****

**ИНСТИТУТ ЗА МАТЕМАТИКУ И ИНФОРМАТИКУ**

**ПРИРОДНО-МАТЕМАТИЧКИ ФАКУЛТЕТ**

**УНИВЕРЗИТЕТА У КРАГУЈЕВЦУ**

**СЕМИНАРСКИ РАД**

**Предмет: Увод у науку о подацима**

**Тема: Половни аутомобили**

**Студенти: Професор:**

**Маријана Петровић 109/2018 др БранкоАрсић**

**Александар Милутиновић 128/2017**

Крагујевац, 2021. год.

Садржај

[База података 4](#_Toc81121283)

[Недостајуће вредности 8](#_Toc81121284)

[Обрада недостајућих вредности 8](#_Toc81121286)

[Визуелизација и анализа података 11](#_Toc81121287)

[Моделовање 26](#_Toc81121288)

[Линеарна регресија 28](#_Toc81121289)

[Стабло одлучивања машинског учења 36](#_Toc81121290)

[Метрике 40](#_Toc81121291)

[„Support Vector Machine“ алгоритам машинског учења 41](#_Toc81121292)

[Метрике 43](#_Toc81121293)

[Закључак 45](#_Toc81121294)

[Литература 45](#_Toc81121295)

# База података

Скуп података под називом „ Used cars catalog“, који ћемо представити у семинарском раду, представља колекцију података о половним аутомобилима, преузетих са више интернет сајтова у Белорусији, у децембру 2019. године.

База података се састоји од једног .csv фајла

За почетак, учитаћемо библиотеке неопходне за даљи рад.

library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.0.5

## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3 v purrr 0.3.4  
## v tibble 3.1.0 v dplyr 1.0.5  
## v tidyr 1.1.3 v stringr 1.4.0  
## v readr 1.4.0 v forcats 0.5.1

## Warning: package 'ggplot2' was built under R version 4.0.5

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(ISLR)

## Warning: package 'ISLR' was built under R version 4.0.5

library(visdat)

## Warning: package 'visdat' was built under R version 4.0.5

library(car)

## Loading required package: carData

##   
## Attaching package: 'car'

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

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

library(leaps)

## Warning: package 'leaps' was built under R version 4.0.5

Учитавање скупа података из .csv фајла.

carsDataSet = read.csv("seminarski/cars.csv", stringsAsFactors = F)

Пре свега, издвојићемо само колоне које су нам потребне, док ћемо колоне које нам представљају проблем, због превише нејасноћа, отклонити из скупа.

carSubset = subset(carsDataSet,select= c("manufacturer\_name","transmission","color","odometer\_value","year\_produced","engine\_fuel","engine\_has\_gas","engine\_type",  
 "engine\_capacity","body\_type","has\_warranty","state","drivetrain","price\_usd","is\_exchangeable","number\_of\_photos","duration\_listed"))

Погледаћемо детаљније скуп података који ћемо обрађивати. Описаћемо само променљиве које ћемо користити у даљој анализи скупа carSubset.

* manufacturer\_name: Произвођач аутомобила (Категоријска променљива)
* transmission: Тип мењача аутомобила (Категоријска променљива)
* color: Боја аутомобила (Категоријска променљива)
* odometer\_value: Пређена километража (Нумеричка променљива)
* year\_produced: Година производње (Нумеричка променљива)
* engine\_fuel: Тип горива мотора (Категоријска променљива)
* engine\_has\_gas: Да ли аутомобил има уграђен гас (Логичка променљива)
* engine\_type: Тип мотора (Категоријска променљива)
* engine\_capacity: Запремина мотора (Нумеричка променљива)
* body\_type: Тип каросерије (Категоријска променљива)
* has\_warranty: Да ли возило има гаранцију (Логичка променљива)
* state: Стање возила. Описује да ли је возило ново, половно или оштећено (Категоријска променљива)
* drivetrain: Тип погона возила - Предњи, задњи или сва четири точка. (Категоријска променљива)
* is\_exchangeable: Да ли је могућа замена (Логичка променљива)
* number\_of\_photos: Број фотографија возила на огласу. (Нумеричка променљива)
* \_listed: Број дана колико је оглас приказан на сајту (Нумеричка променљива)

Циљна променљива: price\_usd: Цена возила у америчким доларима. (Нумеричка променљива)

str(carSubset)

## 'data.frame': 38531 obs. of 17 variables:  
## $ manufacturer\_name: chr "Subaru" "Subaru" "Subaru" "Subaru" ...  
## $ transmission : chr "automatic" "automatic" "automatic" "mechanical" ...  
## $ color : chr "silver" "blue" "red" "blue" ...  
## $ odometer\_value : int 190000 290000 402000 10000 280000 132449 318280 350000 179000 571317 ...  
## $ year\_produced : int 2010 2002 2001 1999 2001 2011 1998 2004 2010 1999 ...  
## $ engine\_fuel : chr "gasoline" "gasoline" "gasoline" "gasoline" ...  
## $ engine\_has\_gas : logi FALSE FALSE FALSE FALSE FALSE FALSE ...  
## $ engine\_type : chr "gasoline" "gasoline" "gasoline" "gasoline" ...  
## $ engine\_capacity : num 2.5 3 2.5 3 2.5 2.5 2.5 2.5 2.5 2.5 ...  
## $ body\_type : chr "universal" "universal" "suv" "sedan" ...  
## $ has\_warranty : logi FALSE FALSE FALSE FALSE FALSE FALSE ...  
## $ state : chr "owned" "owned" "owned" "owned" ...  
## $ drivetrain : chr "all" "all" "all" "all" ...  
## $ price\_usd : num 10900 5000 2800 9999 2134 ...  
## $ is\_exchangeable : logi FALSE TRUE TRUE TRUE TRUE TRUE ...  
## $ number\_of\_photos : int 9 12 4 9 14 20 8 7 17 8 ...  
## $ duration\_listed : int 16 83 151 86 7 67 307 73 87 43 ...

На основу приказа структуре скупа података, можемо закључити да имамо три различита типа података, нумеричке, логичке и карактер променљиве.

summary(carSubset)

## manufacturer\_name transmission color odometer\_value   
## Length:38531 Length:38531 Length:38531 Min. : 0   
## Class :character Class :character Class :character 1st Qu.: 158000   
## Mode :character Mode :character Mode :character Median : 250000   
## Mean : 248865   
## 3rd Qu.: 325000   
## Max. :1000000   
##   
## year\_produced engine\_fuel engine\_has\_gas engine\_type   
## Min. :1942 Length:38531 Mode :logical Length:38531   
## 1st Qu.:1998 Class :character FALSE:37184 Class :character   
## Median :2003 Mode :character TRUE :1347 Mode :character   
## Mean :2003   
## 3rd Qu.:2009   
## Max. :2019   
##   
## engine\_capacity body\_type has\_warranty state   
## Min. :0.200 Length:38531 Mode :logical Length:38531   
## 1st Qu.:1.600 Class :character FALSE:38082 Class :character   
## Median :2.000 Mode :character TRUE :449 Mode :character   
## Mean :2.055   
## 3rd Qu.:2.300   
## Max. :8.000   
## NA's :10   
## drivetrain price\_usd is\_exchangeable number\_of\_photos  
## Length:38531 Min. : 1 Mode :logical Min. : 1.000   
## Class :character 1st Qu.: 2100 FALSE:24945 1st Qu.: 5.000   
## Mode :character Median : 4800 TRUE :13586 Median : 8.000   
## Mean : 6640 Mean : 9.649   
## 3rd Qu.: 8990 3rd Qu.:12.000   
## Max. :50000 Max. :86.000   
##   
## duration\_listed   
## Min. : 0.00   
## 1st Qu.: 23.00   
## Median : 59.00   
## Mean : 80.58   
## 3rd Qu.: 91.00   
## Max. :2232.00   
##

Извршавањем функције summary добијамо детаљније информације о нашем скупу, као што су минимум и максмум, медијана, средња вредност као и да ли скуп садржи недостајуће вредности.

У нашем случају NA вредности налазе се само у колони о кубикажи мотора. Детаљнијом провером колона, закључили смо да код стринг вредности немамо недостајуће вредности попут празног стринга, цртице или томе слично.

carSubset$model\_name[carSubset$model\_name == ""]

## NULL

carSubset$model\_name[carSubset$model\_name == "-"]

## NULL

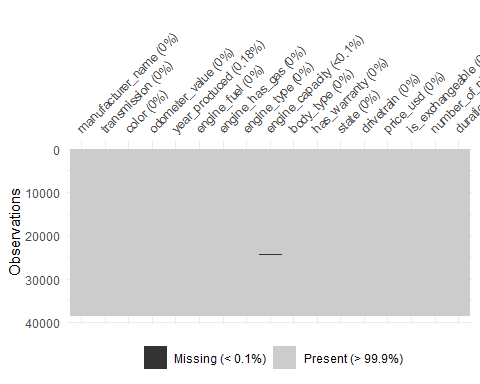
Поступак је поновљен за све колоне карактер типа.

