automobila. Skup podataka sadrži 25 stupaca (26. je identifikator vozila) raspoređenih na 205 redaka koji bi nakon obrade i pripreme trebali biti dovoljni za oblikovanje modela koji na temelju podataka o vozilu može predvidjeti cijenu tog vozila.

U dijelu rada u kojem će se pričati o istraživačkoj analizi biti će provedena deskriptivna analiza atributa, vizualizacija istih te pronalaženje korelacije između pojedinih podskupova podataka što bi već u ovoj fazi trebalo dati uvid u to koji podaci utječu u kojoj mjeri na konačnu cijenu jednog automobila.

Odabrana tema predstavlja znanje stečeno u području statističke analize podataka i programiranja za podatkovnu znanost, uz naglasak na korištenje programskog jezika R za odrađivanje pojedinih zadataka. S obzirom na trend umjetne inteligencije koji više ni nije predmet rasprave o budućnosti, teme poput ove postaju sve značajnije i važnije. Ono na čemu se svi modeli umjetne inteligencije temelje su upravo podaci nad kojima su ti modeli istrenirani. Da bi model mogao biti kvalitetan i upotrebljiv ključna je i kvaliteta skupa podataka koji mora biti potpun i precizan.

2. Učitavanje i uređivanje podataka

2.1. Učitavanje podataka

Skup podataka koji će biti obrađen u ovom završnom radu se sastoji od brojnih karakteristika automobila koji utječu na konačnu kupovnu cijenu automobila. Nazivi atributa i tip njihovih podataka su navedeni u tablici 1.

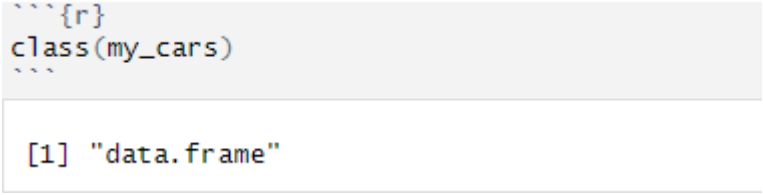
Tablica 1. Nazivi i tipovi atributa iz skupa podataka

car_ID	Numeric
symboling	Numeric
CarName	Character
fueltype	Character
aspiration	Character
doornumber	Character
carbody	Character
drivewheel	Character
enginelocation	Character
wheelbase	Numeric
carlength	Numeric
carwidth	Numeric
carheight	Numeric
curbweight	Numeric
enginetype	Character
cylindernumber	Character
enginesize	Numeric
fuelsystem	Character
boreration	Numeric
stroke	Numeric
compressionratio	Numeric
horsepower	Numeric
peakrpm	Numeric
citympg	Numeric
highwaympg	Numeric
price	Numeric

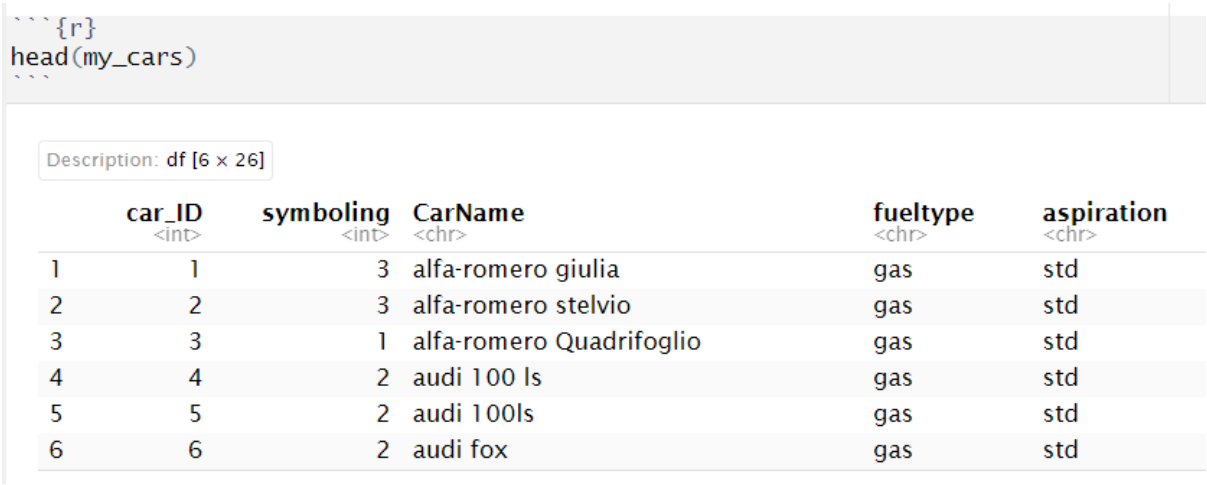
U R-u podaci se učitavaju u strukturu podataka zvanu podatkovni okvir (engl. *DataFrame*). S obzirom na to da se relevantni skup podataka nalazi u Comma Separated Values (CSV) formatu, tako se i koristi pripadajuća funkcija koja služi za uvoz podataka iz CSV datoteke u R-ov podatkovni okvir:

```
cars = read.csv("carprice.csv", header = TRUE)
```

Ovom linijom koda je u varijablu *cars* spremljen podatkovni okvir koji predstavlja skup podataka iz CSV datoteke. To se može potvrditi pozivom funkcije `class` koja vraća tip podatka spemljenog u varijablu, kao na slici 1., te sa funkcijom *head* koja (bez specificiranja dodatnih argumenata) vraća prvih 5 zapisa iz podatkovnog okvira, kao na slici 2.



Slika 1. Rezultat pozivanja funkcije *class*



	car_ID <int>	symboling <int>	CarName <chr>	fueltype <chr>	aspiration <chr>
1	1	3	alfa-romero giulia	gas	std
2	2	3	alfa-romero stelvio	gas	std
3	3	1	alfa-romero Quadrifoglio	gas	std
4	4	2	audi 100 ls	gas	std
5	5	2	audi 100ls	gas	std
6	6	2	audi fox	gas	std

Slika 2. Rezultat pozivanja funkcije *head*

2.2. Nedostajuće vrijednosti

U velikim skupovima podataka lako je moguće naići na mjesta na kojima nije unesena ili zabilježena nikakva vrijednost. U R-u takve vrijednosti se obilježavaju sa NA te R sadrži funkcije s kojima se mogu takvi retci skroz ukloniti ili nedostajuće vrijednosti zamijeniti sa nekim određenim podatkom.

Za testiranje skupa podataka na nedostajuće vrijednosti postoji funkcija `is` koja odgovara na pitanje odgovara li neka vrijednost konkretnom tipu podatka. Točnije kada bi htjeli provjeriti nedostaje li vrijednost u nekoj ćeliji bilo gdje u podatkovnom okviru može se iskoristiti funkcija `is.na(x)` gdje x predstavlja jednu konkretnu vrijednost. Ako kao parametar takvoj funkciji proslijedimo cijeli podatkovni okvir, za rezultat će se dobiti podatkovni okvir istih dimenzija, samo će umjesto konkretnih vrijednosti biti rezultat funkcije *is.na* nad svakom vrijednošću kao na slici 3.

```
{r}
is.na(my_cars[, 2:12])
```

```
      symboling brand model fueltype aspiration doornumber carbody drivewheel
enginelocation wheelbase carlength
  [1,]     FALSE FALSE FALSE    FALSE      FALSE      FALSE   FALSE      FALSE
FALSE     FALSE     FALSE
  [2,]     FALSE FALSE FALSE    FALSE      FALSE      FALSE   FALSE      FALSE
FALSE     FALSE     FALSE
  [3,]     FALSE FALSE FALSE    FALSE      FALSE      FALSE   FALSE      FALSE
FALSE     FALSE     FALSE
  [4,]     FALSE FALSE FALSE    FALSE      FALSE      FALSE   FALSE      FALSE
FALSE     FALSE     FALSE
  [5,]     FALSE FALSE FALSE    FALSE      FALSE      FALSE   FALSE      FALSE
FALSE     FALSE     FALSE
  [6,]     FALSE FALSE FALSE    FALSE      FALSE      FALSE   FALSE      FALSE
FALSE     FALSE     FALSE
  [7,]     FALSE FALSE FALSE    FALSE      FALSE      FALSE   FALSE      FALSE
FALSE     FALSE     FALSE
  [8,]     FALSE FALSE FALSE    FALSE      FALSE      FALSE   FALSE      FALSE
FALSE     FALSE     FALSE
  [9,]     FALSE FALSE FALSE    FALSE      FALSE      FALSE   FALSE      FALSE
FALSE     FALSE     FALSE
 [10,]     FALSE FALSE FALSE    FALSE      FALSE      FALSE   FALSE      FALSE
FALSE     FALSE     FALSE
 [11,]     FALSE FALSE FALSE    FALSE      FALSE      FALSE   FALSE      FALSE
FALSE     FALSE     FALSE
 [12,]     FALSE FALSE FALSE    FALSE      FALSE      FALSE   FALSE      FALSE
FALSE     FALSE     FALSE
 [13,]     FALSE FALSE FALSE    FALSE      FALSE      FALSE   FALSE      FALSE
FALSE     FALSE     FALSE
 [14,]     FALSE FALSE FALSE    FALSE      FALSE      FALSE   FALSE      FALSE
FALSE     FALSE     FALSE
 [15,]     FALSE FALSE FALSE    FALSE      FALSE      FALSE   FALSE      FALSE
FALSE     FALSE     FALSE
 [16,]     FALSE FALSE FALSE    FALSE      FALSE      FALSE   FALSE      FALSE
FALSE     FALSE     FALSE
 [17,]     FALSE FALSE FALSE    FALSE      FALSE      FALSE   FALSE      FALSE
FALSE     FALSE     FALSE
 [18,]     FALSE FALSE FALSE    FALSE      FALSE      FALSE   FALSE      FALSE
FALSE     FALSE     FALSE
 [19,]     FALSE FALSE FALSE    FALSE      FALSE      FALSE   FALSE      FALSE
FALSE     FALSE     FALSE
```

Slika 3. Rezultat funkcije *is.na* nad podatkovnim okvirom

Imajući na umu da FALSE predstavlja nule, a TRUE jedinice, zbrajanjem cijelog novonastalog podatkovnog okvira se može dobiti podatak o broju nedostajućih vrijednosti u izvornom podatkovnom okviru. Sumiranje se kao na slici 4. može ostvariti korištenjem funkcije `sum` i prosljeđivanjem podatkovnog okvira kao parametra funkcije.

```
{r}
sum(is.na(my_cars))
```

```
[1] 0
```

Slika 4. Podatak o broju nedostajućih vrijednosti u skupu podataka

Kako bi se testirao rad funkcije `is.na`, jedno rješenje je da se ručno iz *.csv* datoteke obrišu vrijednosti iz nekih ćelija te nakon toga kroz programski kôd kao na slici 5. na prazne ćelije postavi NA vrijednost.

```
{r}
my_cars[my_cars == ""] <- NA
sum(is.na(my_cars))
```

```
[1] 2
```

Slika 5. Podatak o broju nedostajućih vrijednosti u skupu podataka nakon ručnog brisanja vrijednosti

3. Deskriptivna statistika

Deskriptivna statistika podrazumijeva uređivanje, tablični i grafički prikaz podataka te izračunavanje opisnih statističkih pokazatelja. Uključuje izračunavanje mjera centralne tendencije podataka pod koje spadaju srednja vrijednost, medijan i mod, mjere varijabilnosti pod koje spadaju raspon, varijanca i standardna devijacija te distribucija frekvencije kod koje se vizualizira frekvencija varijable.