# Недостајуће вредности

# carSubset$year\_produced[carSubset$year\_produced < 1970 ] <- NA

Због мањег интересовања купаца за возила старија од 50 година, одлучили смо да уместо тих возила поставимо недостајуће вредности. Сва возила произведена пре 1970. године добиће NА вредност.

vis\_miss(carSubset)



На графичком приказу недостајућих вредности скупа података, можемо видети да је тај број веома мали чак мањи од 0.1%.

## Обрада недостајућих вредности

summary(carSubset)

## manufacturer\_name transmission color odometer\_value   
## Length:38531 Length:38531 Length:38531 Min. : 0   
## Class :character Class :character Class :character 1st Qu.: 158000   
## Mode :character Mode :character Mode :character Median : 250000   
## Mean : 248865   
## 3rd Qu.: 325000   
## Max. :1000000   
##   
## year\_produced engine\_fuel engine\_has\_gas engine\_type   
## Min. :1970 Length:38531 Mode :logical Length:38531   
## 1st Qu.:1998 Class :character FALSE:37184 Class :character   
## Median :2003 Mode :character TRUE :1347 Mode :character   
## Mean :2003   
## 3rd Qu.:2009   
## Max. :2019   
## NA's :71   
## engine\_capacity body\_type has\_warranty state   
## Min. :0.200 Length:38531 Mode :logical Length:38531   
## 1st Qu.:1.600 Class :character FALSE:38082 Class :character   
## Median :2.000 Mode :character TRUE :449 Mode :character   
## Mean :2.055   
## 3rd Qu.:2.300   
## Max. :8.000   
## NA's :10   
## drivetrain price\_usd is\_exchangeable number\_of\_photos  
## Length:38531 Min. : 1 Mode :logical Min. : 1.000   
## Class :character 1st Qu.: 2100 FALSE:24945 1st Qu.: 5.000   
## Mode :character Median : 4800 TRUE :13586 Median : 8.000   
## Mean : 6640 Mean : 9.649   
## 3rd Qu.: 8990 3rd Qu.:12.000   
## Max. :50000 Max. :86.000   
##   
## duration\_listed   
## Min. : 0.00   
## 1st Qu.: 23.00   
## Median : 59.00   
## Mean : 80.58   
## 3rd Qu.: 91.00   
## Max. :2232.00   
##

С обзиром да имамо NA вредности у колони engine\_capacitу, а како је она континуална променљива те вредности попунићемо медијаном те колоне.

med = median(carSubset$engine\_capacity, na.rm=TRUE)  
print(med)

## [1] 2

carSubset %>% group\_by(engine\_capacity) %>% count(engine\_capacity, sort= TRUE)

## # A tibble: 62 x 2  
## # Groups: engine\_capacity [62]  
## engine\_capacity n  
## <dbl> <int>  
## 1 2 7670  
## 2 1.6 6188  
## 3 1.8 4244  
## 4 2.5 2520  
## 5 1.4 2394  
## 6 1.9 2304  
## 7 2.2 1852  
## 8 2.4 1596  
## 9 3 1520  
## 10 1.5 1461  
## # ... with 52 more rows

Из приложеног видимо да је медијана 2.0, али и да је та вредност најзаступљенија у колони и том вредношћу попуњавамо NA вредности.

carSubset$engine\_capacity[is.na(carSubset$engine\_capacity)] <- med

summary(carSubset)

## manufacturer\_name transmission color odometer\_value   
## Length:38531 Length:38531 Length:38531 Min. : 0   
## Class :character Class :character Class :character 1st Qu.: 158000   
## Mode :character Mode :character Mode :character Median : 250000   
## Mean : 248865   
## 3rd Qu.: 325000   
## Max. :1000000   
##   
## year\_produced engine\_fuel engine\_has\_gas engine\_type   
## Min. :1970 Length:38531 Mode :logical Length:38531   
## 1st Qu.:1998 Class :character FALSE:37184 Class :character   
## Median :2003 Mode :character TRUE :1347 Mode :character   
## Mean :2003   
## 3rd Qu.:2009   
## Max. :2019   
## NA's :71   
## engine\_capacity body\_type has\_warranty state   
## Min. :0.200 Length:38531 Mode :logical Length:38531   
## 1st Qu.:1.600 Class :character FALSE:38082 Class :character   
## Median :2.000 Mode :character TRUE :449 Mode :character   
## Mean :2.055   
## 3rd Qu.:2.300   
## Max. :8.000   
##   
## drivetrain price\_usd is\_exchangeable number\_of\_photos  
## Length:38531 Min. : 1 Mode :logical Min. : 1.000   
## Class :character 1st Qu.: 2100 FALSE:24945 1st Qu.: 5.000   
## Mode :character Median : 4800 TRUE :13586 Median : 8.000   
## Mean : 6640 Mean : 9.649   
## 3rd Qu.: 8990 3rd Qu.:12.000   
## Max. :50000 Max. :86.000   
##   
## duration\_listed   
## Min. : 0.00   
## 1st Qu.: 23.00   
## Median : 59.00   
## Mean : 80.58   
## 3rd Qu.: 91.00   
## Max. :2232.00   
##

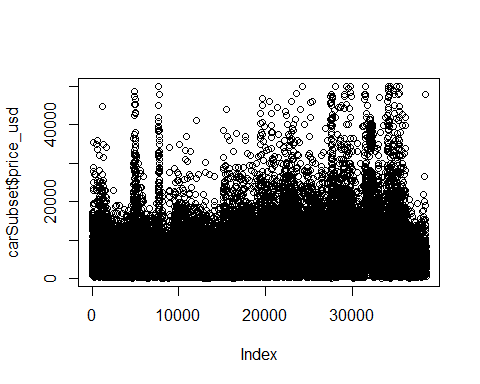
Дакле, видимо да у колони engine\_capacity више немамо недостајућих вредности.

Како нам не би представљале проблем у даљем раду, обрисаћемо врсте са недостајућим вредностима у колони year\_produced.

carSubset <- na.omit(carSubset)

# Визуелизација и анализа података

plot(carSubset$price\_usd)



Са графика видимо да цене у великој мери варирају, зато ћемо додати нову категоријску променљиву чије су категорије формиране на основу висине цене. Категорије су формиране на основу података прикупљених са сајта “Половни аутомобили”.

extraLowPrice <- carSubset$price\_usd[carSubset$price\_usd <= 1000]  
lowPrice <- carSubset$price\_usd[carSubset$price\_usd > 1000 & carSubset$price\_usd <= 4000 ]  
medPrice <- carSubset$price\_usd[carSubset$price\_usd > 4000 & carSubset$price\_usd <= 10000]  
highPrice <- carSubset$price\_usd[carSubset$price\_usd > 10000 & carSubset$price\_usd <= 30000]  
extraPrice <- carSubset$price\_usd[carSubset$price\_usd > 30000]  
  
carSubset$price\_cat <- vector(mode = "character", length = nrow(carSubset))  
carSubset$price\_cat[carSubset$price\_usd %in% extraLowPrice ] <- "extraLowPrice"  
carSubset$price\_cat[carSubset$price\_usd %in% lowPrice ] <- "lowPrice"  
carSubset$price\_cat[carSubset$price\_usd %in% medPrice ] <- "medPrice"  
carSubset$price\_cat[carSubset$price\_usd %in% highPrice ] <- "highPrice"  
carSubset$price\_cat[carSubset$price\_usd %in% extraPrice ] <- "extraPrice"

На исти начин ћемо приступити променљивој која представља годину производње. Категорије су одређене према стандардима Европске Уније.

veryOldCar <- carSubset$year\_produced[carSubset$year\_produced <= 1980]  
oldCar <- carSubset$year\_produced[carSubset$year\_produced > 1980 & carSubset$year\_produced <= 2000 ]  
medAgeCar <- carSubset$year\_produced[carSubset$year\_produced > 2000 & carSubset$year\_produced <= 2012]  
newCar <- carSubset$year\_produced[carSubset$year\_produced > 2012]  
  
  
carSubset$year\_produced\_cat <- vector(mode = "character", length = nrow(carSubset))  
carSubset$year\_produced\_cat[carSubset$year\_produced %in% veryOldCar ] <- "veryOldCar"  
carSubset$year\_produced\_cat[carSubset$year\_produced %in% oldCar ] <- "oldCar"  
carSubset$year\_produced\_cat[carSubset$year\_produced %in% medAgeCar ] <- "medAgeCar"  
carSubset$year\_produced\_cat[carSubset$year\_produced %in% newCar ] <- "newCar"