3.1. Mjere centralne tendencije i mjere varijabilnosti

U programskom jeziku R funkcija `summary` daje uvid u raspon (najmanju i najveću vrijednost), medijan, mod te vrijednosti prvog i trećeg kvartila varijable. Drugi kvartil je isto što i medijan. Na slici 6. se nalazi rezultat funkcije `summary` za sve numeričke varijable skupa podataka.

```
### Range, median, mode and quartiles for each variable except the car identifier
```{r}
summary(my_cars[(!names(my_cars) %in% c("car_ID"))] %>% select(where(is.numeric)))
```
```

```
  symboling         wheelbase        carlength        carwidth        carheight       curbweight   
 Min.   :-2.0000   Min.   : 86.60   Min.   :141.1   Min.   :60.30   Min.   :47.80   Min.   :1488  
 1st Qu.: 0.0000   1st Qu.: 94.50   1st Qu.:166.3   1st Qu.:64.10   1st Qu.:52.00   1st Qu.:2145  
 Median : 1.0000   Median : 97.00   Median :173.2   Median :65.50   Median :54.10   Median :2414  
 Mean   : 0.8341   Mean   : 98.76   Mean   :174.0   Mean   :65.91   Mean   :53.72   Mean   :2556  
 3rd Qu.: 2.0000   3rd Qu.:102.40   3rd Qu.:183.1   3rd Qu.:66.90   3rd Qu.:55.50   3rd Qu.:2935  
 Max.   : 3.0000   Max.   :120.90   Max.   :208.1   Max.   :72.30   Max.   :59.80   Max.   :4066  
   enginesize      boreratio        stroke      compressionratio   horsepower       peakrpm     
 Min.   : 61.0   Min.   :2.54   Min.   :2.070   Min.   : 7.00    Min.   : 48.0   Min.   :4150  
 1st Qu.: 97.0   1st Qu.:3.15   1st Qu.:3.110   1st Qu.: 8.60    1st Qu.: 70.0   1st Qu.:4800  
 Median :120.0   Median :3.31   Median :3.290   Median : 9.00    Median : 95.0   Median :5200  
 Mean   :126.9   Mean   :3.33   Mean   :3.255   Mean   :10.14    Mean   :104.1   Mean   :5125  
 3rd Qu.:141.0   3rd Qu.:3.58   3rd Qu.:3.410   3rd Qu.: 9.40    3rd Qu.:116.0   3rd Qu.:5500  
 Max.   :326.0   Max.   :3.94   Max.   :4.170   Max.   :23.00    Max.   :288.0   Max.   :6600  
    citympg       highwaympg        price      
 Min.   :13.00   Min.   :16.00   Min.   : 5118  
 1st Qu.:19.00   1st Qu.:25.00   1st Qu.: 7788  
 Median :24.00   Median :30.00   Median :10295  
 Mean   :25.22   Mean   :30.75   Mean   :13277  
 3rd Qu.:30.00   3rd Qu.:34.00   3rd Qu.:16503  
 Max.   :49.00   Max.   :54.00   Max.   :45400  
```

Slika 6. Raspon, medijan, mod i kvartili za svaku varijablu osim identifikatora

Funkcija `summary` pokriva sve mjere centralne tendencije, ali ne i sve mjere varijabilnosti. Standardna devijacija i varijanca se mogu izračunati zasebno. Za standardnu devijaciju se može iskoristiti funkcija `sd` koja prima listu brojeva, a može se i kombinirati sa funkcijom sapply tako da se izračuna standardna devijacija svakog stupca u podatkovnom okviru. Varijanca se može izračunati na isti način kao i standardna devijacija, samo se za varijancu koristi funkcija `var` umjesto `sd`. Na slici 7. su rezultati izračuna za standardnu devijaciju i varijancu numeričkih varijabli (osim identifikatora vozila).

```
### Standard deviation
```{r}
sapply(my_cars[(!names(my_cars) %in% c("car_ID"))] %>% select(where(is.numeric)),_sd)
```
```
      symboling       wheelbase       carlength        carwidth       carheight      curbweight
      1.2453068       6.0217757      12.3372885       2.1452039       2.4435220     520.6802035
     enginesize       boreratio          stroke compressionratio      horsepower        peakrpm
     41.6426934       0.2708437       0.3135970       3.9720403      39.5441668     476.9856431
        citympg      highwaympg           price
      6.5421417       6.8864431    7988.8523317
```
```
### Variance
```{r}
sapply(my_cars[(!names(my_cars) %in% c("car_ID"))] %>% select(where(is.numeric)), var)
```
```
      symboling       wheelbase       carlength        carwidth       carheight      curbweight
   1.550789e+00    3.626178e+01    1.522087e+02    4.601900e+00    5.970800e+00    2.711079e+05
     enginesize       boreratio          stroke compressionratio      horsepower        peakrpm
   1.734114e+03    7.335631e-02    9.834309e-02    1.577710e+01    1.563741e+03    2.275153e+05
        citympg      highwaympg           price
   4.279962e+01    4.742310e+01    6.382176e+07
```

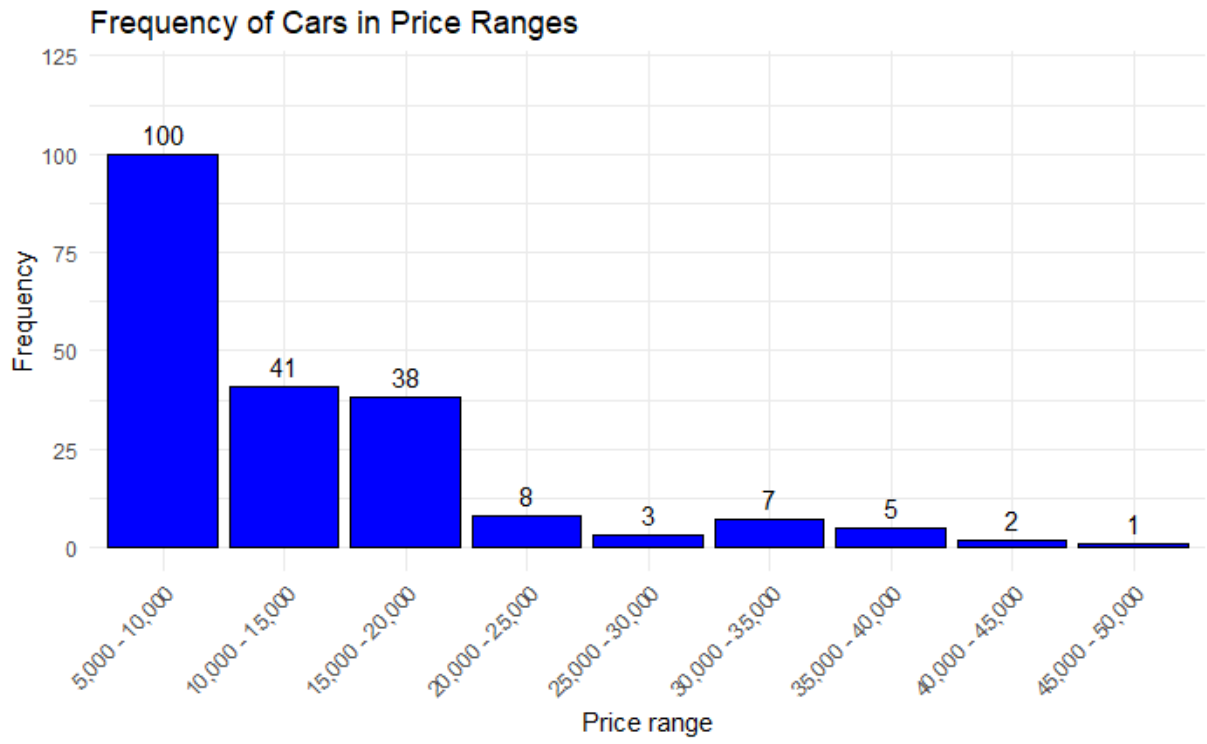
Slika 7. Standardne devijacije i varijance numeričkih varijabli skupa podataka