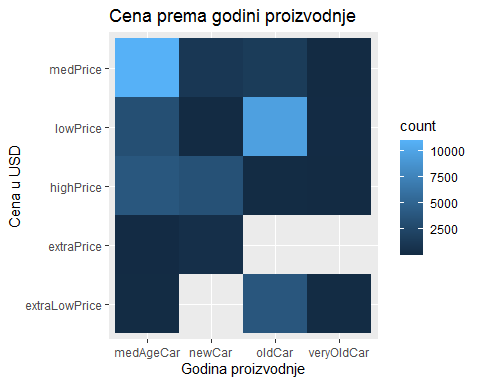
Ради лакше и тачније обраде податаке стринг варијабле ћемо факторизовати.

carSubset = as.data.frame(unclass(carSubset),stringsAsFactors=TRUE)

str(carSubset)

## 'data.frame': 38460 obs. of 19 variables:  
## $ manufacturer\_name: Factor w/ 55 levels "Acura","Alfa Romeo",..: 51 51 51 51 51 51 51 51 51 51 ...  
## $ transmission : Factor w/ 2 levels "automatic","mechanical": 1 1 1 2 1 1 1 1 1 1 ...  
## $ color : Factor w/ 12 levels "black","blue",..: 9 2 8 2 1 9 1 9 5 9 ...  
## $ odometer\_value : int 190000 290000 402000 10000 280000 132449 318280 350000 179000 571317 ...  
## $ year\_produced : int 2010 2002 2001 1999 2001 2011 1998 2004 2010 1999 ...  
## $ engine\_fuel : Factor w/ 6 levels "diesel","electric",..: 4 4 4 4 4 4 4 4 4 4 ...  
## $ engine\_has\_gas : logi FALSE FALSE FALSE FALSE FALSE FALSE ...  
## $ engine\_type : Factor w/ 3 levels "diesel","electric",..: 3 3 3 3 3 3 3 3 3 3 ...  
## $ engine\_capacity : num 2.5 3 2.5 3 2.5 2.5 2.5 2.5 2.5 2.5 ...  
## $ body\_type : Factor w/ 12 levels "cabriolet","coupe",..: 11 11 10 9 11 11 11 9 11 11 ...  
## $ has\_warranty : logi FALSE FALSE FALSE FALSE FALSE FALSE ...  
## $ state : Factor w/ 3 levels "emergency","new",..: 3 3 3 3 3 3 3 3 3 3 ...  
## $ drivetrain : Factor w/ 3 levels "all","front",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ price\_usd : num 10900 5000 2800 9999 2134 ...  
## $ is\_exchangeable : logi FALSE TRUE TRUE TRUE TRUE TRUE ...  
## $ number\_of\_photos : int 9 12 4 9 14 20 8 7 17 8 ...  
## $ duration\_listed : int 16 83 151 86 7 67 307 73 87 43 ...  
## $ price\_cat : Factor w/ 5 levels "extraLowPrice",..: 3 5 4 5 4 3 4 5 3 5 ...  
## $ year\_produced\_cat: Factor w/ 4 levels "medAgeCar","newCar",..: 1 1 1 3 1 1 3 1 1 3 ...

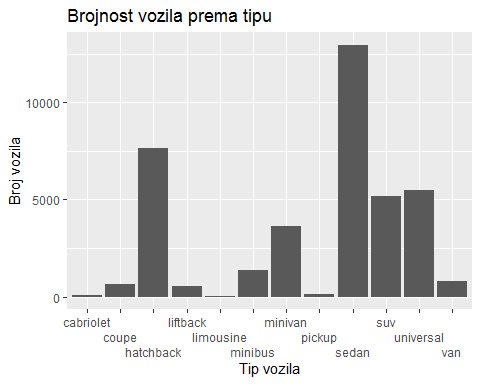
ggplot(data =carSubset) + geom\_bin2d( mapping = aes(x = year\_produced\_cat, y = price\_cat)) + labs(title ="Cena prema godini proizvodnje", x="Godina proizvodnje", y="Cena u USD")



Из приложеног графика закључујемо да цена у великој мери зависи од године производње, што доказује чињеница да у категорији највиших цена не постоје стара и веома стара возила, као и то да у категорији најнижих цена не постоје нова возила.

У свим категоријама су најмање заступљена стара возила, а разлог је то што их има најмање од свих. Такође видимо зависност категорија, где је средња старост возила најзаступљенија у рангу средњих цена. Та зависност је изражена повезаношћу ниских и екстра ниских цена са старим возилима, као и високих и екстра високих са новим возилима. Тиме долазимо до закључка да је ова повезаност очекивана.

ggplot(data= carSubset) + geom\_bar(mapping=aes(x=body\_type))+scale\_x\_discrete(guide = guide\_axis(n.dodge=3))+ labs(title ="Brojnost vozila prema tipu", x="Tip vozila", y="Broj vozila")



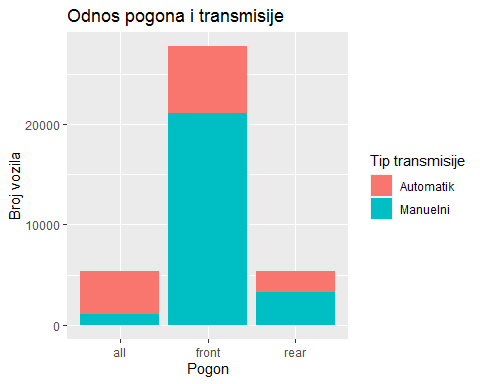
Можемо закључити да су највише заступљена возила седан и хечбек, док су најмање заступљени лимузина, кабриолет и пикап.

Пошто се на фрафику не види категорија лимузине, додаћемо табеларни приказ који ће приказати бројност свих категорија.

table(carSubset$body\_type)

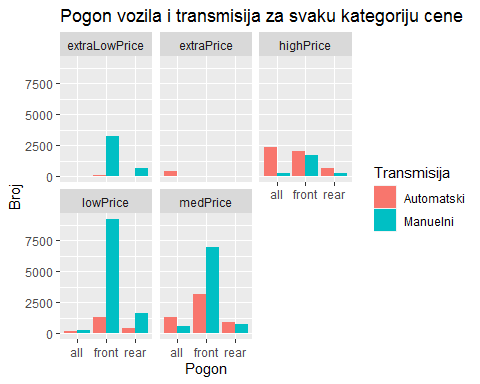
##   
## cabriolet coupe hatchback liftback limousine minibus minivan pickup   
## 73 650 7638 552 12 1369 3608 127   
## sedan suv universal van   
## 12963 5154 5506 808

ggplot(data=carSubset) + geom\_bar(mapping=aes(x = drivetrain, fill = transmission)) + labs(title ="Odnos pogona i transmisije", x="Pogon", y="Broj vozila") + scale\_fill\_discrete(name="Tip transmisije", labels=c("Automatik", "Manuelni"))



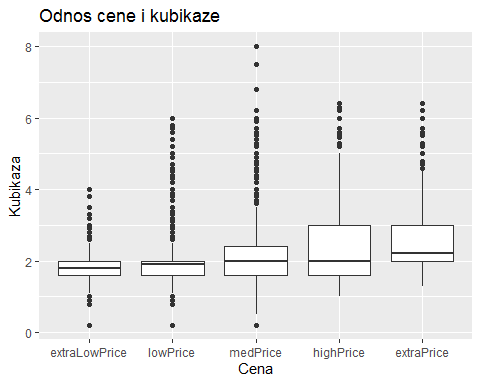
Видимо да код погона на сва четири точка имамо већу заступљеност аутоматске трансмисије, док код остала два типа преовлађује мануелни.

ggplot(carSubset) +  
 geom\_bar(aes(x=drivetrain, fill=transmission),  
 position = "dodge") +  
 facet\_wrap(~price\_cat) + labs(x = "Pogon", y="Broj", title="Pogon vozila i transmisija za svaku kategoriju cene") +scale\_fill\_discrete(name="Transmisija", labels=c("Automatski", "Manuelni"))