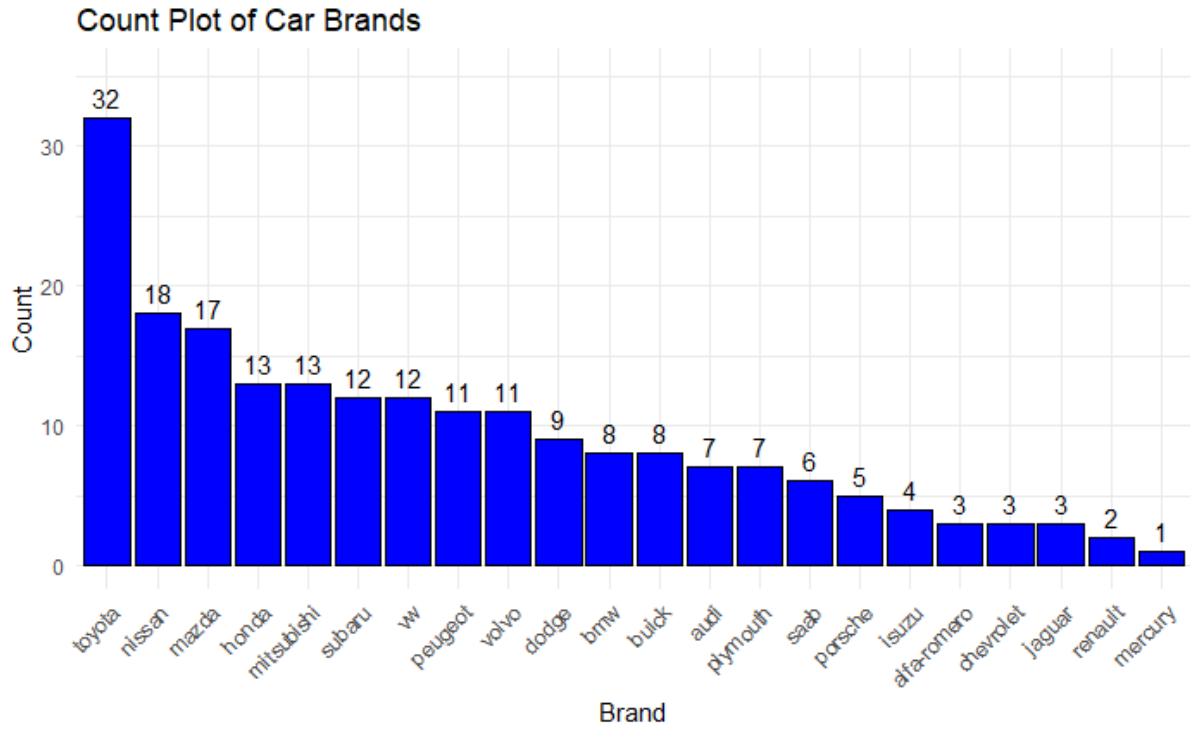
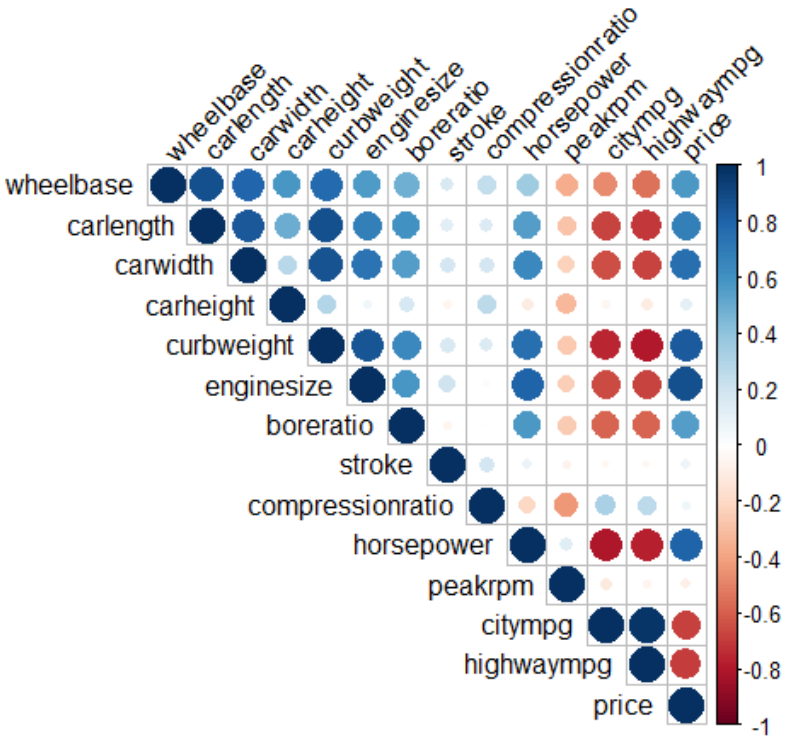
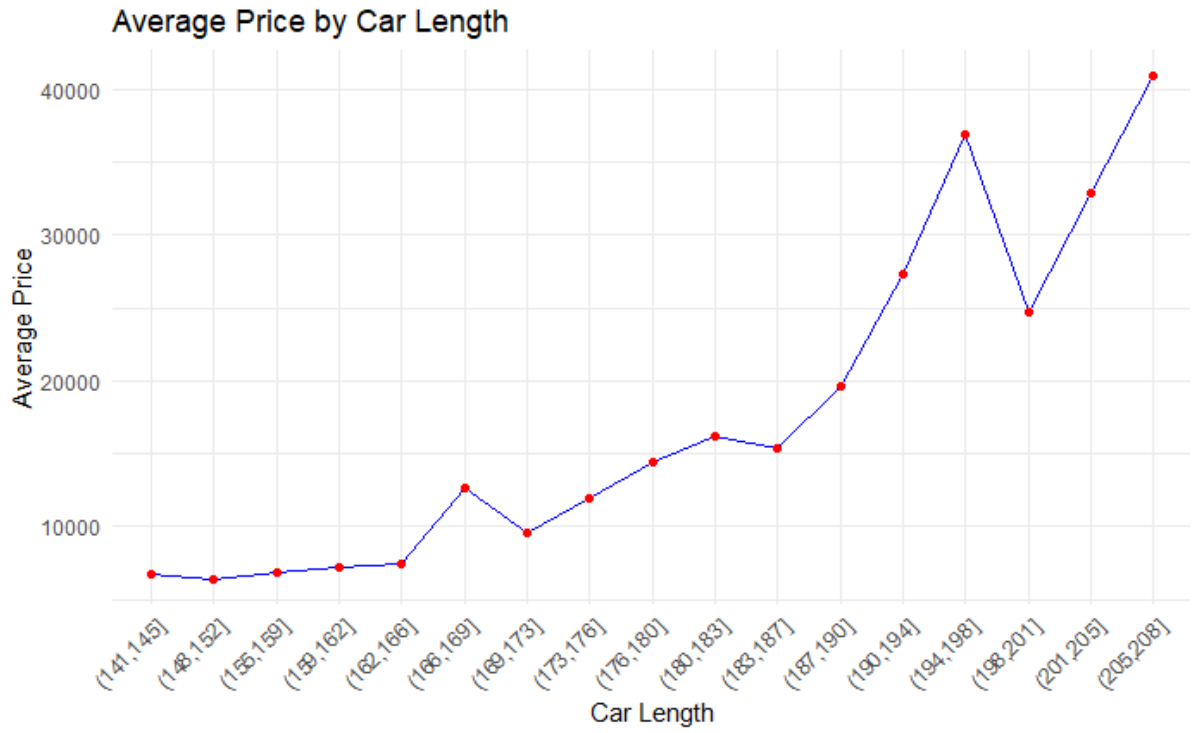
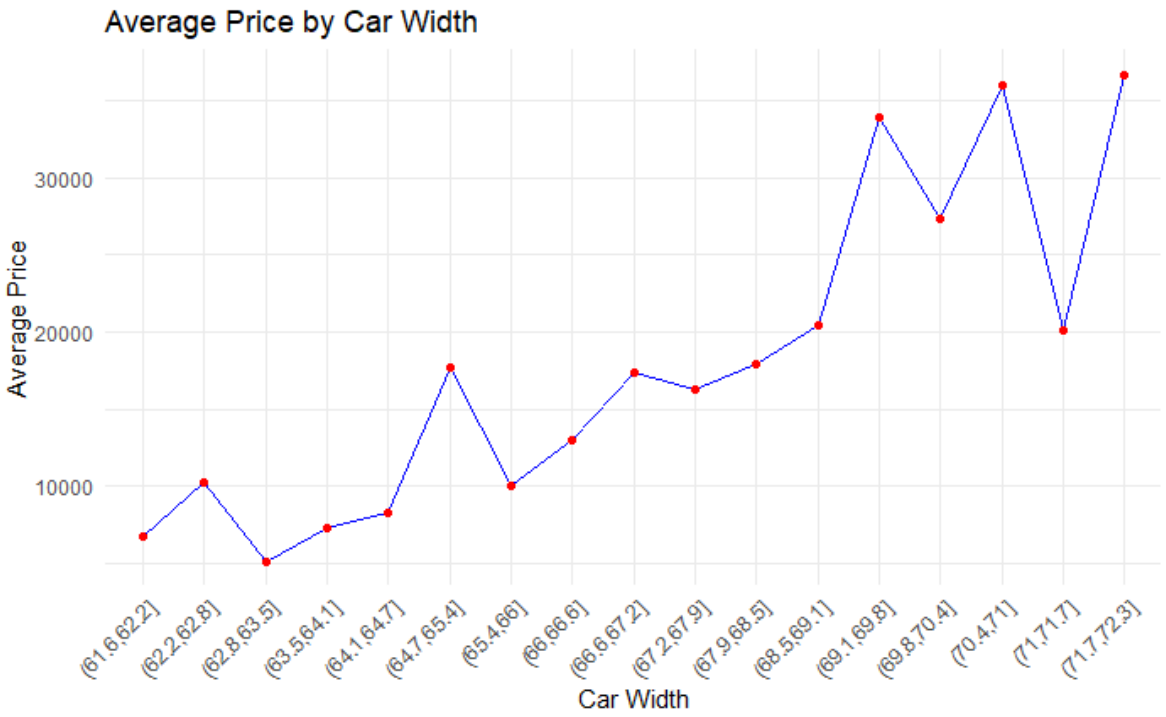
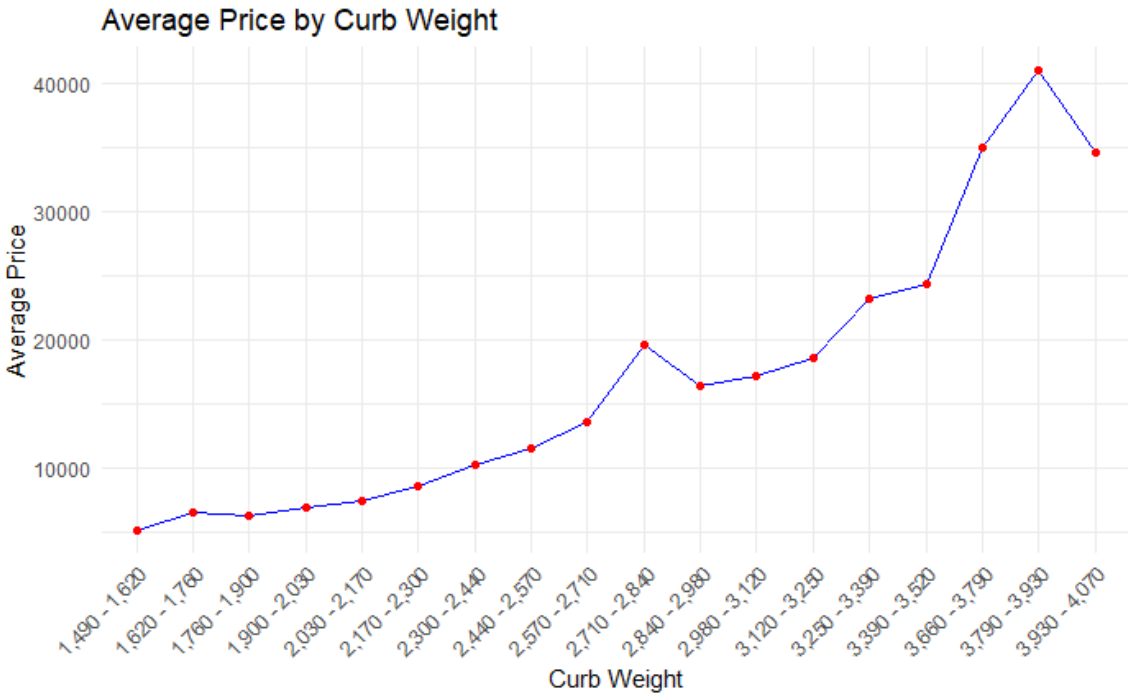
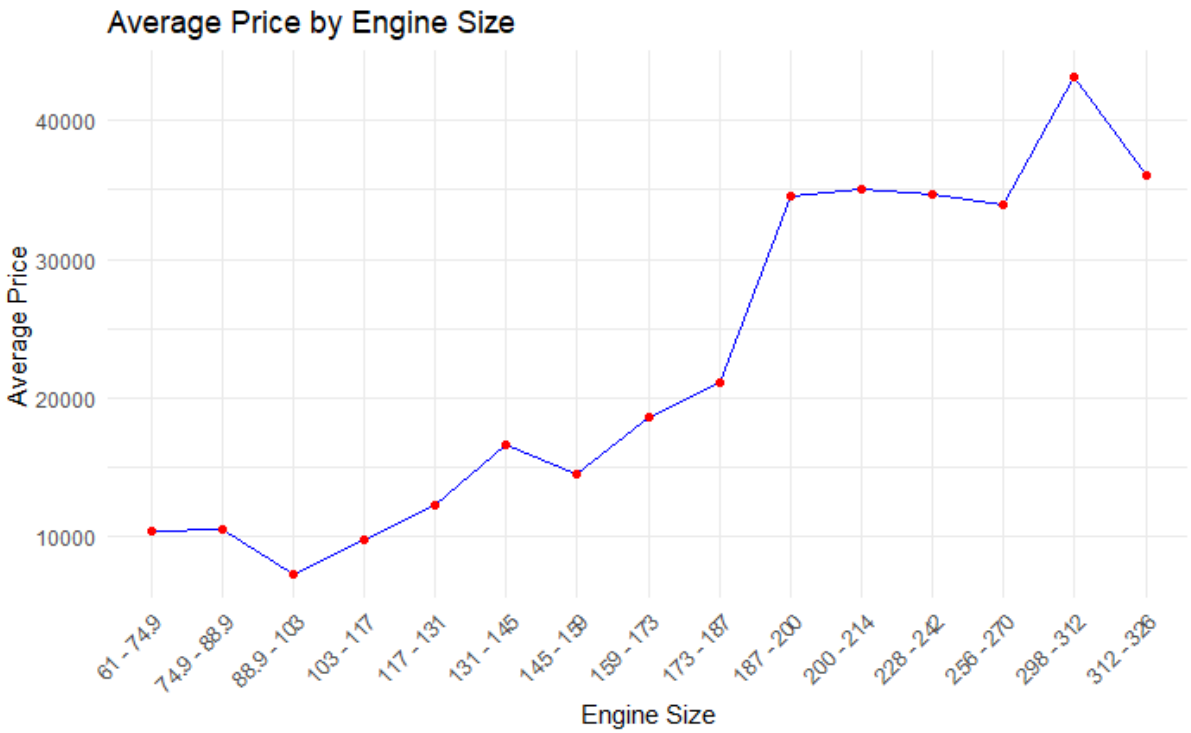
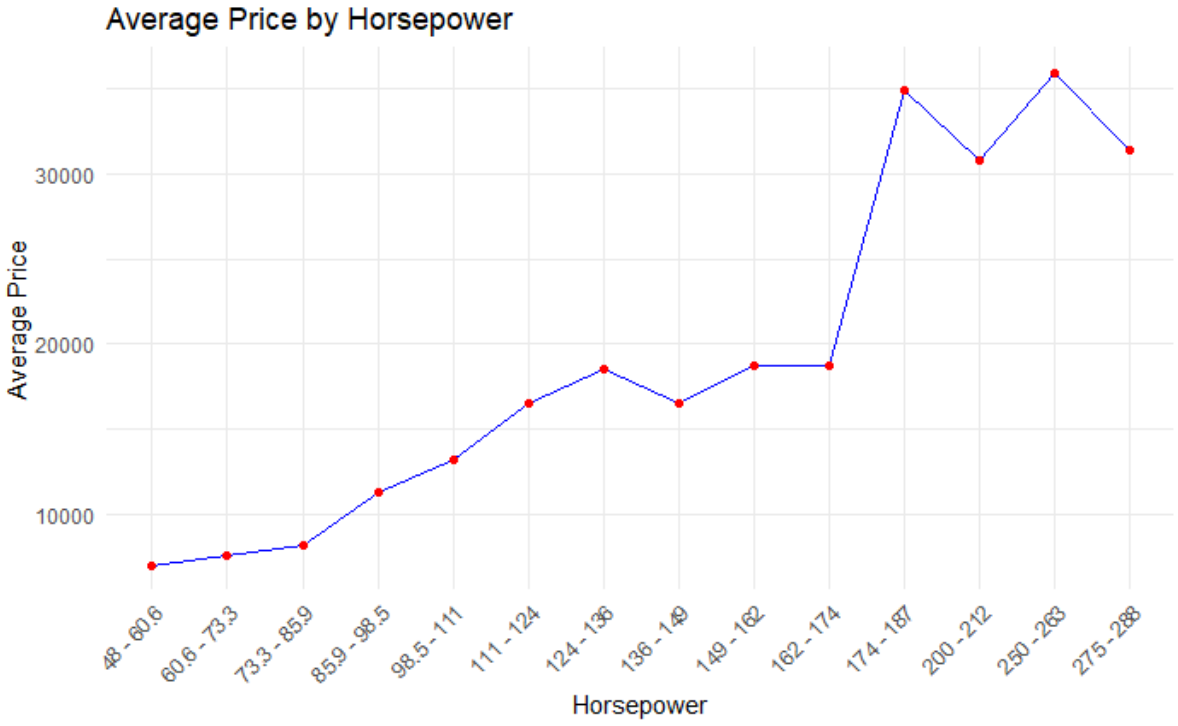
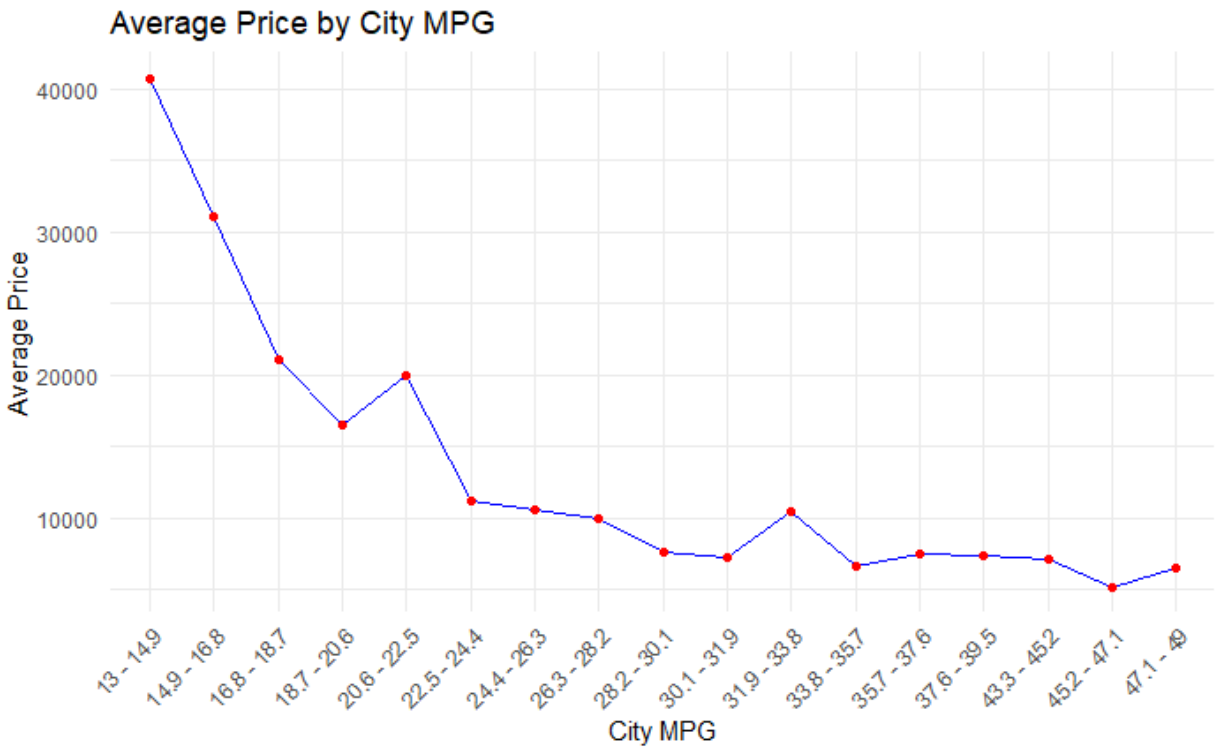
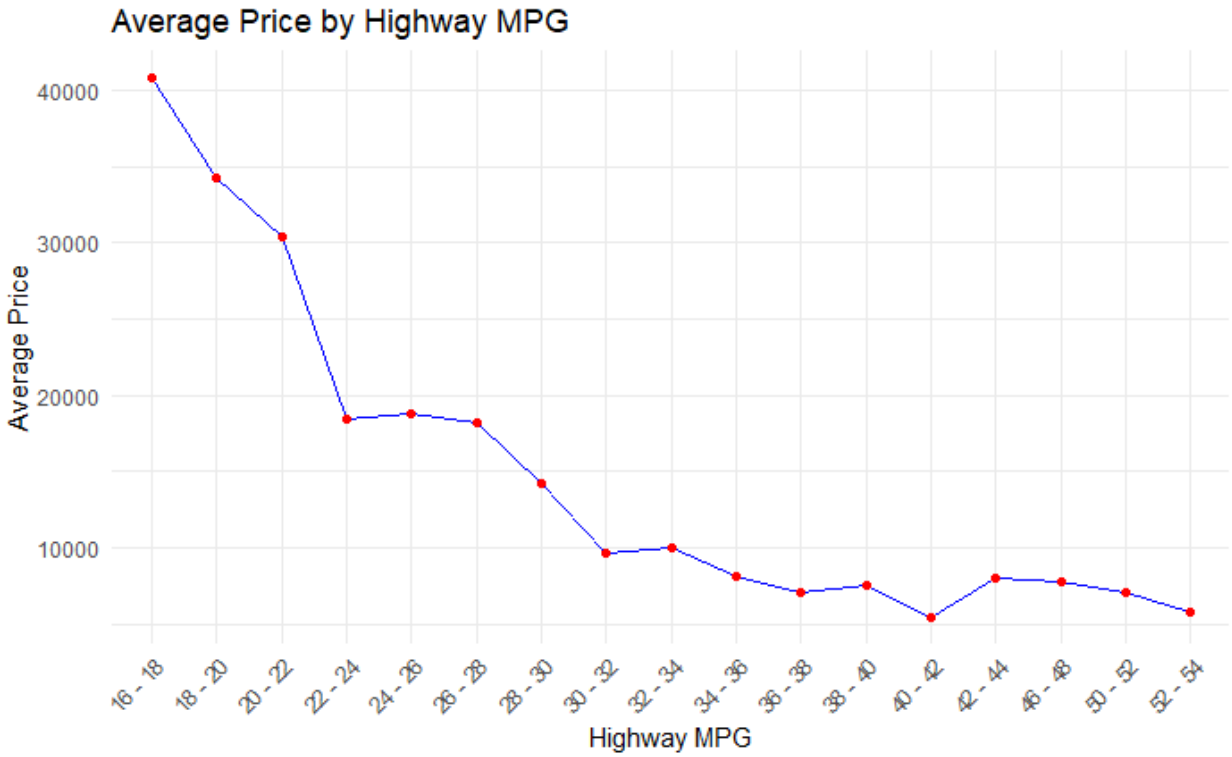
3.2. Distribucija podataka