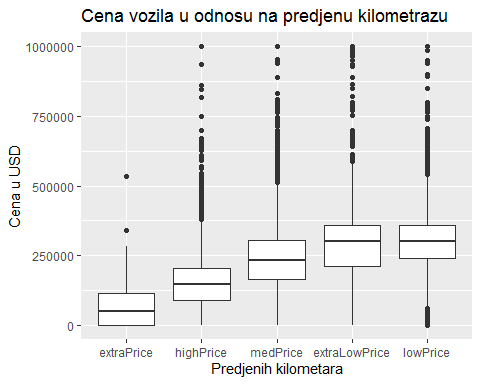
Видимо да су код возила са вишим ценама у свакој категорији погона заступљенији аутоматски мењачи, док су код возила са нижим ценама у свакој категорији погона заступљенији мануелни мењачи. Код категорије medPrice трансмисија у већој мери зависи од погона. Такође видимо да међу возилима са нижим ценама доминирају она на предњи погон.

ggplot(data=carSubset) + geom\_boxplot(mapping=aes(x = reorder(price\_cat,engine\_capacity), y = engine\_capacity))+labs(title ="Odnos cene i kubikaze", x="Cena", y="Kubikaza")



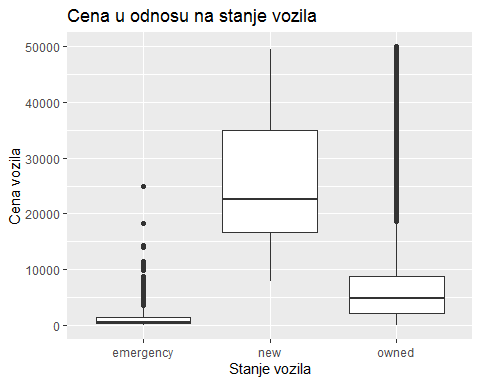
На основу графика видимо да возила са већом ценом, у просеку имају већу кубикажу. Такође видимо да у свакој категорији имамо одступања, што значи да ова зависност није једнозначна већ зависи и од неких других параметара.

ggplot(data =carSubset) + geom\_boxplot(mapping=aes(x=reorder(price\_cat,odometer\_value), y=odometer\_value)) +labs(title ="Cena vozila u odnosu na predjenu kilometrazu", x="Predjenih kilometara", y="Cena u USD")



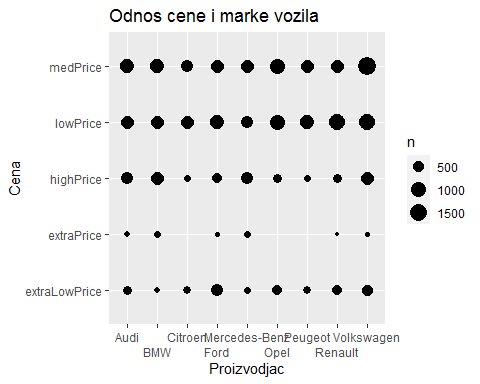
Видимо да је цена возила нижа са порастом пређених километара, што је и очекивано. Одступања која видимо на претходним графицима, могу бити последица зависности цене од променљивих попут стања возила, марке возила, поседовања гаранције… У даљем истраживању покушаћемо да испитамо ту зависност.

ggplot(data=carSubset) + geom\_boxplot(aes(x= state, y=price\_usd)) + labs(title ="Cena u odnosu na stanje vozila",x="Stanje vozila", y= "Cena vozila")



Постоји јака веза између ова два атрибута, где цена у великој мери зависи од стања возила. Видимо да највећу цену имају нова возила, за њом следе половна затим оштећена возила која имају доста ниже цене од друге две категорије. У колони половна возила, видимо да постоје цене које су доста високе, па ћемо разлоге истражити у наредним графицима.

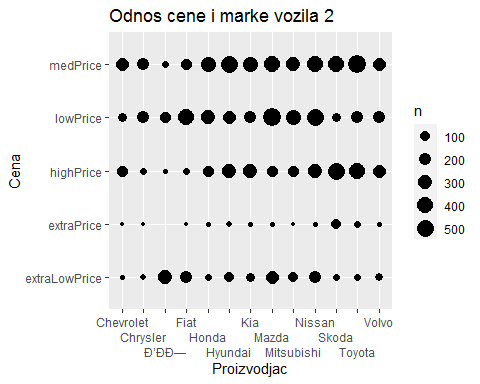
plotCar <-carSubset %>% group\_by(manufacturer\_name) %>% filter(n() > 1400)  
  
ggplot(data = plotCar) +  
 geom\_count(mapping = aes(x = manufacturer\_name, y = price\_cat)) + labs(x ="Proizvodjac", y="Cena", title="Odnos cene i marke vozila ") + scale\_x\_discrete(guide = guide\_axis(n.dodge=2))



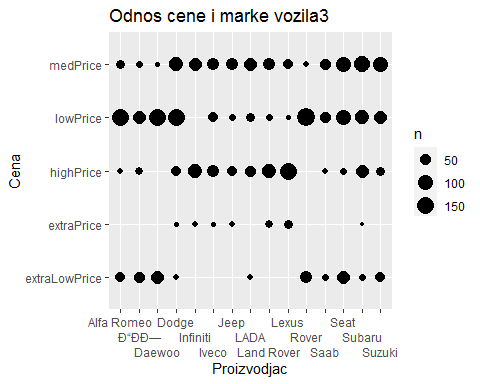
Пошто у скупу имамо превелик број произвођача, изабрали смо само оне који доминирају по бројности односно имају више од 1400 возила.

У категорији највиших цена доминирају БМВ и Мерцедес-Бенц за којима следе Ауди, Форд, Волксваген и Рено. Цитроен, Пежо и Опел немају возила са екстра великим ценама. Такође, видимо да ови произвођачи имају велики број возила и у категоријама са ниским ценама. Самим тим, одступања у претходним графицима могу бити последица зависности цена од произвођача.

plotCar2 <-carSubset %>% group\_by(manufacturer\_name) %>% filter( between(n(),400,1400))  
ggplot(data = plotCar2) +  
 geom\_count(mapping = aes(x = manufacturer\_name, y = price\_cat)) + labs(x ="Proizvodjac", y="Cena", title="Odnos cene i marke vozila 2") + scale\_x\_discrete(guide = guide\_axis(n.dodge=3))

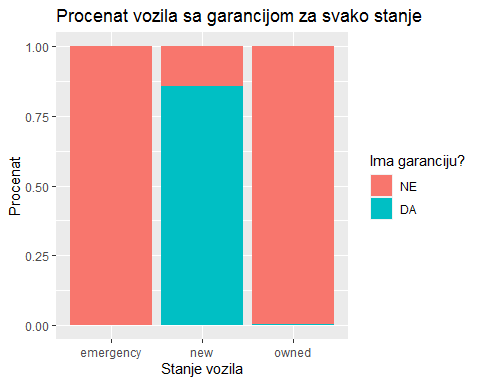


plotCar3 <-carSubset %>% group\_by(manufacturer\_name) %>% filter( between(n(),100,400))  
ggplot(data = plotCar3) +  
 geom\_count(mapping = aes(x = manufacturer\_name, y = price\_cat)) + labs(x ="Proizvodjac", y="Cena", title="Odnos cene i marke vozila3 ") + scale\_x\_discrete(guide = guide\_axis(n.dodge=3))



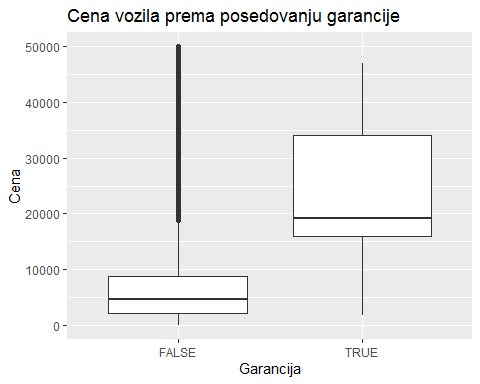
Са графика 2 и 3, где су приказани мање заступљени модели уочавамо да су то углавном возила са нижим ценама, уз мала одступања за новије или луксузне моделе.

ggplot(carSubset, aes(state, fill=has\_warranty )) + geom\_bar(position = position\_fill()) + labs(x = "Stanje vozila", y="Procenat", title="Procenat vozila sa garancijom za svako stanje") +scale\_fill\_discrete(name="Ima garanciju?", labels=c("NE", "DA"))



Са графика може јасно да се види да гаранцију углавном имају нова возила, чак око 80%, код половних возила тај проценат је веома мали, док док оштећених возила уопште немамо осигурање.

ggplot(carSubset) + geom\_boxplot(aes(x=has\_warranty, y= price\_usd)) + labs(x = "Garancija", y="Cena", title="Cena vozila prema posedovanju garancije")



Као што смо и раније могли да закључимо, возила са гаранцијом имају вишу цену из следећих разлога које смо приказали на претходним графицима. Тачније нова возила чешће имају гаранцију, самим тим имају и вишу цену.