Kod distribucije podataka cilj je pomoću grafova vizualizirati raspršenost individualnih numeričkih varijabli i parove varijabli, kategoričke i numeričke ili numeričke i numeričke, kako bi se identificirale potencijalne korelacije među varijablama.

3.2.1. Frekvencija automobila u cjenovnim rasponima

Kod sa slike 8. stvara histogram koji prikazuje frekvenciju automobila u određenim rasponima cijena. Da bi se stvorio takav histogram, prvo se definira raspon cijena, zatim kreira nova varijabla koja grupira automobile unutar tih raspona, izračunava broj automobila u svakom rasponu i na kraju generira histogram s pripadajućim oznakama.

```
## Price Range Frequency Histogram
```{r}
price_breaks <- seq(0, max(my_cars$price + 5000, na.rm = TRUE), by = 5000)

my_cars <- my_cars %>%
  mutate(price_range = cut(price, breaks = price_breaks, include.lowest = TRUE, right = FALSE))

price_range_counts <- my_cars %>%
  group_by(price_range) %>%
  summarize(count = n())

format_price_range <- function(labels) {
  sapply(labels, function(x) {
    range <- gsub("\\[|\\)|\\]", "", x)
    range <- strsplit(range, ",")[[1]]
    paste0(format(as.numeric(range[1]), big.mark = ","), " - ", format(as.numeric(range[2]), big.mark = ","))
  })
}

max_count <- max(price_range_counts$count)

ggplot(price_range_counts, aes(x = price_range, y = count)) +
  geom_bar(stat = "identity", fill = "blue", color = "black") +
  geom_text(aes(label = count), vjust = -0.5, color = "black") +
  theme_minimal() +
  labs(title = "Frequency of Cars in Price Ranges",
       x = "Price range",
       y = "Frequency") +
  scale_x_discrete(labels = format_price_range(levels(price_range_counts$price_range))) +
  scale_y_continuous(limits = c(0, max_count * 1.2)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```
```

Slika 8. Kod za grafikon cijena automobila raspoređenih po rasponima cijena

```
price_breaks <- seq(0, max(my_cars$price + 5000, na.rm = TRUE), by = 5000)
```

Ova linija stvara sekvencu raspona cijena koja počinje od 0 pa sve do maksimalne cijene automobila u skupu podataka, povećane za 5000. Koristi `seq` funkciju s korakom od 5000 kako bi stvorila intervale od po 5000 jedinica.

```
my_cars <- my_cars %>%
  mutate(price_range = cut(price, breaks = price_breaks, include.lowest = TRUE, right =
FALSE))
```

Ova linija dodaje novu varijablu `price_range` u skup podataka `my_cars`. Varijabla `price_range` grupira cijene automobila unutar definiranih raspona cijena (`price_breaks`) pomoću funkcije `cut`.

```
price_range_counts <- my_cars %>%
  group_by(price_range) %>%
  summarize(count = n())
```

Ova linija grupira automobile prema novostvorenoj varijabli `price_range` i izračunava broj automobila (`count`) unutar svakog raspona cijena. Rezultat je novi podatkovni okvir `price_range_counts` koji sadrži frekvenciju automobila po rasponima cijena.

```
format_price_range <- function(labels) {
  sapply(labels, function(x) {
    range <- gsub("\\[|\\)|\\]", "", x)
    range <- strsplit(range, ",")[[1]]
    paste0(format(as.numeric(range[1]), big.mark = ","), " - ",
format(as.numeric(range[2]), big.mark = ","))
  })
}
```

Ova funkcija formatira oznake raspona cijena kako bi bile čitljivije. Zamjenjuje zagrade s praznim znakovnim nizom i razdvaja raspon cijena u dva dijela (donji i gornji), a zatim ih formatira sa separatorom za tisućice.

```
max_count <- max(price_range_counts$count)
```

Ova linija izračunava maksimalnu frekvenciju automobila u bilo kojem rasponu cijena, što će se koristiti za podešavanje y-osi u histogramu.

```
ggplot(price_range_counts, aes(x = price_range, y = count)) +
  geom_bar(stat = "identity", fill = "blue", color = "black") +
  geom_text(aes(label = count), vjust = -0.5, color = "black") +
  theme_minimal() +
  labs(title = "Frequency of Cars in Price Ranges",
       x = "Price range",
       y = "Frequency") +
  scale_x_discrete(labels = format_price_range(levels(price_range_counts$price_range))) +
  scale_y_continuous(limits = c(0, max_count * 1.2)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

Ovaj dio koda koristi *ggplot2* paket za kreiranje histograma koji prikazuje frekvenciju automobila u različitim rasponima cijena. Postavlja varijable za x i y osi, kreira stupce s plavim ispunama i crnim rubovima, te dodaje oznake s brojem automobila iznad svakog stupca. Primjenjuje se minimalni stil, dodaje se naslov i oznake osi, koriste se prilagođene oznake za x-os s formatiranjem raspona cijena, podešava se y-os kako bi se dodalo prostora iznad najvišeg stupca, te se rotiraju oznake na x-osi za bolju čitljivost.



Slika 9. Stupčasti grafikon cijena automobila raspoređen po rasponima od 5000 jedinica

Iz grafikona sa slike 9. se vidi raspršenost cijena automobila iz skupa podataka: što se cijena diže to je manje automobila u skupu podataka.

3.2.2. Zastupljenost marki automobila

Kod sa slike 10. stvara stupčasti grafikon koji prikazuje frekvenciju automobila prema brendu, gdje je brend s najviše pojavljivanja prvi na x-osi. Da bi se stvorio takav histogram, prvo se izračunava frekvencija svakog brenda, zatim sortiraju brendovi prema frekvenciji i koristi taj redoslijed za kreiranje grafikona.

```
## Frequency of Cars per Brand
```{r warning=FALSE}
brand_counts <- my_cars %>%
  count(brand) %>%
  arrange(desc(n))

my_cars$brand <- factor(my_cars$brand, levels = brand_counts$brand)

ggplot(my_cars, aes(x = brand)) +
  geom_bar(fill = "blue", color = "black") +
  geom_text(stat = 'count', aes(label = ..count..), vjust = -0.5, color = "black") +
  theme_minimal() +
  labs(title = "Count Plot of Car Brands", x = "Brand", y = "Count") +
  scale_y_continuous(limits = c(0, max(brand_counts$n) * 1.1)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```
```

Slika 10. Stupčasti grafikon cijena automobila raspoređen po rasponima od 5000 jedinica

```
brand_counts <- my_cars %>%
  count(brand) %>%
  arrange(desc(n))

my_cars$brand <- factor(my_cars$brand, levels = brand_counts$brand)
```

Ove linije koda prvo izračunavaju koliko se puta svaki brend pojavljuje u skupu podataka koristeći `count(brand)`, zatim silazno sortiraju brendove prema broju pojavljivanja (`arrange(desc(n))`). Nakon toga, varijabla `brand` u originalnom skupu podataka se ređava prema tim sortiranim frekvencijama koristeći `factor`.

```
ggplot(my_cars, aes(x = brand)) +
  geom_bar(fill = "blue", color = "black") +
  geom_text(stat = 'count', aes(label = ..count..), vjust = -0.5, color = "black") +
  theme_minimal() +
  labs(title = "Count Plot of Car Brands", x = "Brand", y = "Count") +
  scale_y_continuous(limits = c(0, max(brand_counts$n) * 1.1)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

Ova linija kreira stupčasti grafikon koji prikazuje broj automobila za svaki brend, gdje se brendovi nalaze na x-osi, a frekvencija na y-osi. Grafikon ima plave stupce s crnim rubovima, minimalni stil te rotirane oznake na x-osi za bolju čitljivost.

Sa grafikona na slici 11. vidljivo je da su redom najzastupljenije marke automobila: Toyota, Nissan, Mazda, Honda, Mitsubishi, itd.



Slika 11. Stupčasti grafikon sa brojem automobila po brendu

3.2.3. Pronalažene korelacija među varijablama

Korelacija je statistička mjera koja opisuje snagu i smjer linearne veze između dvije varijable. Korelacijski koeficijent može imati vrijednosti između -1 i 1, gdje:

- 1 označava savršenu pozitivnu linearnu korelaciju (obje varijable rastu).
- -1 označava savršenu negativnu linearnu korelaciju (kako jedna varijabla raste, druga opada).
- 0 označava da nema linearne korelacije između varijabli.

U R-u postoji paket `corrplot` koji omogućuje vizualizacija korelacijskih matrica i olakšava interpretaciju korelacijskih koeficijenata između više varijabli.

Prva linija koda na slici 12. izdvaja sve numeričke vrijednosti potencijalno relevantne za algoritam strojnog učenja. U drugoj liniji sa funkcijom `cor` se izračunava korelacijska matrica za izdvojene varijable. Funkcija koristi `complete.obs` opciju kako bi se osiguralo da se korelacije izračunavaju samo na temelju kompletnih parova podataka (ignorirajući redove s nedostajućim vrijednostima). Zadnja linija koda koristi funkciju `corrplot` iz paketa `corrplot` za vizualizaciju korelacijske matrice. Metoda *"circle"* prikazuje korelacije pomoću krugova čija veličina i boja predstavljaju jačinu korelacije. Opcija `type = "upper"` prikazuje samo gornji trokut korelacijske matrice, `tl.col = "black"` postavlja boju oznaka (tekstova) na crnu, a `tl.srt = 45` rotira oznake za 45 stupnjeva radi bolje čitljivosti.

```
## Correlational Matrix
```{r}
numerical_vars <- my_cars %>% select_if(is.numeric) %>% select(-car_ID, -symboling)

cor_matrix <- cor(numerical_vars, use = "complete.obs")

corrplot(cor_matrix, method = "circle", type = "upper", tl.col = "black", tl.srt = 45)
```
```

Slika 12. Kod za vizualizaciju korelacijske matrice

Iz grafičkog prikaza korelacijske matrice na slici 13. može se vidjeti koje varijable utječu na koju. Na primjeru zavisne varijable *price* vidi se da je u najvećoj korelaciji sa varijablama *carlength*, *carwidth*, *curbweight*, *enginesize*, *horsepower* te *citympg* i *highwaympg* koje imaju negativnu linearnu korelaciju.



Slika 13. Vizualizacija korelacijske matrice

Tablica 2. Parovi varijabli čija je apsolutna vrijednost korelacije sa varijablom cijene veća od 0.6

price	carlength	0.68292
price	carwidth	0.759325
price	curbweight	0.835305
price	enginesize	0.874145
price	horsepower	0.808139
price	citympg	-0.68575
price	highwaympg	-0.6976

Kako bi se jasno prikazao rast ili pad cijene s obzirom na neku varijablu, može se kreirati linijski graf koji bi prikazivao prosječnu cijenu automobila za neku konkretnu vrijednost varijable ili neki manji raspon vrijednosti. Tako se na primjer na slici 14. vidi da duljina auta proporcionalno utječe na njegovu konačnu cijenu što potvrđuje i koeficijent korelacije od 0.68292 očitan iz tablice 2.



Slika 14. Prosječna cijena automobila po njegovoj duljini

Isto se može potvrditi i za ostale parove varijabli iz tablice 2. Na slikama 15., 16., 17. i 18. također cijena raste proporcionalno sa nekom varijablom. Na slici 15. to su cijena i širina auta koji imaju koeficijent korelacije 0.759325, na slici 16. cijena i masa praznog vozila sa koeficijentom korelacije 0.835305, na slici 17. cijena i veličina motora sa koeficijentom korelacije od 0.874145 te na slici 18. cijena i konjska snaga automobila proporcionalno rastu uz koeficijent korelacije od 0.808139.

Za varijable o potrošnji goriva u gradskim uvjetima i potrošnji goriva na autocesti se vidi iz tablice 2. da imaju negativan koeficijent korelacije sa cijenom. Na slikama 19. i 20. se može to i potvrditi jer cijena pada kako se vrijednost navedenih dviju varijabli povećava.



Slika 15. Prosječna cijena automobila po njegovoj širini



Slika 16. Prosječna cijena automobila po njegovoj težini



Slika 17. Prosječna cijena automobila po veličini motora



Slika 18. Prosječna cijena automobila po konjskoj snazi



Slika 19. Prosječna cijena automobila po gradskoj potrošnji goriva



Slika 20. Prosječna cijena automobila po potrošnji goriva na autocesti

4. Inferencijalna statistika

Inferencijalna statistika omogućuje donošenje zaključaka o populaciji na temelju uzorka podataka. Uključuje metode poput testiranja hipoteza (t-test, ANOVA, chi-square test) i izračunavanje intervala pouzdanosti.