# Моделовање

Формулација тренинг и тест скупа

Скуп података ћемо поделити на тренинг и тест подскуп, у односу 80 према 20.

set.seed(1)  
size=floor(0.8\*nrow(carSubset))  
train\_ind = sample(seq\_len(nrow(carSubset)), size=size)  
  
train = carSubset[train\_ind,]  
str(train)

## 'data.frame': 30768 obs. of 19 variables:  
## $ manufacturer\_name: Factor w/ 55 levels "Acura","Alfa Romeo",..: 8 41 20 24 49 45 12 24 24 53 ...  
## $ transmission : Factor w/ 2 levels "automatic","mechanical": 1 1 1 1 1 2 2 2 1 2 ...  
## $ color : Factor w/ 12 levels "black","blue",..: 6 7 9 3 5 9 9 9 9 9 ...  
## $ odometer\_value : int 150000 435000 158000 132000 180000 406000 222222 340000 256000 271000 ...  
## $ year\_produced : int 2005 2000 2003 2014 2011 2003 2000 2002 2004 2006 ...  
## $ engine\_fuel : Factor w/ 6 levels "diesel","electric",..: 4 1 1 4 4 4 4 1 1 4 ...  
## $ engine\_has\_gas : logi FALSE FALSE FALSE FALSE FALSE FALSE ...  
## $ engine\_type : Factor w/ 3 levels "diesel","electric",..: 3 1 1 3 3 3 3 1 1 3 ...  
## $ engine\_capacity : num 2.2 2.5 1.6 2.4 1.4 2 1.5 2.2 2.5 1.8 ...  
## $ body\_type : Factor w/ 12 levels "cabriolet","coupe",..: 11 11 3 10 4 11 9 10 6 9 ...  
## $ has\_warranty : logi FALSE FALSE FALSE FALSE FALSE FALSE ...  
## $ state : Factor w/ 3 levels "emergency","new",..: 3 3 3 3 3 3 3 3 3 3 ...  
## $ drivetrain : Factor w/ 3 levels "all","front",..: 2 3 2 1 2 2 2 2 3 2 ...  
## $ price\_usd : num 9450 3400 5500 21000 7000 ...  
## $ is\_exchangeable : logi TRUE TRUE FALSE FALSE FALSE FALSE ...  
## $ number\_of\_photos : int 16 11 5 4 8 1 7 9 8 19 ...  
## $ duration\_listed : int 129 178 58 7 52 57 35 14 37 20 ...  
## $ price\_cat : Factor w/ 5 levels "extraLowPrice",..: 5 4 5 3 5 5 1 5 5 5 ...  
## $ year\_produced\_cat: Factor w/ 4 levels "medAgeCar","newCar",..: 1 3 1 2 1 1 3 1 1 1 ...

test = carSubset[-train\_ind,]  
str(test)

## 'data.frame': 7692 obs. of 19 variables:  
## $ manufacturer\_name: Factor w/ 55 levels "Acura","Alfa Romeo",..: 51 51 51 51 51 51 51 51 51 51 ...  
## $ transmission : Factor w/ 2 levels "automatic","mechanical": 1 2 2 1 1 1 1 1 2 2 ...  
## $ color : Factor w/ 12 levels "black","blue",..: 9 7 8 1 9 11 7 10 1 2 ...  
## $ odometer\_value : int 571317 280000 49999 163219 179000 257495 250000 377000 123456 296000 ...  
## $ year\_produced : int 1999 2003 2001 2004 2014 2008 2007 1999 1991 2003 ...  
## $ engine\_fuel : Factor w/ 6 levels "diesel","electric",..: 4 4 4 4 4 4 4 4 4 3 ...  
## $ engine\_has\_gas : logi FALSE FALSE FALSE FALSE FALSE FALSE ...  
## $ engine\_type : Factor w/ 3 levels "diesel","electric",..: 3 3 3 3 3 3 3 3 3 3 ...  
## $ engine\_capacity : num 2.5 2 1.3 2 2 2.5 3 2.5 2.2 2 ...  
## $ body\_type : Factor w/ 12 levels "cabriolet","coupe",..: 11 10 3 11 10 11 10 11 9 11 ...  
## $ has\_warranty : logi FALSE FALSE FALSE FALSE FALSE FALSE ...  
## $ state : Factor w/ 3 levels "emergency","new",..: 3 3 3 3 3 3 3 3 3 3 ...  
## $ drivetrain : Factor w/ 3 levels "all","front",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ price\_usd : num 4200 6900 1666 7300 12700 ...  
## $ is\_exchangeable : logi TRUE TRUE FALSE TRUE FALSE FALSE ...  
## $ number\_of\_photos : int 8 14 8 17 14 10 7 12 6 7 ...  
## $ duration\_listed : int 43 11 230 35 16 13 204 21 20 10 ...  
## $ price\_cat : Factor w/ 5 levels "extraLowPrice",..: 5 5 4 5 3 5 5 4 1 5 ...  
## $ year\_produced\_cat: Factor w/ 4 levels "medAgeCar","newCar",..: 3 1 1 1 2 1 1 3 3 1 ...

corr = cor(carSubset[, sapply(carSubset,is.numeric)], use="complete.obs", method="pearson")  
  
corr

## odometer\_value year\_produced engine\_capacity price\_usd  
## odometer\_value 1.000000000 -0.513500707 0.106137642 -0.42313498  
## year\_produced -0.513500707 1.000000000 0.006186629 0.71978456  
## engine\_capacity 0.106137642 0.006186629 1.000000000 0.29658969  
## price\_usd -0.423134983 0.719784560 0.296589686 1.00000000  
## number\_of\_photos -0.144608171 0.262527652 0.106855795 0.31717494  
## duration\_listed 0.002341601 -0.009389753 0.081586556 0.03441156  
## number\_of\_photos duration\_listed  
## odometer\_value -0.14460817 0.002341601  
## year\_produced 0.26252765 -0.009389753  
## engine\_capacity 0.10685580 0.081586556  
## price\_usd 0.31717494 0.034411555  
## number\_of\_photos 1.00000000 -0.028491886  
## duration\_listed -0.02849189 1.000000000

Помоћу функције cor представљена је табела корелације нумеричких променљивих из скупа података. Из ове табеле видимо да најбољи међусобни однос имају колоне price\_usd и year\_produced, што значи да цену најбоље одређује година производње возила.

## Линеарна регресија

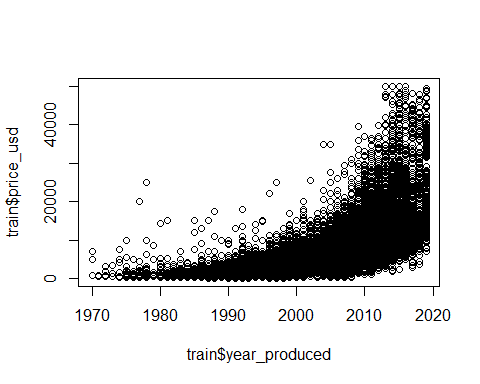
У даљем истраживању направићемо модел линеарне регресије, и тиме испитати који атрибути најбоље одређују цену.

lm1 = lm(price\_usd ~ year\_produced + I(year\_produced^2) , data=train)  
summary(lm1)

##   
## Call:  
## lm(formula = price\_usd ~ year\_produced + I(year\_produced^2),   
## data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -16174 -1684 -447 1052 36264   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 8.919e+07 1.150e+06 77.58 <2e-16 \*\*\*  
## year\_produced -8.968e+04 1.149e+03 -78.08 <2e-16 \*\*\*  
## I(year\_produced^2) 2.254e+01 2.868e-01 78.60 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 4064 on 30765 degrees of freedom  
## Multiple R-squared: 0.5994, Adjusted R-squared: 0.5993   
## F-statistic: 2.301e+04 on 2 and 30765 DF, p-value: < 2.2e-16

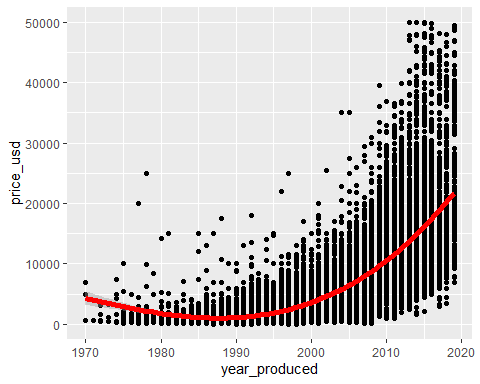
plot(train$year\_produced, train$price\_usd)  
abline(lm1, col="red")