4.1. Testiranje hipoteza

Testiranje hipoteza je statistički postupak koji se koristi za donošenje zaključaka o populaciji na temelju uzorka podataka. Osnovna ideja je postaviti dvije konkurentske hipoteze: nultu hipotezu (H0), koja kaže da ne postoji statistički značajna razlika među skupinama, i alternativnu hipotezu (H1), koja se odnosi na tvrdnju da postoji statistički značajna razlika između skupina.

4.1.1. T-test

T-test je statistički test koji se koristi za usporedbu srednjih vrijednosti dvije skupine podataka i koristi se kada se želi utvrditi postojanje razlike između njihovih srednjih vrijednosti. U programskom jeziku R, T-test se može provesti sa funkcijom `t.test()` koja vraća nekoliko podataka vezano za dvije promatrane skupine podataka.

```
### T-test
```{r}
t_test_result <- t.test(price ~ fueltype, data = my_cars)
t_test_result
```
```

```
        Welch Two Sample t-test

data:  price by fueltype
t = 1.5495, df = 23.566, p-value = 0.1346
alternative hypothesis: true difference in means between group diesel and
group gas is not equal to 0
95 percent confidence interval:
 -945.9072 6622.6108
sample estimates:
mean in group diesel    mean in group gas
            15838.15             12999.80
```

Slika 21. T-test varijabli *fueltype* i *price*

Rezultat funkcije `t.test()` na slici 21. daje nekoliko podataka kojima se može interpretirati odnos između dvije skupine podataka. Najbitnija je p-vrijednost koja govori koja hipoteza se prihvaća, a koja odbacuje. P-vrijednost od 0.1346 je veća od uobičajene razine značajnosti (npr. 0.05) što znači da se prihvaća nulta hipoteza, odnosno ona koja govori da nema statistički značajne razlike među skupinama podataka.

Interval pouzdanosti u ovom slučaju predstavlja raspon vrijednosti unutar kojeg bi se stvarna razlika između srednjih cijena automobila za grupe s dizelskim i benzinskim motorom očekivala s određenom razinom pouzdanosti (u ovom slučaju 95%). U ovom slučaju taj interval je jako širok što znači da postoji velika nesigurnost oko procijenjene razlike u srednjim

cijenama između dizelskih i benzinskih automobila. Interval također sadrži i negativne i pozitivne vrijednosti, što znači da postoji slučaj gdje su automobili na benzin skuplji od automobila na dizel i obrnuto. Također interval sadrži i nulu što znači da se može dogoditi da nema razlike u cijeni između dizelskih automobila i benzinskih. S obzirom na to da *fueltype* ima samo dvije kategorije, interval pouzdanosti se može izračunati sa t-testom, u suprotnom bi se morao koristiti ANOVA test.

4.1.2. ANOVA test

ANOVA test je statistički test koji se koristi za uspoređivanje srednjih vrijednosti više od dvije nezavisne skupine podataka. ANOVA test može biti jednofaktorska ili višefaktorska: jednofaktorska je u slučaju da postoji jedna varijabla, ali više od dvije kategorije, a višefaktorska je kada postoje dvije ili više varijable. ANOVA testira nultu hipotezu (H0) koja kaže da su sve srednje vrijednosti jednake, protiv alternativne hipoteze (H1) koja kaže da barem jedna srednja vrijednost odstupa.

```
### ANOVA test
```{r}
anova_result <- aov(price ~ carbody, data = my_cars)
summary(anova_result)
```
```

```
             Df    Sum Sq   Mean Sq F value   Pr(>F)
carbody       4 1.802e+09 450499206   8.032 5.03e-06 ***
Residuals   200 1.122e+10  56088213
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Slika 22. ANOVA test varijable *carbody* u odnosu na cijenu automobila

U ANOVA testu je relevantna F vrijednost koja se dobiva dijeljenjem varijabilnosti između skupina sa varijabilnošću unutar skupina i koja se koristi za provjeru hipoteze o jednakosti srednjih vrijednosti između skupina. Ako je F-vrijednost visoka to znači da je varijabilnost između grupa mnogo veća od varijabilnosti unutar grupa. To ukazuje na činjenicu da su razlike između grupa statistički značajne. U rezultatu ANOVA testa sa slike 22. F-vrijednost je 8.032 što ukazuje na statistički značajnu razliku između grupa.

Pr(>F) označava p-vrijednost ANOVA testa i u slučaju da je ona manja od odabranog nivoa značajnosti (npr. 0.05) onda se može zaključiti da postoje statistički značajne razlike između barem dvije od tri skupine podataka. U rezultatu sa slike 22. p-vrijednost je vrlo blizu nuli (0.00000503) što znači da postoje statistički značajne razlike među skupinama.

4.1.3. Chi-kvadrat test

Chi-kvadrat test se koristi za testiranje nezavisnosti između dvije varijable. Uspoređuje očekivanu frekvenciju pojavljivanja neke pojave sa stvarnom frekvencijom, i utvrđuje postoje li statistički značajne razlike između očekivane i stvarne frekvencije.

Tablica kontingencije, poznata i kao križna tablica, je dvodimenzionalna tablica koja prikazuje frekvencije (broj pojavljivanja) dviju kategorijskih varijabli. U kontekstu Chi-

kvadrat testa, tablica kontingencije omogućava pregled odnosa između dviju varijabli te izračunavanje očekivanih frekvencija za testiranje hipoteza o njihovoj povezanosti ili neovisnosti. Tablica kontigencije se u R-u može dobiti pozivom funkcije `table()` i prosljeđivanjem varijabli za koje se želi dobiti tablica.

Tablica 3. Tablica kontingencije varijabli *carbody* i *fueltype*

	diesel	gas
convertible	0	6
hardtop	1	7
hatchback	1	69
sedan	15	81
wagon	3	22

```
### Chi-square test
```{r}
chisq.test(table(my_cars$carbody, my_cars$fueltype))
```

Warning: Chi-squared approximation may be incorrect
        Pearson's Chi-squared test

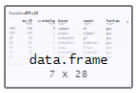
data: table(my_cars$carbody, my_cars$fueltype)
X-squared = 10.129, df = 4, p-value = 0.0383
```

Slika 23. Rezultat chi-kvadrat testa varijabli *carbody* i *fueltype*

U rezultatu sa slike 23. p-vrijednost je 0.0383, što je manje od 0.05, pa se može zaključiti da postoji statistički značajna povezanost između tipa karoserije automobila i vrste goriva.

4.2. Interval pouzdanosti

Interval pouzdanosti se koristi za procjenu raspona vrijednosti u kojem se s određenom vjerojatnošću nalazi nepoznati parametar populacije (na primjer, srednju vrijednost). Interval pouzdanosti obično se izražava kao niz vrijednosti s donjom i gornjom granicom.

Za izračun intervala pouzdanosti se može koristiti ista funkcija kao i za izvođenje T-testa. Na slici 21. se vidi da ta funkcija također vraća i 95% interval pouzdanosti za srednju vrijednost proslijeđene skupine podataka.

```
## Confidence Intervals
```{r}
t_test_result <- t.test(my_cars$price)
t_test_result$conf.int
```

[1] 12176.59 14376.83
attr(,"conf.level")
[1] 0.95
```

Slika 24. 95% interval pouzdanosti za srednju vrijednost cijene automobila

Interpretacija rezultata sa slike 24. glasi da će se srednja vrijednost cijene automobila sa sigurnošću od 95% nalaziti u intervalu od 12176.59 do 14376.83.

Za izračun intervala pouzdanosti nekog drugog postotka sigurnosti postoji paket *DescTools* u koji se može proslijediti željeni nivo sigurnosti.

```
```{r}
MeanCI(my_cars$price, conf.level = 0.95)
MeanCI(my_cars$price, conf.level = 0.70)
MeanCI(my_cars$price, conf.level = 0.98)
```

    mean   lwr.ci   upr.ci
13276.71 12176.59 14376.83
    mean   lwr.ci   upr.ci
13276.71 12696.94 13856.48
    mean   lwr.ci   upr.ci
13276.71 11968.41 14585.01
```

Slika 25. Intervali pouzdanosti za srednju vrijednost cijene automobila sa različitim postotcima sigurnosti

Na slici 25. se vidi rezultat funkcije `MeanCI` iz paketa *DescTools* koja vraća interval sigurnosti za srednju vrijednost automobila po željenom postotku sigurnosti. Tako se u prvom rezultatu za 95% interval pouzdanosti vidi da je isti kao na slici 21., dok je drugi rezultat vrati uži interval s obzirom na je postotak sigurnosti manji, a zadnji rezultat za 98% interval pouzdanosti je vratio najširi interval. Tako bi interpretacije rezultata sa slike 25. glasile:

- Srednja vrijednost cijene automobila se sa 70% vjerojatnosti nalazi između 12696.94 i 13856.48
- Srednja vrijednost cijene automobila se sa 98% vjerojatnosti nalazi između 11968.41 i 14585.01

5. Transformacija podataka

5.1. Normalizacija i skaliranje (standardizacija)

Kako u skupu podataka postoji mnogo stupaca sa numeričkim vrijednostima, dobra praksa je numeričke vrijednosti normalizirati ili skalirati. Termini normalizacija i skaliranje se odnose na postupke prilagođavanja numeričkih podataka, normalizacija radi tako da sve vrijednosti dovede u interval između 0 i 1 dok skaliranje radi na isti princip samo što se konačni brojevi ne moraju nužno nalaziti između 0 i 1. Bitno je da razlika između vrijednosti ostane proporcionalna onoj koja je bila u originalnom skupu.

Takvi postupci mogu pridonijeti performansama algoritma strojnog učenja u slučaju da algoritam mora računati udaljenost, npr. algoritmi grupiranja koji svrstavaju podatak u najbližu grupu, odnosno grupu koja sadrži njemu najsličnije podatke. Formule koje se koriste za udaljenost su osjetljive na raspon dobivenih podataka tako da se skaliranjem može dobiti na brzini izvođenja algoritma.

Sa takvim postupcima se postiže jednak učinak svih vrijednosti na analizu. Npr. u odabranom skupu podataka postoje stupci za dimenzije automobila (visina, širina i duljina) i za njegovu težinu, a ta 4 broja se međusobno mogu znatno razlikovati.