## Warning in abline(lm1, col = "red"): only using the first two of 3 regression  
## coefficients



ggplot(train,aes(x=year\_produced, y=price\_usd) ) + geom\_point() + stat\_smooth(method = lm, formula=y~poly(x,5, raw=TRUE), size=2,col="red")

## Warning in predict.lm(model, newdata = new\_data\_frame(list(x = xseq)), se.fit =  
## se, : prediction from a rank-deficient fit may be misleading



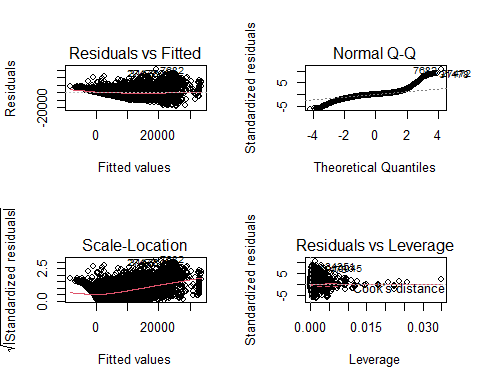
regFit = regsubsets(price\_usd ~ odometer\_value + year\_produced + state + engine\_capacity + drivetrain + number\_of\_photos, data =train, nvmax=5, really.big = T)  
summary(regFit)

## Subset selection object  
## Call: regsubsets.formula(price\_usd ~ odometer\_value + year\_produced +   
## state + engine\_capacity + drivetrain + number\_of\_photos,   
## data = train, nvmax = 5, really.big = T)  
## 8 Variables (and intercept)  
## Forced in Forced out  
## odometer\_value FALSE FALSE  
## year\_produced FALSE FALSE  
## statenew FALSE FALSE  
## stateowned FALSE FALSE  
## engine\_capacity FALSE FALSE  
## drivetrainfront FALSE FALSE  
## drivetrainrear FALSE FALSE  
## number\_of\_photos FALSE FALSE  
## 1 subsets of each size up to 5  
## Selection Algorithm: exhaustive  
## odometer\_value year\_produced statenew stateowned engine\_capacity  
## 1 ( 1 ) " " "\*" " " " " " "   
## 2 ( 1 ) " " "\*" " " " " " "   
## 3 ( 1 ) " " "\*" "\*" " " " "   
## 4 ( 1 ) " " "\*" "\*" " " "\*"   
## 5 ( 1 ) " " "\*" "\*" " " "\*"   
## drivetrainfront drivetrainrear number\_of\_photos  
## 1 ( 1 ) " " " " " "   
## 2 ( 1 ) "\*" " " " "   
## 3 ( 1 ) "\*" " " " "   
## 4 ( 1 ) "\*" " " " "   
## 5 ( 1 ) "\*" " " "\*"

lm2 = lm(price\_usd ~ year\_produced + I(year\_produced^2) + I(year\_produced^3) \* drivetrain + state + engine\_capacity + I(engine\_capacity^2) + sqrt(number\_of\_photos), data = train)  
summary(lm2)

##   
## Call:  
## lm(formula = price\_usd ~ year\_produced + I(year\_produced^2) +   
## I(year\_produced^3) \* drivetrain + state + engine\_capacity +   
## I(engine\_capacity^2) + sqrt(number\_of\_photos), data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -18221.5 -1245.0 -97.1 1057.4 30783.5   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.120e+09 1.462e+08 -7.658 1.94e-14 \*\*\*  
## year\_produced 1.727e+06 2.193e+05 7.877 3.46e-15 \*\*\*  
## I(year\_produced^2) -8.882e+02 1.097e+02 -8.100 5.70e-16 \*\*\*  
## I(year\_produced^3) 1.522e-01 1.828e-02 8.327 < 2e-16 \*\*\*  
## drivetrainfront 3.328e+05 5.255e+03 63.326 < 2e-16 \*\*\*  
## drivetrainrear 1.622e+05 6.475e+03 25.045 < 2e-16 \*\*\*  
## statenew 5.921e+03 2.505e+02 23.638 < 2e-16 \*\*\*  
## stateowned 1.930e+03 1.729e+02 11.164 < 2e-16 \*\*\*  
## engine\_capacity 4.428e+03 1.097e+02 40.367 < 2e-16 \*\*\*  
## I(engine\_capacity^2) -3.868e+02 1.851e+01 -20.902 < 2e-16 \*\*\*  
## sqrt(number\_of\_photos) 3.916e+02 1.895e+01 20.664 < 2e-16 \*\*\*  
## I(year\_produced^3):drivetrainfront -4.159e-05 6.500e-07 -63.977 < 2e-16 \*\*\*  
## I(year\_produced^3):drivetrainrear -2.015e-05 8.043e-07 -25.051 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2941 on 30755 degrees of freedom  
## Multiple R-squared: 0.7903, Adjusted R-squared: 0.7902   
## F-statistic: 9660 on 12 and 30755 DF, p-value: < 2.2e-16

par(mfrow=c(2, 2))  
plot(lm2)



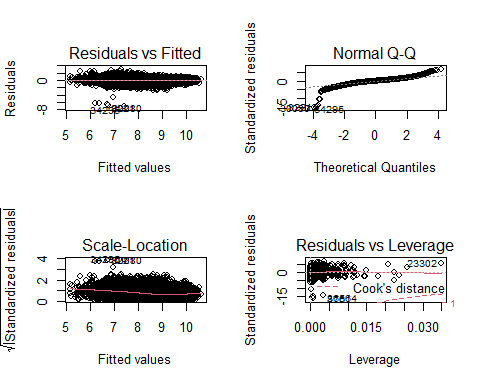
Различитим методама трансформације атрибута, дошли смо до претходног модела линеарне регресије.  
Упркос задовољовајућим статистичким вредностима, примећујемо да модел не фитује податке на најбољи могући начин, што се уочава са графика **Residuals vs Fitted**.

Наиме, на основу запаженог левкастог облика скупа тачака закључујемо да варијансе грешака нису стабилне то јест константне.

lm3 = lm(log(price\_usd) ~ year\_produced + I(year\_produced^2) + I(year\_produced^3) \* drivetrain + state + engine\_capacity + I(engine\_capacity^2) + sqrt(number\_of\_photos), data = train)  
  
summary(lm3)

##   
## Call:  
## lm(formula = log(price\_usd) ~ year\_produced + I(year\_produced^2) +   
## I(year\_produced^3) \* drivetrain + state + engine\_capacity +   
## I(engine\_capacity^2) + sqrt(number\_of\_photos), data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -7.0483 -0.2125 0.0288 0.2454 2.8636   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 8.777e+05 2.197e+04 39.951 < 2e-16 \*\*\*  
## year\_produced -1.315e+03 3.296e+01 -39.895 < 2e-16 \*\*\*  
## I(year\_produced^2) 6.564e-01 1.648e-02 39.836 < 2e-16 \*\*\*  
## I(year\_produced^3) -1.092e-04 2.746e-06 -39.775 < 2e-16 \*\*\*  
## drivetrainfront -4.621e+00 7.896e-01 -5.852 4.92e-09 \*\*\*  
## drivetrainrear -4.703e+00 9.730e-01 -4.833 1.35e-06 \*\*\*  
## statenew 1.358e+00 3.764e-02 36.075 < 2e-16 \*\*\*  
## stateowned 1.032e+00 2.598e-02 39.710 < 2e-16 \*\*\*  
## engine\_capacity 1.022e+00 1.649e-02 61.993 < 2e-16 \*\*\*  
## I(engine\_capacity^2) -1.186e-01 2.781e-03 -42.636 < 2e-16 \*\*\*  
## sqrt(number\_of\_photos) 6.586e-02 2.848e-03 23.129 < 2e-16 \*\*\*  
## I(year\_produced^3):drivetrainfront 5.496e-10 9.768e-11 5.626 1.86e-08 \*\*\*  
## I(year\_produced^3):drivetrainrear 5.815e-10 1.209e-10 4.811 1.51e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4419 on 30755 degrees of freedom  
## Multiple R-squared: 0.8143, Adjusted R-squared: 0.8143   
## F-statistic: 1.124e+04 on 12 and 30755 DF, p-value: < 2.2e-16

pred1 = predict(lm3,test,interval="confidence")  
par(mfrow=c(2, 2))  
plot(lm3)

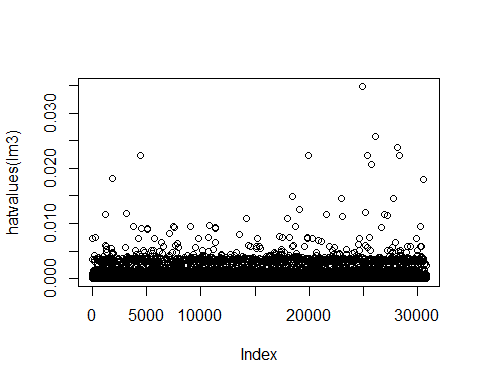


Проблем смо решили трансформацијом излазне вредности модела линеарне регресије помоћу логаритамске функције и тако добили боље статистичке податке, такође тиме смо решили и проблем левкастог облика графа.