```
## Normalization and Standardization of Numeric Variables
```{r}
min_max_normalize <- function(x) {
  return ((x - min(x)) / (max(x) - min(x)))
}

z_score_standardize <- function(x) {
  return ((x - mean(x)) / sd(x))
}

numeric_columns <- c("wheelbase", "carlength", "carwidth", "carheight", "curbweight",
                     "enginesize", "boreratio", "stroke", "compressionratio", "horsepower",
                     "peakrpm", "citympg", "highwaympg")

head(my_cars[numeric_columns], 10)

my_cars[numeric_columns] <- lapply(my_cars[numeric_columns], min_max_normalize)

head(round(my_cars[numeric_columns], digits = 4), 10)

my_cars[numeric_columns] <- lapply(my_cars[numeric_columns], z_score_standardize)

head(round(my_cars[numeric_columns], digits = 4), 10)
```
```

Slika 26. Intervali pouzdanosti za srednju vrijednost cijene automobila sa različitim postotcima sigurnosti

U programskom kodu na slici 26. definirane su dvije funkcije, jedna za normalizaciju te jedna za standardizaciju vrijednosti proslijeđenog skupa podataka. U varijablu `numeric_columns` spremljen je vektor koji sadrži imena svih stupaca koje sadrže numeričke vrijednosti. Nakon toga se vrše redom normalizacija pa standardizacija svih numeričkih vrijednosti, popraćene sa ispisom originalnih, normaliziranih te normaliziranih i standardiziranih numeričkih vrijednosti.

	wheelbase <dbl>	carlength <dbl>	carwidth <dbl>	carheight <dbl>	curbweight <int>	enginesize <int>
1	88.6	168.8	64.1	48.8	2548	130
2	88.6	168.8	64.1	48.8	2548	130
3	94.5	171.2	65.5	52.4	2823	152
4	99.8	176.6	66.2	54.3	2337	109
5	99.4	176.6	66.4	54.3	2824	136
6	99.8	177.3	66.3	53.1	2507	136
7	105.8	192.7	71.4	55.7	2844	136
8	105.8	192.7	71.4	55.7	2954	136
9	105.8	192.7	71.4	55.9	3086	131
10	99.5	178.2	67.9	52.0	3053	131

1-10 of 10 rows | 1-7 of 13 columns

Slika 27. Numeričke vrijednosti prije normalizacije i standardizacije

Pregledom originalnih numeričkih vrijednosti na slici 27. vidi se da zbog korištenja raznih mjernih jedinica vrijednosti među stupcima imaju veliku razliku. Zbog toga je potrebna normalizacija i standardizacija numeričkih vrijednosti.

	wheelbase <dbl>	carlength <dbl>	carwidth <dbl>	carheight <dbl>	curbweight <dbl>	enginesize <dbl>
1	0.0583	0.4134	0.3167	0.0833	0.4112	0.2604
2	0.0583	0.4134	0.3167	0.0833	0.4112	0.2604
3	0.2303	0.4493	0.4333	0.3833	0.5178	0.3434
4	0.3848	0.5299	0.4917	0.5417	0.3293	0.1811
5	0.3732	0.5299	0.5083	0.5417	0.5182	0.2830
6	0.3848	0.5403	0.5000	0.4417	0.3953	0.2830
7	0.5598	0.7701	0.9250	0.6583	0.5260	0.2830
8	0.5598	0.7701	0.9250	0.6583	0.5687	0.2830
9	0.5598	0.7701	0.9250	0.6750	0.6199	0.2642
10	0.3761	0.5537	0.6333	0.3500	0.6071	0.2642

1-10 of 10 rows | 1-7 of 13 columns

Slika 28. Numeričke vrijednosti poslije normalizacije

Nakon normalizacije može se uočiti na slici 28. kako se sve vrijednosti nalaze u intervalu [0, 1]. Normalizirane numeričke vrijednosti u pravilu ne zahtijevaju još i standardizaciju jer ta dva postupka služe istoj svrsi na vrlo sličan način. Također standardizacija normaliziranih vrijednosti i originalnih vrijednosti uvijek daje isti rezultat tako da su normalizacija i standardizacija jedna za drugom nepotrebne osim ako su nužno potrebne normalizirane vrijednosti. Jedan od takvih scenarija se nalazi kod izgradnje neuronskih mreža koje imaju bolje performanse kada su im ulazni podaci normalizirani, a kasnije za lakšu interpretaciju modela su poželjne skalirane vrijednosti. Na slici 29. se nalazi i rezultat standardizacije.

	wheelbase <dbl>	carlength <dbl>	carwidth <dbl>	carheight <dbl>	curbweight <dbl>
1	-1.6866429	-0.4254799	-0.8427194	-2.0154834	-0.01453071
2	-1.6866429	-0.4254799	-0.8427194	-2.0154834	-0.01453071
3	-0.7068655	-0.2309477	-0.1901008	-0.5422002	0.51362457
4	0.1732736	0.2067498	0.1362086	0.2353660	-0.41976986
5	0.1068480	0.2067498	0.2294398	0.2353660	0.51554514

5 rows | 1-6 of 13 columns

Slika 29. Numeričke vrijednosti poslije normalizacije i skaliranja (standardizacije)

Nakon standardizacije podataka vidi se da više nisu u intervalu [0, 1] već da izlaze iz njega i u negativnu i u pozitivnu stranu.

```
## Mean and Standard Deviation of Numeric Variables
```{r}
my_cars[numeric_columns] %>% colMeans
sapply(my_cars[numeric_columns], sd)
```
```

Slika 30. Ispis srednjih vrijednost i standardnih devijacija numeričkih varijabli

Funkcija `colMeans` prima skup podataka te za svaki stupac (podskup) vraća njegovu srednju vrijednost. Za izračun standardne devijacije postoji funkcija `sd` koja vraća standardnu devijaciju jednog podskupa, a da bi se dobile standardne devijacije za svaki od podskupova potrebna je funkcija `sapply` koja primjenjuje proslijeđenu funkciju na svaki od podskupova i rezultate te funkcije vraća unutar vektora.

```
      wheelbase        carlength        carwidth       carheight
   1.071907e-15    -9.500530e-16    1.088899e-15    -5.182677e-16
     curbweight       enginesize       boreratio          stroke
   1.129009e-16     2.863563e-17    5.991650e-16     6.562332e-16
compressionratio       horsepower          peakrpm         citympg
   6.959203e-17     1.756386e-16    2.155457e-16     1.177920e-16
     highwaympg
   1.521818e-16
      wheelbase        carlength        carwidth       carheight
              1                1               1               1
     curbweight       enginesize       boreratio          stroke
              1                1               1               1
compressionratio       horsepower          peakrpm         citympg
              1                1               1               1
     highwaympg
              1
```

Slika 31. Srednje vrijednosti i standardne devijacije skaliranih vrijednosti

Pozivom funkcije `colMeans` se vidi da je srednja svih numeričkih stupaca približna nuli. Također slika 31. prikazuje da je standardna devijacija za svaki numerički podskup nad kojim

je provedena standardizacija jednaka 1 čime je dokazano da je standardizacija uspješno provedena.

5.1.1. Dplyr paket

Isti postupak može se provesti i koristeći `dplyr` paket, odnosno njegove funkcije `mutate`, `across` i `all_of` s kojim se može reći da se neka funkcija izvede nad svakom vrijednosti unutar nekog stupca odnosno vektora. Programski kôd koji provodi takav postupak je prikazan na slici 32.

```
## Standardization and Normalization with dplyr
```{r}
my_cars <- my_cars %>%
      mutate(across(all_of(numeric_columns), min_max_normalize)) %>%
      mutate(across(all_of(numeric_columns), z_score_standardize))

head(my_cars[numeric_columns], 5)
```
```

Slika 32. Srednje vrijednosti i standardne devijacije skaliranih vrijednosti

5.2. Inženjering značajki (*engl. Feature engineering*)

Feature engineering obuhvaća skup operacija koje se izvode na značajkama u nekom skupu podataka kako bi se unaprijedile performanse modela strojnog učenja. U skupu podataka ovog završnog rada postoji stupac *CarName* čije vrijednosti sadrže marku i model automobila. Takve vrijednosti se mogu razdvojiti u dva stupca tako da u jednom bude samo marka, a u drugom samo model automobila.

5.2.1. Razdvajanje stupca *CarName*

Takav postupak stvaranja zasebnog stupca za marku i za model utječe na kvalitetu analize skupa podataka. Sa podatkom o marki mogu se uočiti brojni trendovi i pronaći odgovori na pitanja, npr. koja marka je najčešća u skupu podataka, koje marke automobila imaju najveću cijenu, čiji automobili imaju najveću konjsku snagu i brojna druga pitanja.

U programskom jeziku R takav postupak može se ostvariti funkcijom `separate` iz paketa `tidyr` koja prima podatkovni okvir, odnosno stupac koji se razdvaja, imena varijabli u koje će se spremiti rezultat razdvajanja, separator, odnosno niz znakova koji predstavlja ono po čemu će se vrijednosti razdvajati te pravilo koje će se primijeniti na vrijednosti koje nakon razdvajanja budu imale više dijelova u stupaca u koje se trebaju pospremiti.

Na slici 33. je prikazan kôd za razdvajanje stupca. Kao *separator* je proslijeđen razmak, a u slučaju da vrijednost u stupcu *CarName* ima više riječi, prva riječ će predstavljati marku automobila, odnosno *Brand*, a sve ostalo model automobila. Tako će na primjer vrijednost „Chevrolet monte carlo“ završiti tako da će u *Brand* ići „Chevrolet“, a u *Model* „monte carlo“. Nakon razdvajanja s obzirom da je marka automobila izdvojena u zaseban stupac se može odraditi ispravak konzistencije naziva nekih brendova. Na primjer, za automobile marke Volkswagen su postojale 3 različite vrijednosti te je samo bilo potrebno odlučiti se za jednu i sve ostale preimenovati u tu jednu.

```
## Brand Values Consistency Fix
```{r}
my_cars <- my_cars %>% separate(CarName, into = c("brand", "model"), sep = " ", extra = "merge")

my_cars[my_cars == "volkswagen" | my_cars == "vokswagen"] <- "vw"
my_cars[my_cars == "toyouta"] <- "toyota"
my_cars[my_cars == "porcshce"] <- "porsche"
my_cars[my_cars == "Nissan"] <- "nissan"
my_cars[my_cars == "maxda"] <- "mazda"

my_cars[sample(nrow(my_cars), 7), ]
```
```



Description: df [7 × 28]

	car_ID <int>	symboling <int>	brand <chr>	model <chr>	fueltype <chr>	aspiration <chr>	doornumber <chr>	carbody <chr>	drivewheel <chr>
149	149	0	subaru	dl	gas	std	four	wagon	4wd
107	107	1	nissan	clipper	gas	std	two	hatchback	rwd
80	80	1	mitsubishi	g4	gas	turbo	two	hatchback	fwd
88	88	1	mitsubishi	outlander	gas	turbo	four	sedan	fwd
47	47	2	isuzu	D-Max	gas	std	two	hatchback	rwd
79	79	2	mitsubishi	outlander	gas	std	two	hatchback	fwd
45	45	1	isuzu	D-Max	gas	std	two	sedan	fwd

7 rows | 1-10 of 28 columns

Slika 33. Razdvajanje stupca *CarName* i ispravak naziva brendova

6. Kodiranje kategoričkih varijabli

Kategoričke varijable su one varijable koje predstavljaju kategorije ili grupe. One nemaju svojstva numeričkih varijabli kao što su zbrajanje, oduzimanje, računanje srednje vrijednosti i slično, već predstavljaju atribute ili karakteristike nekog entiteta. Na primjer, u skupu podataka sa cijenama automobila kategoričke varijable su: *fueltype, aspiration, carbody, drivewheel, enginelocation, enginetype* i *fuelsystem*. To su sve varijable koje mogu automobile svrstati u grupe ispisane naredbom `unique` na slici 34.

Slika 34. Vrijednosti vektora kategoričkih varijabli skupa podataka

Ovakvi podaci predstavljaju problem algoritmu za strojno učenje s obzirom na to da algoritam traži numeričke vrijednosti, a ne tekstualne. Za taj problem rješenje je kodiranje takvih varijabli koje pretvara tekstualne vrijednost u numeričke. Postoji nekoliko metoda za kodiranje ovisno o podacima koji se nalaze u takvom stupcu.

6.1. One-Hot kodiranje (*engl. One-hot Encoding*)

Način na koji radi *one-hot* kodiranje varijabli je taj da za svaku jedinstvenu vrijednost u stupcu kreira novi binarni stupac (imati će vrijednost ili 0 ili 1) koji će biti indikator o pripadnosti retka određenoj kategoriji. Na primjer za stupac *fueltype* mogu se kreirati 2 nova stupca koja bi se zvala *fueltypediesel* i *fueltypegas*.

Za one-hot kodiranje postoji paket *caret* sa kojim se u nekoliko koraka može postići kodiranje varijabli. Ovako izgleda primjer koda u kojem se kodira varijabla *fueltype*:

U prvoj liniji koda na slici 35. nakon pozivanja paketa, poziva se funkcija *dummyVars* koja kreira model za nove varijable koje će nastati kao rezultat kodiranja, a nakon nje se poziva *predict* koja kada se koristi zajedno sa *dummyVars* objektom služi da primijeni transformacije na dani skup podataka. Funkcija *predict* vraća matricu tako da je još i potrebna prebacivanje u

data.frame tip varijable. Na kraju da bi se stupci spojili sa originalnim skupom podataka moguće je sa funkcijom *cbind* spojiti novonastale stupce te sve stupce iz originalnog skupa podataka osim onog koji se kodirao zato što taj nije potreban algoritmu za strojno učenje.

```r
library(caret)

dummy_formula <- dummyVars(~ fueltype, data = my_cars)

carprice_encoded <- predict(dummy_formula, newdata = my_cars)

carprice_encoded <- as.data.frame(carprice_encoded)

my_cars_new <- cbind(carprice_encoded, my_cars[ , !names(my_cars) %in% c("fueltype")])

sample_n(my_cars_new[c("fueltypediesel", "fueltypegas", "brand", "model")], 10)
```

Description: df [10 × 4]

	fueltypediesel <dbl>	fueltypegas <dbl>	brand <chr>	model <chr>
144	0	1	subaru	baja
59	0	1	mazda	glc 4
22	0	1	dodge	rampage
132	0	1	renault	5 gtl
125	0	1	plymouth	duster
91	1	0	nissan	gt-r
108	0	1	peugeot	504
109	1	0	peugeot	304
80	0	1	mitsubishi	g4
172	0	1	toyota	corolla

1-10 of 10 rows

Slika 35. Deset nasumičnih redaka nakon *one-hot* kodiranja varijable *fueltype*

6.2. Ordinalno kodiranje (*engl. Ordinal Encoding*)

Ordinalno kodiranje je postupak dodjeljivanja numeričke vrijednosti svakoj kategoriji unutar kategoričke varijable. Na primjer u svijetu softverskog inženjerstva programeri se prema svom iskustvu svrstavaju u 3 glavne kategorije: *junior, mid* i *senior* programeri. Takve kategorije su rangirane gdje je junior programer sa najmanje iskustva, a senior sa najviše. Prema tome kada bi se svakom stupnju iskustva dodijelila numerička vrijednost, *junior* kategoriji bi se dodijelio broj 1, *mid* kategoriji broj 2 i *senior* kategoriji broj 3 čime se zadržao i redoslijed tako što najslabija kategorija ima najmanji broj, a najjača ima najveći broj.

U skupu podataka sa cijenama automobila postoje dva takva stupca koja su kategoričke varijable i čije varijable su rangirane. To su *cylindernumber* i *doornumber*. Vrijednosti u tim stupcima su tekstualnog karaktera, ali zapravo predstavljaju konkretan broj što se vidi na slici 36.

```r
{r}
head(my_cars$cylindernumber, 10)
head(my_cars$doornumber, 10)
```

```
 [1] "four" "four" "six"  "four" "five" "five" "five" "five" "five" "five"
 [1] "two"  "two"  "two"  "four" "four" "two"  "four" "four" "four" "two"
```

Slika 36. Vrijednosti stupaca cylindernumber i doornumber prije kodiranja

Da bi se takvi znakovni nizovi pretvorili u brojeve, može se kreirati vektor u kojem se definira koja riječ označava koji broj, te onda koristeći funkciju *sapply* koja prima listu ili vektor te vraća listu ili vektor iste duljine nakon što provuče svaki njen element kroz određenu funkciju, pretvoriti sve riječi u brojeve. Na slici 37. to je i implementirano. Prvo se definira vektor *number_mapping* koji sadrži informaciju o tome koja riječ označava koji broj te se onda na vrijednosti vektora *doornumber* i *cylindernumber* primjenjuje funkcija koja vraća pripadajuću vrijednost iz vektora *number_mapping* za trenutnu vrijednost u stupcu.

```r
```{r}
number_mapping <- c("two" = 2, "three" = 3, "four" = 4, "five" = 5, "six" = 6, "seven" = 7,
"eight" = 8, "nine" = 9, "ten" = 10, "eleven" = 11, "twelve" = 12)

head(my_cars$cylindernumber, 10)
head(my_cars$doornumber, 10)

my_cars$doornumber <- sapply(my_cars$doornumber, function(x) number_mapping[x])
my_cars$cylindernumber <- sapply(my_cars$cylindernumber, function(x) number_mapping[x])

head(my_cars$cylindernumber, 10)
head(my_cars$doornumber, 10)
```
```

```
 [1] "four" "four" "six"  "four" "five" "five" "five" "five" "five" "five"
 [1] "two"  "two"  "two"  "four" "four" "two"  "four" "four" "four" "two"
 [1] 4 4 6 4 5 5 5 5 5 5
 [1] 2 2 2 4 4 2 4 4 4 2
```

Slika 37. Vrijednosti stupaca cylindernumber i doornumber nakon kodiranja

S obzirom na to da se već proveo postupak normalizacije i standardizacije numeričkih podataka, može se isto provesti i na novonastalim numeričkim podacima iz kodiranja kao na slici 38.

```r
number_mapping <- c("two" = 2, "three" = 3, "four" = 4, "five" = 5, "six" = 6, "seven" = 7, "eight" = 8, "nine" = 9,
"ten" = 10, "eleven" = 11, "twelve" = 12)

head(my_cars$cylindernumber, 10)
head(my_cars$doornumber, 10)

my_cars$doornumber <- sapply(my_cars$doornumber, function(x) number_mapping[x])
my_cars$cylindernumber <- sapply(my_cars$cylindernumber, function(x) number_mapping[x])

head(my_cars$cylindernumber, 10)
head(my_cars$doornumber, 10)

encoded_columns <- c("cylindernumber", "doornumber")

my_cars <- my_cars %>%
      mutate(across(all_of(encoded_columns), min_max_normalize)) %>%
      mutate(across(all_of(encoded_columns), z_score_standardize))

head(my_cars$cylindernumber, 10)
head(my_cars$doornumber, 10)
```

```
 [1] "four" "four" "six"  "four" "five" "five" "five" "five" "five" "five"
 [1] "two"  "two"  "two"  "four" "four" "two"  "four" "four" "four" "two"
 [1] 4 4 6 4 5 5 5 5 5 5
 [1] 2 2 2 4 4 2 4 4 4 2
 [1] -0.3520252 -0.3520252  1.4983638 -0.3520252  0.5731693  0.5731693  0.5731693  0.5731693  0.5731693  0.5731693
 [1] -1.1276279 -1.1276279 -1.1276279  0.8824914  0.8824914 -1.1276279  0.8824914  0.8824914  0.8824914 -1.1276279
```

Slika 38. Vrijednosti stupaca cylindernumber i doornumber nakon kodiranja, normalizacije i standardizacije

6.3. Ostale metode kodiranja

6.3.1. Binarno kodiranje (*engl. Binary Encoding*)

Binarno kodiranje je metoda kodiranja slična *one-hot* kodiranju i sastoji se od 3 koraka. Prvi korak je primijeniti ordinalno kodiranje na određenu kategoričku varijablu i svakoj kategoriji dodijeliti numeričku vrijednost. Drugi korak je pretvaranje novonastale numeričke vrijednosti u binarni zapis uzevši u obzir i koliko kategorija postoji, tako na primjer broj 3 će postati 11 ili 011 ako ima više od 3 kategorije. Zadnji korak je po principu *one-hot* kodiranja, binarni zapis rastaviti u onoliko stupaca koliko ima znamenki.

6.3.2. Ciljno kodiranje (*engl. Target Encoding*)

Ciljno kodiranje ili *Target Encoding* je tehnika kodiranja kod koje se umjesto kategorije postavlja srednja vrijednost ciljne varijable. Na primjeru skupa podataka sa cijenama automobila, varijabla *brand* se mogla kodirati tako da se na mjesto svake marke automobila stavi srednja vrijednost cijene automobila te marke.

6.3.3. Frekvencijsko kodiranje (*engl. Frequency Encoding*)

Frekvencijsko kodiranje je metoda kod koje se umjesto kategorije postavlja njen broj pojavljivanja u kategoričkoj varijabli. Ova metoda može biti korisna za hvatanje informacije o važnosti ili popularnosti određene kategorije. Frekvencijsko kodiranje je poželjno kad kategorička varijabla ima visoku kardinalnost, odnosno kada je broj jedinstvenih vrijednosti u stupcu velik i kada je informacija o učestalosti kategorije od značajnije važnosti.

7. Zaključak

U ovom radu provedena je priprema i istraživačka analiza skupa podataka o cijenama automobila s ciljem identifikacije ključnih faktora koji utječu na cijenu automobila. Korištene su metode deskriptivne i inferencijalne statistike za ispitivanje međusobnih odnosa između različitih karakteristika automobila i njihove cijene.

Analiza je pokazala da varijable kao što su: tip karoserije, veličina motora, snaga motora i potrošnja goriva imaju značajan utjecaj na cijenu automobila. ANOVA test je otkrio značajne razlike u srednjim cijenama između različitih tipova karoserije, dok je Chi-kvadrat test pokazao povezanost između tipa karoserije i vrste goriva.

Najvažniji zaključak rada je da različite tehničke i funkcionalne karakteristike automobila značajno doprinose njihovoj konačnoj cijeni. Ovakvi rezultati mogu pružiti korisne uvide proizvođačima automobila za optimizaciju strategija određivanja cijena, kao i potencijalnim kupcima za donošenje informiranih odluka prilikom kupovine automobila.

Preporuke za daljnji rad uključuju proširenje analize na veći skup podataka i uključivanje dodatnih varijabli kao što su sigurnosne karakteristike i ocjene zadovoljstva korisnika.

Literatura

(18. 3 2024). R Documentation: https://www.rdocumentation.org

(28. 5 2024). Clustering in Machine Learning: https://www.geeksforgeeks.org/clustering-in-machine-learning/

(28. 5 2024). Scaling and Normalization: https://www.kaggle.com/code/alexisbcook/scaling-and-normalization

(28. 5 2024). Feature Scaling: https://en.wikipedia.org/wiki/Feature_scaling

(28. 5 2024). z-score Standardization in R: https://www.geeksforgeeks.org/z-score-standardization-in-r/

(22. 6 2024). Categorical Data Encoding Techniques: https://medium.com/aiskunks/categorical-data-encoding-techniques-d6296697a40f

(23. 6 2024). Encoding Categorical Data in R: https://www.geeksforgeeks.org/encoding-categorical-data-in-r/

(23. 6 2024). cut: Convert Numeric to Factor: https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/cut

(25. 6 2024). What Is Analysis of Variance (ANOVA)?: https://www.investopedia.com/terms/a/anova.asp

Popis tablica

Popis slika

Popis priloga

Uz završni rad priložen je skup podataka u CSV formatu i RMarkdown dokument sa programskim kodom i interpretacijom rezultata.