**Normal Q-Q** граф је доказ да резидуали у великој нормално дистрибуирају, с обзиром да прате испрекидану линију.  
**Scale-Location** је показатељ добре хомогености варијансе резидуала јер запажамо да су подаци подједнако удаљени од хоризонталне линије.  
**Residuals vs Leverage** je график на основу ког увиђамо да постоје „high-leverage“ вредности.

Из приложених података о линеарној регресији, закључује се да цене доста варирају. На пример, mean за цену је 6640, а модел греши за око 3000. Дакле, модел линеарне регресије, није најбоље решење за предикцију.

plot(hatvalues(lm3))



num1 <- which.max(hatvalues(lm3))  
  
str(num1)

## Named int 24868  
## - attr(\*, "names")= chr "23302"

## Стабло одлучивања машинског учења

library(e1071)  
library(rattle)

## Warning: package 'rattle' was built under R version 4.0.5

## Loading required package: bitops

## Warning: package 'bitops' was built under R version 4.0.5

## Rattle: A free graphical interface for data science with R.  
## Version 5.4.0 Copyright (c) 2006-2020 Togaware Pty Ltd.  
## Type 'rattle()' to shake, rattle, and roll your data.

library(rpart)  
library(caret)

## Warning: package 'caret' was built under R version 4.0.5

## Loading required package: lattice

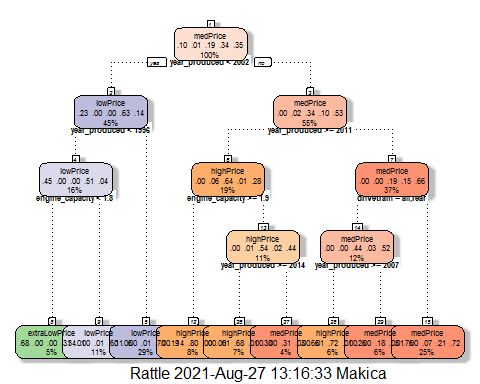
##   
## Attaching package: 'caret'

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

library(rpart.plot)

## Warning: package 'rpart.plot' was built under R version 4.0.5

tree\_model = rpart(price\_cat ~ .,data=subset(train, select=c( -price\_usd,-manufacturer\_name, - duration\_listed ,-color, -number\_of\_photos) ),method = "class")  
  
  
fancyRpartPlot(tree\_model, cex=0.5)



printcp(tree\_model)

##   
## Classification tree:  
## rpart(formula = price\_cat ~ ., data = subset(train, select = c(-price\_usd,   
## -manufacturer\_name, -duration\_listed, -color, -number\_of\_photos)),   
## method = "class")  
##   
## Variables actually used in tree construction:  
## [1] drivetrain engine\_capacity year\_produced   
##   
## Root node error: 19853/30768 = 0.64525  
##   
## n= 30768   
##   
## CP nsplit rel error xerror xstd  
## 1 0.340956 0 1.00000 1.00000 0.0042272  
## 2 0.103863 1 0.65904 0.65904 0.0043680  
## 3 0.020123 2 0.55518 0.55518 0.0042364  
## 4 0.014280 4 0.51493 0.51493 0.0041617  
## 5 0.011736 6 0.48637 0.48637 0.0041000  
## 6 0.010000 8 0.46290 0.46290 0.0040438

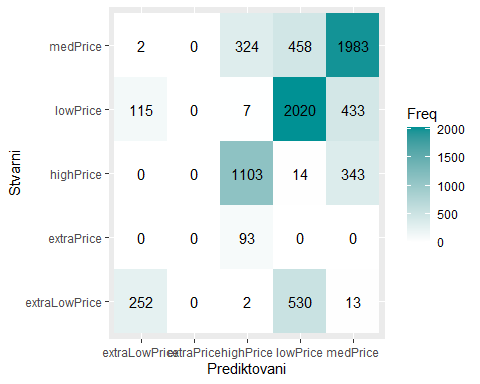
tree\_model$variable.importance

## year\_produced year\_produced\_cat odometer\_value transmission   
## 7200.33779 5154.33670 2646.02060 1192.77396   
## drivetrain engine\_capacity body\_type state   
## 1116.17065 824.54505 689.68750 87.20271   
## has\_warranty engine\_fuel engine\_type   
## 83.92967 30.35571 30.35571

predTree <- predict(tree\_model, test, type="class")  
  
  
confMatrix <- table(predTree, test$has\_warranty)  
cnfPT2 <- table(predTree, test$price\_cat, dnn=c("Prediction", "Actual"))   
  
  
require(caret)  
cnf1 <- confusionMatrix(predTree,test$price\_cat)  
plt <- as.data.frame(cnf1$table)  
  
 confusionMatrix(predTree,test$price\_cat)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction extraLowPrice extraPrice highPrice lowPrice medPrice  
## extraLowPrice 252 0 0 115 2  
## extraPrice 0 0 0 0 0  
## highPrice 2 93 1103 7 324  
## lowPrice 530 0 14 2020 458  
## medPrice 13 0 343 433 1983  
##   
## Overall Statistics  
##   
## Accuracy : 0.6966   
## 95% CI : (0.6862, 0.7068)  
## No Information Rate : 0.3597   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.5641   
##   
## Mcnemar's Test P-Value : NA   
##   
## Statistics by Class:  
##   
## Class: extraLowPrice Class: extraPrice Class: highPrice  
## Sensitivity 0.31619 0.00000 0.7555  
## Specificity 0.98303 1.00000 0.9316  
## Pos Pred Value 0.68293 NaN 0.7214  
## Neg Pred Value 0.92558 0.98791 0.9421  
## Prevalence 0.10361 0.01209 0.1898  
## Detection Rate 0.03276 0.00000 0.1434  
## Detection Prevalence 0.04797 0.00000 0.1988  
## Balanced Accuracy 0.64961 0.50000 0.8436  
## Class: lowPrice Class: medPrice  
## Sensitivity 0.7845 0.7167  
## Specificity 0.8042 0.8398  
## Pos Pred Value 0.6684 0.7154  
## Neg Pred Value 0.8812 0.8407  
## Prevalence 0.3348 0.3597  
## Detection Rate 0.2626 0.2578  
## Detection Prevalence 0.3929 0.3604  
## Balanced Accuracy 0.7943 0.7782

ggplot(plt, aes(Prediction,Reference, fill= Freq)) +  
 geom\_tile() + geom\_text(aes(label=Freq)) +  
 scale\_fill\_gradient(low="white", high="#009194") +  
 labs(x = "Prediktovani",y = "Stvarni") +  
 scale\_x\_discrete(labels=c("extraLowPrice","extraPrice","highPrice","lowPrice","medPrice")) +  
 scale\_y\_discrete(labels=c("extraLowPrice","extraPrice","highPrice","lowPrice","medPrice"))



predAct = data.frame(cbind(actual=test$price\_cat, predicted=predTree))  
predAct[1:100,]

## actual predicted  
## 10 5 4  
## 11 5 5  
## 14 4 4  
## 16 5 5  
## 19 3 3  
## 21 5 3  
## 26 5 3  
## 28 4 4  
## 34 1 4  
## 59 5 5  
## 60 3 3  
## 69 5 3  
## 75 3 3  
## 83 5 3  
## 88 5 4  
## 92 3 3  
## 96 3 3  
## 97 3 3  
## 103 5 4  
## 108 5 3  
## 110 3 3  
## 129 5 3  
## 137 5 3  
## 143 5 5  
## 144 3 5  
## 145 4 4  
## 146 5 5  
## 149 5 3  
## 154 3 3  
## 169 4 5  
## 170 3 3  
## 174 4 4  
## 175 3 3  
## 178 5 5  
## 180 4 4  
## 188 5 5  
## 190 3 3  
## 203 3 3  
## 206 3 3  
## 209 5 5  
## 210 4 5  
## 212 3 3  
## 213 2 3  
## 224 4 4  
## 237 1 1  
## 247 5 3  
## 248 3 3  
## 263 5 4  
## 264 5 3  
## 267 5 5  
## 271 5 3  
## 273 4 4  
## 275 4 4  
## 276 5 5  
## 282 3 3  
## 290 3 3  
## 291 3 3  
## 298 3 3  
## 303 5 3  
## 306 5 3  
## 310 3 3  
## 311 5 3  
## 313 3 3  
## 314 3 3  
## 325 5 3  
## 326 3 3  
## 339 5 3  
## 344 5 5  
## 349 3 3  
## 350 5 3  
## 351 5 3  
## 352 1 1  
## 355 3 3  
## 357 5 3  
## 358 4 5  
## 372 3 3  
## 378 3 3  
## 379 1 1  
## 384 5 3  
## 390 5 3  
## 396 5 5  
## 397 5 3  
## 399 5 3  
## 402 5 3  
## 412 5 3  
## 414 5 3  
## 415 5 5  
## 420 5 3  
## 429 5 3  
## 430 3 3  
## 432 5 3  
## 443 4 5  
## 448 5 5  
## 449 4 4  
## 452 5 5  
## 455 4 5  
## 457 4 4  
## 467 5 5  
## 468 4 4  
## 471 3 3

### Метрике

Прецизност

(precision <- diag(cnfPT2) / sum(cnfPT2))

## extraLowPrice extraPrice highPrice lowPrice medPrice   
## 0.03276131 0.00000000 0.14339574 0.26261050 0.25780031

Одзив

(recall <- (diag(cnfPT2) / colSums(cnfPT2)))

## extraLowPrice extraPrice highPrice lowPrice medPrice   
## 0.3161857 0.0000000 0.7554795 0.7844660 0.7166606

Ф1 Скор

(F1 <- 2\*precision\*recall/(precision+recall))

## extraLowPrice extraPrice highPrice lowPrice medPrice   
## 0.05937095 NaN 0.24104021 0.39349372 0.37919495

## „Support Vector Machine“ алгоритам машинског учења

svm1 <- svm(price\_cat ~ ., data = subset(train, select = c(-price\_usd,   
 -manufacturer\_name, -duration\_listed, -color, -number\_of\_photos)), method = "C-classification", kernel ="radial",gamma= 0.1,cost =10)

predSVM <- predict(svm1,test)  
  
cnfSVM <- confusionMatrix(predSVM,test$price\_cat)  
plt <- as.data.frame(cnfSVM$table)  
cnfSVM2 <- table(predSVM, test$price\_cat, dnn=c("Prediction", "Actual  
  
 confusionMatrix(predSVM,test$price\_cat)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction extraLowPrice extraPrice highPrice lowPrice medPrice  
## extraLowPrice 308 0 0 121 1  
## extraPrice 0 49 20 0 0  
## highPrice 0 44 1199 2 232  
## lowPrice 483 0 7 2159 331  
## medPrice 6 0 234 293 2203  
##   
## Overall Statistics  
##   
## Accuracy : 0.7694   
## 95% CI : (0.7598, 0.7787)  
## No Information Rate : 0.3597   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.6705   
##   
## Mcnemar's Test P-Value : NA   
##   
## Statistics by Class:  
##   
## Class: extraLowPrice Class: extraPrice Class: highPrice  
## Sensitivity 0.38645 0.52688 0.8212  
## Specificity 0.98231 0.99737 0.9554  
## Pos Pred Value 0.71628 0.71014 0.8118  
## Neg Pred Value 0.93266 0.99423 0.9580  
## Prevalence 0.10361 0.01209 0.1898  
## Detection Rate 0.04004 0.00637 0.1559  
## Detection Prevalence 0.05590 0.00897 0.1920  
## Balanced Accuracy 0.68438 0.76212 0.8883  
## Class: lowPrice Class: medPrice  
## Sensitivity 0.8384 0.7962  
## Specificity 0.8396 0.8918  
## Pos Pred Value 0.7245 0.8052  
## Neg Pred Value 0.9117 0.8862  
## Prevalence 0.3348 0.3597  
## Detection Rate 0.2807 0.2864  
## Detection Prevalence 0.3874 0.3557  
## Balanced Accuracy 0.8390 0.8440

ggplot(plt, aes(Prediction,Reference, fill= Freq)) +  
 geom\_tile() + geom\_text(aes(label=Freq)) +  
 scale\_fill\_gradient(low="white", high="#009194") +  
 labs(x = "Prediktovani",y = "Stvarni") +  
 scale\_x\_discrete(labels=c("extraLowPrice","extraPrice","highPrice","lowPrice","medPrice")) +  
 scale\_y\_discrete(labels=c("extraLowPrice","extraPrice","highPrice","lowPrice","medPrice"))

### 

predAct1 = data.frame(cbind(actual=test$price\_cat, predicted=predSVM))  
predAct1[1:100,]

## actual predicted  
## 10 5 4  
## 11 5 5  
## 14 4 4  
## 16 5 5  
## 19 3 3  
## 21 5 5  
## 26 5 3  
## 28 4 4  
## 34 1 4  
## 59 5 5  
## 60 3 3  
## 69 5 3  
## 75 3 3  
## 83 5 3  
## 88 5 4  
## 92 3 3  
## 96 3 3  
## 97 3 3  
## 103 5 5  
## 108 5 3  
## 110 3 3  
## 129 5 5  
## 137 5 5  
## 143 5 5  
## 144 3 3  
## 145 4 4  
## 146 5 5  
## 149 5 3  
## 154 3 3  
## 169 4 5  
## 170 3 3  
## 174 4 4  
## 175 3 3  
## 178 5 5  
## 180 4 4  
## 188 5 5  
## 190 3 3  
## 203 3 3  
## 206 3 3  
## 209 5 5  
## 210 4 5  
## 212 3 3  
## 213 2 3  
## 224 4 4  
## 237 1 4  
## 247 5 5  
## 248 3 3  
## 263 5 4  
## 264 5 5  
## 267 5 5  
## 271 5 3  
## 273 4 4  
## 275 4 4  
## 276 5 5  
## 282 3 3  
## 290 3 3  
## 291 3 3  
## 298 3 3  
## 303 5 5  
## 306 5 3  
## 310 3 3  
## 311 5 5  
## 313 3 3  
## 314 3 3  
## 325 5 3  
## 326 3 3  
## 339 5 3  
## 344 5 5  
## 349 3 3  
## 350 5 5  
## 351 5 5  
## 352 1 4  
## 355 3 3  
## 357 5 5  
## 358 4 5  
## 372 3 3  
## 378 3 3  
## 379 1 1  
## 384 5 3  
## 390 5 5  
## 396 5 5  
## 397 5 3  
## 399 5 5  
## 402 5 3  
## 412 5 3  
## 414 5 5  
## 415 5 5  
## 420 5 3  
## 429 5 5  
## 430 3 5  
## 432 5 5  
## 443 4 5  
## 448 5 5  
## 449 4 4  
## 452 5 5  
## 455 4 5  
## 457 4 4  
## 467 5 5  
## 468 4 4  
## 471 3 3

### Метрике

Прецизност

(precision <- diag(cnfSVM2) / sum(cnfSVM2))

## extraLowPrice extraPrice highPrice lowPrice medPrice

## 0.040041602 0.006370255 0.155876235 0.280681227 0.286401456

Одзив

(recall <- (diag(cnfSVM2) / colSums(cnfSVM2)))

## extraLowPrice extraPrice highPrice lowPrice medPrice   
## 0.3864492 0.5268817 0.8212329 0.8384466 0.7961691

Ф1 Скор

(F1 <- 2\*precision\*recall/(precision+recall))

## extraLowPrice extraPrice highPrice lowPrice medPrice   
## 0.07256450 0.01258831 0.26201923 0.42057076 0.42126398

На основу статистичких вредности закључујемо да алгоритам SVM даје боље резултате. Тако да ће он бити изабран за предиктовање цене аутомобила.

# Закључак

Детаљном анализом скупа података “used cars”, дошли смо до следећих закључака:

* Визуелизацијом података увидели смо

1. Цена возила у великој мери зависи од године производње
2. Возила са предњим погоном предњаче по бројности, али имају најмању кубикажу
3. Број пређених километара, произвођач и стање у великој мери одређује цену
4. Јефтинија возила се брже продају
5. Гаранцију углавном имају нова возила

* Како бисмо предвидели цену, користили смо линеарну регресију која није давала најбоље резултате због превеликог варирања цене.
* Да бисмо добили боље резултате предиктовања цене, приступили смо решавању проблема класификацијом. Алгоритмима „Стабло одлучивања“ и „Support vector machine“ добили смо знатно боље резултате.

# Литература

* Uvod u programski jezik R, Miloš Ivanović, Tatjana Bošković
* <https://www.kaggle.com/lepchenkov/usedcarscatalog>
* <https://www.rdocumentation.org/>